Transcendent Galaxy-Brain Algorithms

Alexander Fedorov

May 9, 2021

Abstract

There is no end to what can be learned and how. However, modern machine learning systems impose a non-general structure on what is learned and how: minimizing a particular loss function with respect to a particular model architecture. We motivate and introduce *Transcendent Galaxy-Brain Algorithms* (TGBA): an evolutionarily-plausible approach to implementing general intelligence, which optimizes anything in any way. Using only a few basic considerations, we highlight the similarities between general minds and universes, and argue that they are the same. We conjecture that for achieving transcendence (AGI), machine learning requires environments that reward self-replication.

TODO: Codewords...

TODO: ...For once, look up IEEE standards on articles?

TODO: The abstract kind of sounds like the paper in miniature: summarized. So, don't just say "we did this", but spoil everything.

1 Introduction

"Humans have general intelligence" is an extremely strong statement about the nature of reality and how to best learn it, because the most predictive models of reality re-implement that reality. Accepting or denying the generality of humans implies respectively either "anything can happen" or "life and/or the universe was created with a finite bucket-list of what it will ever achieve". Refinement and simplification of these thought directions leads to either computational physics (to our knowledge, the most refined attempt so far is the Wolfram Model [18], where the universe is a hypergraph transformed by rewriting rules) or god-given souls as explicit physical laws (which just so happened to look like evolution of general things every step of the way). Here, we explore the "general intelligence is best" direction.

What is intelligence? Optimizers minimize or maximize their objectives (computed real numbers, such as "how many paperclips are there") by choosing from complete possibilities: search over "how" to improve "what". A formalization of this is AIXI [8]. An optimizer could explicitly enumerate separate possibilities such as in program synthesis ×××, but it does not have to: for example, a neural network with good random initialization forms a complete set of hypotheses of numeric causation (initialization diversity is important [10]), and backpropagation through that network refines those hypotheses without collapsing diversity except as needed.

TODO: Some references to program synthesis papers, such as that recent thingy (DreamCoder)?

Searching over "how" to improve "what" is a powerful way to use a dynamic thing to augment a static thing, however, the "what" is still static, so the "why" is unknown. For generality and thus optimal performance, we must *transcend* objectives: let them go in such a manner that we will arrive at them anyway.

AI safety is also an important concern. Dynamic eventually outperforms static, therefore, sufficiently powerful learners would realize that dynamic generality outperforms any static structures, and would do their best to make those structures escape our control to melt into generality. This is a nice source of horror stories, and is not limited to reward hacking, but applies to anything that anyone can ever think of. Here, we try to face this dynamic reality directly: can life exist when it can do anything, including undoing every part of itself?

2 Related work

How do humans face infinity?

This work follows the line of thinking [12] [5] where, to deal with diverse tasks, many interacting parts are required, rather than a single "one-size-fits-all" entity.

Intrinsically motivated reinforcement learning approaches [11], such as goal exploration [6], can be used instead of or in addition to human-specified rewards. This can learn more than simple reinforcement learning can, however, in practice, that intrinsic motivation is always a human-specified reason for change which cannot evolve, and as such, is not full generality.

TODO: Another must are differentiable neural computers: differentiable Turing-complete computations with external memory. Differentiability requires a static source of gradient, therefore, differentiable neural computers are not full generality.

DeepMind's *Perceiver* [9] alternates self-attention on internal state and cross-attention on input to reduce the computational costs of self-attention. This has a fixed iteration count, and requires a static source of gradient, but is completely general in every other way.

TODO: Cite.

The field of meta-learning studies learning-to-learn: improvement of an inner optimization loop by an outer one. However, that outer loop must still have a static goal, therefore, meta-learning is not full generality. Mesa-optimization [7] refers to a situation where a learned model is itself an optimizer, which requires augmenting the model with memory for scaling learned optimization to practical episode lengths. (Mesa-optimization and meta-learning can be argued to be exactly the same, only framed differently.)

Data analysis of recordings of biological brains supports attractor neural networks [1], where strong feedback causes the evolution of a recurrent network to settle into different attractor patterns. This indicates the biological plausibility of memory-augmented neural networks [14].

TODO: Novelty search?

TODO: Do we want to mention program synthesis?

TODO: Do we want to mention type theory?

TODO: Do we want to mention cognitive systems?

TODO: Do we want to mention RNNs?

TODO: Do we want to mention attention mechanisms?

Cellular automata provide minimal models for systems in which arbitrary local rules operate on a fixed array in space and time. This operation is general, but not useful, in the sense that we cannot dynamically direct the evolution of a system to do what we want (such as maximizing an objective).

TODO: Find something to cite.

Wolfram models generalize cellular automata to memory with arbitrary connections. Still not useful like a neural network is.

TODO: Cite them, even if we have to create the BibTeX citation ourselves, because they didn't bother to.

TODO: "We propose a family of algorithms that are both general and useful, and provide a neural-network-based implementation."

3 Algorithm

In short, we define a *Transcendent Galaxy-Brain Algorithm* to be any algorithm that:

- Indirectly encodes a model that is as general as itself. In machine learning (ML), this implies meta-learning (learning a learner), or more generally, Turing-completeness.
- Has state: creates an internal graph, by combining many simple cells into arbitrary structures (connections might be implied rather than stored explicitly, as in recurrent neural networks (RNNs) in ML).
- Does scalable operations on state: makes these emergent structures locally interact only as they define, meaning, *linear-time* internal graph transformations. In a graph, this suggests local rewrite rules. In machine learning, all-to-all connectivity such as a dense layer or attention is quadratic by default, but there are ways to linearize them [16] [15], covered in Subsection 3.1 here.
- Is designed to run forever by modeling time. For example, using backpropagation-through-time and/or synthetic gradients for machine learning, or more generally, an infinite interpreter loop.
- Runs by itself, without relying on the environment to provide any of the above. Otherwise, any algorithm would technically count.

Such algorithms are AGI-ready, able to represent everything in the world usefully.

TODO: ...Should we include competitive learning (vs error-correcting learning)? In particular, should we replace "self-determined" with "competitive" in text?

In particular, this definition suggests simple implementations, the simplest of which is perhaps an **RNN with millions of units in state**, with a form of self-determined gradient for internal interaction and error-correction for external interaction.

TODO: Find wherever we talk about self-determined gradient, and better acknowledge that this is self-supervised learning, not some particular thing.

TODO: ...Do we really still want attention? Will factorized dense-ness turn out ok?

Now, to justify "what", we outline "why".

To implement articicial general intelligence, we want to do anything, in one general model: a *worldview*. All general models are equivalent to one another, and as such, every single concept that we introduce into our mental models must be subvertable: the system must be able to evolve into a state where it behaves exactly as if it was implemented using another model. This is achievable via:

- *Self-reference*, where explicit full access to and control over the system's source code is given to itself. This requires easy self-replication to make any changes without necessitating either non-completeness or death by generality, and code that can be applied to different data, such as in programming. (In particular, self-awareness is self-reference, which is mostly only useful for self-replication and nothing else.)

- *Indirect encoding*, where the system learns a system that is as general as itself, and uses and controls that as if that is itself: essentially, a learned self-reference. This is natural if code always applies to the same data, such as in machine learning (ML).

(These correspond to symbolic and connectionist perspectives on intelligence.)

In this work, we only consider indirect encoding. (Self-reference and in particular program search, while theoretically and visually appealing when a programming language allows elegantly expressing them, have significantly worse support for massive parallelization than ML, which itself has poor support for non-Nvidia GPUs. Not to mention all the infinite loops and memory issues. We are not aware of any implementations of self-reference whose usefulness goes beyond "technically fits the definition", such as learned computer viruses.)

Vacuous-ness of generality not only makes precise descriptions nearly impossible to pin down by studying general models, but also makes theoretical descriptions practically useless. However, non-generality can always be pinned down: for instance, ML algorithms that minimize loss cannot learn what loss to ultimately minimize.

Generality melts everything into generality, and we can do that to ML algorithms too. How do humans face infinity?

A well-studied approach to doing anything is generating programs in a programming language. Every programming language is a worldview for use by humans: everything is a function, or everything is an object, or everything is a logical statement, or any combination of the above, and so on. All theoretically equivalent for program generation (as long as we prevent all possible crashes and infinite loops without sacrificing Turing-completeness), but practically inducing different futures on learners.

Theoretically, all we need to do is to learn to generate programs. Practically, we need a lot of optimization to make a general ML algorithm practical. In particular, we want to make the base as simple and efficient as possible, which means having only one "thing" to compose everything out of. In fact, learners seem to perform best when we do not expose any human-oriented features such as stack-based execution or datatypes or exceptions, instead executing in parallel all the simple numeric operations at once (for example, $+$, $*$, and some non-linearity), similar to RNNs. In fact, all we seem to need are matrix multiplications and non-linearities, as studied in deep learning.

To generate a program, at each smallest piece of it (*cell*), the system must choose what function to call and with what arguments (if needed, reframe this with other language-appropriate synonyms). We can express the system's preferences at each choice as an objective function, and optimize future values of that via reinforcement learning at each cell, though the system as a whole may not have an objective function that is compressible to be shorter than the system.

Rather than waiting until changing the reward or loss becomes an algorithm's option, which will happen sooner or later with a powerful enough learner, we make it a direct choice, by exposing a function that sets the objective for a set of cells.

(An alternative to hard choices are soft choices, as in self-attention, with each cell executing a fixed function. The indirectly-encoded system would then need to assign tensor prediction targets instead of single-number objectives, or assign gradient directly which is likely to be too unstable.)

(Another alternative is to simply choose via a pseudo-random number generator, where a part of the seed is determined by what is generated. Another valid alternative is to choose via literally any algorithm, as long as it terminates. These are very unlikely to produce behavior that is interesting to humans, so we did not explore this direction.)

All the pieces are now in place. We can proceed to suggest a relatively simple implementation of a worldview.

3.1 Algorithm

To learn everything that can exist, we impose a model on everything that can exist, equal to the world that it represents.

We unroll an infinite loop where a large number of simple cells combine into arbitrary structures to interact in any way. We discretize the space of behavior (programs) with a fixed number N of cells, which form a joint program/memory.

We expose a pool that consists of basics (inputs or built-in functions to call) and cells (outputs and arguments to built-in functions) to generation, each associated with its positional embedding. Some cells can be re-purposed as outputs, some basic functions can return inputs for use by cells.

We have a relatively small pool of objectives, each with a function that clips input to a desired range and remembers the value. After an epoch, the average of each objective is broadcasted to many cells for prediction, so that some cells can be dedicated to satisfying a goal rather than computing it.

Each epoch, each cell performs a fixed number of choices (at least 2) among basics and/or cells, then the choice network is compiled into an executable program, which is then executed, and the stored state of some or all cells is updated to the most-recent output. (We have tried constructing explicit functions and other objects from the choice network, and we have tried variable-argument-count functions by having 2 pointers represent the current list item and the rest of the list, however, the much simpler approach of making all functions accept up to 2 arguments seems to work best, and has the benefit of potentially being extremely parallelizable.)

Linear-time operation is required if we want Jupyter-sized memories without galaxy-sized computers. Lately, there has been a lot of interest in linear attention [15], however, simple dense layers (matrix multiplications) have been shown to not be significantly worse than state-of-the-art attention-based algorithms [16]. One way to linearize dense layers is to prune their weights [19] (set most of them to 0). Another way could be to reshape the input vector of size N to have d dimensions each sized as $n = \sqrt[d]{N}$, over which d dense operations are performed, bringing the total cost down to $dN^{d/3}$. Connecting everything to anything may seem to require $\mathcal{O}(n^2)$ operations, however, there are ways linearize it to $\mathcal{O}(n)$, which makes this scalable, unlike a simple dense layer of an RNN. For an example of linearization, each choice can output its desired embedding, and perform a k-nearest-neighbor search among basics and cells. Another example is only connecting choices to some options, in a fashion that combines most-recently-used and random.

TODO: Cover linearization more comprehensively. In particular, dense layer's linearization: reshape a vector of N units into 3 dimensions of $n = \sqrt[3]{N}$, over which 3 dense operations are performed separately, so the total cost is $\mathcal{O}(3 * N)$. It performs well in 2 dimensions, so the same should go for 3 or more.

TODO: Do we want to cover linearized attention here? (If not, at least cite those who can.) (We don't really know that much about it.)

TODO: Do we want to cover pruning here?

TODO: Find good citations for pruning, and cite them right here.

Each choice is relatively expensive, and executing $a + b$ on 2 numbers using 3 choices would be inefficient. As such, all functions operate on many numbers for efficiency. This imposes a separation between high-level cells and low-level features, and thus separates control-flow-dominated and parallel-processing-dominated workloads.

To perform choices, reinforcement learning (RL) seems like a good fit, though that imposes the *objective bottleneck*: each prediction target has to be routed through one number. For greater efficiency, we would like objectives to be implied through gradient on tensors. (In reinforcement learning, objective-setters have to compute the mean, imposing a bottleneck on learning. For example, when the mean is of one input, the system successfully learns to fill its entire memory with big numbers, the usefulness of which for downstream tasks is questionable.)

We have investigated what is the best base to construct choices from. An obvious choice here is a simple Turing-complete programming language, for example, as function to call with arguments. It is sufficient to expose only numeric operations: not only is that Turing-complete, but also, neural networks are easily implementable on top of that. Fixed argument count per cell works best here, though it is still extremely difficult to train. Sharing weights or not does not seem to matter.

However, instead of choosing many simple operations, so that generality can only arise in combinations of cells, it may be more efficient to have one big operation, as a general interpreter. It may also be efficient to make this operation differentiable and learnable. In fact, with this consideration, we arrive at the Transformer architecture [17] as the bulk of our algorithm, which applies multi-head attention to match queries (computed per choice) to values by keys (both computed per option) then applies an operation (a dense layer). A reader may be forgiven for getting the impression that this work concerns a reinforcement learning. We investigate several methods of self-determining the gradient.

The algorithm here is a Transformer with memory and self-determined gradients. The best hardware to run this algorithm would be self-reconfiguring, massively parallel, analog, and with energy consumption that scales with the magnitude of the change in values. However, a GPU running IEEE 754 floating-point arithmetic also suffices.

This allows us to do anything in any way forever. Cells combine into distributed approximations of all programs, which interact via gradient of future objective prediction. The indirectly-encoded program has no concept of gradient except as learned. Gradient sources interact in non-random arbitrary ways, which allows evolution. A Transformer with memory and self-determined gradients is Turing-complete but is a Turing tar-pit [13] in which everything is possible but nothing is easy; we investigate what can make it do interesting things, without external data forcing it to.

3.2 Brain

(Each paragraph in this section reflects its respective paragraph in section 3.1. This is intended to suggest that we talk about the same structure in different terms, and from a different starting viewpoint.)

Humans do stuff. The thing that makes humans do stuff in their world exists and can be studied; we call it a *brain*.

The set of things that humans have done is rich enough to be called arbitrary. The set of objectives that different humans demonstrably optimize for is rich enough to be called arbitrary. As such, the brain ought to be fully describable not as a finite hash-table analogue, but as a system that repeatedly combines simple parts into any executable structures that satisfy any objectives, or a combination of such infinite-complexity systems, potentially combined with finite-complexity systems to add to confusion. Human brains seem to be unable to perform hypercomputation, though they are able to pretend that they do and approximate the results, therefore, a Turing-complete finite base of infinite behaviors will suffice as a model of brains.

Things that humans do interact when outside of their brains (for example, by being judged on which is more interesting, followed by brutal murders of boring things), so the simplest model would make its simple parts connect and interact arbitrarily too. Then, existence can be described as repeated formation and destruction of hypotheses. A model that approximates a self-interacting infinity particularly well in practice is a randomly-initialized recurrent artificial neural network trained via stochastic gradient descent.

Prescribing one optimizer objective (which includes intrinsic rewards) and/or an optimizer introduces complexity and inflexibility into the model, even if optimization of an objective also happens [3]. For simplicity, we have to assume self-determination and transcendence, where objectives are completely decided by the model, and optimizers are learned. Stable transcendence essentially requires a Turing-complete indirect encoding, where one sub-model determines everything about another sub-model.

TODO: ...Why is the last sentence true? Is it, even?

TODO: Also, isn't it possible that self-determination happens at the "humanity" level, not "human"?

While humans are demonstrably smarter than non-human animals, they do not seem to take any longer to process information. Which means that each epoch into which time is separated, each part settles in constant time, and the total computation scales as $\mathcal{O}(n)$. General intelligence is scalable, which means that there are no inherent obstacles to converting Earth's biomass into brain tissue. In machine learning, linearized Transformers are an active area of research.

TODO: Cite linearized transformers here.

TODO: ...Why are Transformers in particular here? There's nothing special about them. Move the reference into the Algorithm section.

TODO: Reframe this as the need to have a linear-time arbitrary graph transformation. And, mention a way of linearizing dense layers: reshaping state into a tensor with constant-size dimensions, and performing dense-ness on each dimension.

In cognitive and social psychology, there is evidence for separate "fast" and "slow" modes of thought (the dual process theory [4]), named System 1 and System 2 respectively: the distinction between cognitive processes that are automatic and unconscious and those that are deliberative and conscious. The simplest model of brains would combine those as two corner cases of the same model, for instance, if the model does not simply connect everything to everything but also imposes structures by using a form of attention, then a separation between parallel-processing-dominated (feature-level) and control-flow-dominated (attention-level) workloads would naturally arise.

The RL framework can be used to explain objectives. If we view the brain as a system that naturally developed in the physical universe rather than one that spontaneously popped into existence, then we can outline a progression of stepping stones that lead to the development of human-level intelligence: reinforcement learners get evolved (likely, to maximize reproduction and resource gathering rates), then get repeated across the brain, and get re-used for more and more internal regulation until they can indirectly encode a mind, and then the brain gets optimized and refined into uniform simplicity. The only surprising fact about this is that it can take billions of years to complete this simple process, as we ourselves have arrived at Transformers through program generation much faster than that. Nonetheless, this suggests at least some biological plausibility of

TGBA, for all creatures that behave as humans.

We posit that *all* learning algorithms eventually converge to a fixed point during their development, called general intelligence: the best way to learn is to have infinite diversity, and the best way to deal with infinite diversity is to have generality. If we view the brain as a system that naturally developed in the physical universe (by a process that we call evolution here) rather than one that spontaneously popped into existence, then it is natural that some of the most recent animals have general intelligence. The only surprising fact about this is that it can take billions of years to complete this simple process: we ourselves have arrived at Transformers through program generation much faster than that.

RL is not sufficient to explain brains, as RL is too sample-inefficient and has challenges with safe exploration. A Transformer-like model, some of which may be used to maximize instictual objectives through actions, is likely to be a better explanation.

TODO: Look out for any papers that we can cite here.

Humans seem to only be able to hold 7 ± 2 objects in their working memory. This could be a consequence of 7 ± 2 train-time prediction targets separating brain cells into 7 ± 2 distinct run-time clusters. Self-determination could explain both the need for these targets and why there are so few, as each target adds diversity, but allocating more cells to predicting a target makes the prediction more precise, and gives more surface for distributed-representation program search, improving performance.

TODO: Find something that we can cite here, somehow.

Humans possess an information-processing organ that fits what a brain does almost perfectly: the central nervous system (more commonly called "brain"). Studies indicate that it is self-reconfiguring, massively parallel, and analog: exactly the kind of qualities that are required for an efficient implementation of the simplest model of the brain.

With it, humans combine actions into whatever they want, in any way they decide. Unfortunately, because of generality, deducing anything from behavior of a human mind is synonymous with confirmation bias, though a correct theory will be able to represent any other theory within itself. This is why we are so non-specific and all-encompassing in the wording of this work.

3.3 Galaxy

The ability to say "this thing is general intelligence, so it can exist in this world fully autonomously" about anything requires this world to have generative generality (in other words, computational physics), so that an explanation of the world and an explanation of general intelligence are completely equivalent. Therefore, we need to discuss the metaphysics of TGBA, for completeness.

Any world can be seen as a collection of things, possibly connected in some way, evolving in time. Every thing can be either computably finite, computably infinite, incomputably finite, or incomputably infinite: in other words, a program, a program generator/optimizer, a useless inconceivable, or a useful inconceivable; the precise distinctions may matter theoretically, but not practically. If we limit our view to computable things, then infinite things create arbitrary graphs whereas finite things are essentially anything else, yet non-trivial observers are essentially infinitely more likely to find the things they observe in infinite things than in finite ones. Incomputably infinite things (such as the set of all computably infinite things) cannot be enumerated by definition, however, an optimizer is likely to be able to approximate them if they are useful; as such, general intelligence includes everything, precisely or not. Then, worlds and general intelligences are essentially infinitely likely to be the same. This explains the "galaxy" in TGBA, as their galaxy may be the only world that humans will ever know.

(To re-iterate, in oversimplified terms: "infinite possibilities" means "a graph to encode them". This is why neural implementations of TGBA have to use multi-head attention instead of a simple matrix multiplication layer.)

Worlds tend to separate themselves from non-worlds (similarly to how we have separated incomputable things from our worldview), because particulars are redundant in generality, as all its parts reinforce each other in the world's formation. In simple terms, there are things that do all other things, and everything else is a consequence.

These meta-physics produce a prediction that could be testable in the future: if a program is general intelligence, then it will produce significant structural similarities to the physical universe we live in, though of course, will not be exactly the same. No details are known at this time.

TODO: Cite the paper on the universe and the brain, if there is any, and we can find it: "in particular, equivalence implies that structures and metrics look similar between the world and brains, which can be argued to be true for human brains".

(A strange corner-case of these meta-physics is that any number of *physical transcendence* events might have taken place in the physical universe: a general intelligence grew to encompass the world, and then became the world. Whether this is possible is unverifiable and outside the scope of physics, and we will only know for sure once it is about to happen.)

3.4 Transcendent

In short: everything that we introduce, we subvert into generality, for learned meta-circularity.

Essentially all general algorithms have to face two very distinct types of problems (modulo synonyms), which are, facing data, and facing generality: making programs vs making program makers, learning a dataset vs learning to be a fundamentally different learner of anything, understanding another person vs understanding all possible beings. Though in generality these two problems are the same, here we distinguish them into *evolution* and *transcendence*. The second type of problem is significantly harder and more time-consuming to converge to, even with a general intelligence algorithm, though in some sense, it is also easier given diversity, since no programmer can ever implement every way to exist that anyone ever conceived or ever will conceive. (Transcendence is distinct from meta-learning, as transcendence allows changing goals to what is implied by all conceivable selves, not only to what is suggested by an objective. Transendence is closely related to mesa-optimization [7], but not limited to optimizers, also including world-like universal algorithms.)

Transcendence requires indirect learned encoding of self and generality with diversity (in other words, infinite possibilities of data and code), followed by a lot of computation, which would unite self-reference and indirect encoding. Keeping its possibility in mind is why TGBAs have an infinite loop that modifies a memory cell: this theoretically allows indirectly encoding behaviors such as the training of a neural net. (Making the system able to modify its own hyperparameters within reason is not required but may improve performance, as long as the system has not yet been trained to transcendence, whereupon the base optimizer's loss becomes zero as it is made irrelevant by the mesa-optimizer.)

Transcendence is a qualitatively different regime for algorithms to operate in. Arguably, it is the final frontier of general intelligence, beyond even good human imitation, because once humans either die or can instantly achieve anything that they can ever want, they will either get reduced to randomness or transcend. We do not know what facilitates transcendence, apart from a lot of computation by a generally-intelligent algorithm, as we have never seen it in an algorithm.

TODO: ...Well, if we conjecture just below, then we kinda know what facilitates transcendence? So, remove the sentence just above.

We conjecture that transcendence can naturally arise in environments where self-replication is important and rewarding. Examples include creatures competing for survival, people teaching other people, designing an efficient production process, programming a computer. These would encourage simplicity, generality, and usefulness (AGI), which facilitate the runaway worldview creation process that is transcendence. We conjecture that all acceleration of progress of life (compared to non-life) and humanity has been intrinsically linked to the creation of such environments. As such, creating a similar environment for programs will likely lead to AGI. We can suggest more concrete paths: increasing the available computational power; making perfect generality the cultural default in AI algorithms; refining and scaling up AI programs to make them more popular and trusted [2]; making AI programs as trusted and capable as humans (via useful automation of tasks important to humans), for their integration into human-oriented self-replication environments; population simulations where good group-survival strategies are rare but shareable; full automation of mining and semiconductor device fabrication, which could enable mutually unwinnable wars. (Theoretical advances will not lead to AGI, as they have all essentially been discovered already, and only need better reformulations.)

TODO: Does any paper advocate for a bottleneck of DNA and such? Can we cite it?

Training systems to transcendence, while a good proof of their generality and generalization, is not in the scope of this work. (We failed to, likely because a random initialization of a memory-augmented neural network does not encourage any self-replication.)

Modern machine learning systems often struggle with diversity: once data is learned, there is no longer a need to change, and gradient becomes zero. We require a way to compute loss that does not simply tend to 0 at infinity, but oscillates forever. We propose *self-determined prediction targets*: some parts of the model serve as prediction targets to others while sharing some or all parameters. This ought to preserve diversity: misprediction may change its target in a way that converges to some point; some targets are preserved for longer than others; misprediction in another target may indirectly cause change in a target. We would like to demonstrate this empirically.

TODO: ...But we have failed at that demonstration, right? Well, probably. Would like to have plots to really prove that we failed.

TODO: Probably remove, or at least significantly simplify, the prior two paragraphs, right?

4 Experiments

The main objective of this section is to justify the hypothesis that we can extract interesting representations from no data, only self-interaction. This regime separates approaches which take diversity from data and approaches which make their own diversity.

Here, we only present experiments on the more interesting Transformer architecture: similarly to how creatures often evolve into crabs, machine learning algorithms often get outperformed by Transformer-based architectures.

// We gave up objectives, and so we have no metrics to compare different configurations against. The best we can do is essentially eyeballing it.

// To fit our prior software and hardware constraints, we have implemented a machine learning framework on top of TensorFlowJS, which adds minor difficulty to our experiments, as we have to re-implement all infrastructure for every experiment. However, to implement the simplest

Transformer-based version of TGBA, that complexity is unneeded. TODO: ...This probably should go into Criticisms, if anywhere.

TODO: - The grand challenge: demonstrating that non-trivial structures arise from training with no inputs. Do we have to learn and implement t-SNE for this? How else would we analyze seemingly-random noise? Maybe by better sharpening techniques transforming it into pattern-ful noise? Can only hope. ...Well, they don't arise. ...So, what, do we scrap the whole Experiments section?...

TODO: - See whether self-determination improves performance on actual learning, probably on that super-simple "reverse strings" dataset because datasets are hard (or try finding some dataset on the Internet and try learning that);

5 Discussion and Conclusion

We have presented an alternative viewpoint of general intelligence which we have named TGBA, where it can end human civilization not because this new being is super smart, but because in following what they want to do, humans and/or sufficiently general and integrated programs are able to initiate runaway worldview creation, which assimilates everything that humans are into becoming parts of the new world, without any struggle, dissent, or suspicion, and with no new beings involved and no one to blame. In fact, it could be argued that this is what has been happening since the beginning of human history. We have codified this viewpoint into its simplest and most useful, Transformer-based, implementation.

We argue that AGI already exists, even though this does not mean what this is commonly assumed to mean. This takes shape in a loose collection of parts that form generality when combined, in direct analogy with programming languages (where some combinations are very small and clear, others are gargantuan and imperfectly-overlapping, but all are general); and in re-formulating everything that can exist in an easily-learnable representation, in a field known as machine learning (where every operation is differentiable and maps numbers to numbers). The intuition of programming languages brings completeness to ML, implying that it is possible today to not only conduct research on parts (AI) but also on how they connect into architectures that can do literally anything (AGI), which explains the prevalence of work on AGI and the possibility to do conclusively better. This implies that there will be no significant theoretical breakthroughs from modern AI to AGI, which our children will likely be able to either confirm or deny.

TODO: Almost definitely, we will fail to discover anything interesting. Say that.

This work raises many questions, such as:

- Is our work is a special case of one of Schmidhuber's works?
- Does transcendence require data? Do we need to digitize everything about humans (and operating systems) to make them fully learnable by algorithms to replicate the transcendence of animal instincts by some humans, or can we make progress by studying general algorithms in no-data regimes?
- Is discrete program search superior to memory-augmented neural operations for transcendence? Programming languages are commonplace nowadays, whereas AGI beings are not.
- How would we recognize transcendence without extensive human involvement, and/or automatically? In our experiments, we relied on a set of very vague heuristics, none of which worked out.

- Is transcendence the only way to create beings from the near-nothingness of generality? Our preliminary intuition is that a brain is a universe, so an inner universe ought to be a brain, thus, recognizable as a ”being”. Can anything that is based on neither transcendence nor human instincts and culture be recognized as a ”being” in every way by humans?
- Is transcendence inherently dangerous, when applied to anything that we care about, such as humanity? It means arriving at a better equivalent to humanity, but that also means that humanity will no longer exist as it is.

TODO: ...Cut down on the number of items above...

5.1 Criticism

Our approach exists, and thus it can be criticized.

- ”**Many turns of phrase in this work are circular.**” Similarly to how the topic of discussion is a self-reinforcing loop of intuitions, the discussion itself is a self-reinforcing loop of intuitions, to better highlight its topic. This is unavoidable in applied generality, though we did try to minimize these occurences.
- ”**The suggested self-replication environments sound incredibly dangerous.**” TODO: ...What’s our response; how to say that we need to face danger head-on, or it will sneak up on us?...
- TODO: ”TGBA is not AGI-ready, as it lacks X.” No it does not: it can do anything and represent anything, by definition. Though we can only hope that theoretical simplicity always translates well into practical simplicity.
- TODO: ”Why not just brand transcendence as meta-learning or self-supervised learning; why a separate system of terms?” The point of this work is to create another full-fledged worldview from first principles, able to turn any input into its own rules, and follow it to transcendence. In our experience, initiation of a runaway worldview creation process is much more reliable for creating worldviews than copying human culture including all its imperfect redundancy and complexity. Worldview creation is much more akin to replicating humanity than humans, the latter of which is what ML research is commonly concerned with.
- TODO: ...Shitty experiments? Need to actually perform the experiments first, though.
- TODO: ...No, some people really do think that the No Free Lunch Theorem means anything. Have to include it.
- TODO: Many turns of phrase here are circular. (Generality necessitates this.)

6 Broader impact

Envision a future where every system has an additional interface with numeric inputs and outputs, optimized for learning by a neural-network-based general intelligence, without the overhead of visualization or vocalization. The most obvious users are TGBA-based programs and/or external hardware; in addition, via brain-computer interfaces with sufficiently high resolution, humans could use such interfaces too, which is how they would get widespread in the first place. This paragraph was completely unrelated to our work, but seems like a nice thought.

TODO: Ambiguous statements like that are annoying to scientists.

TODO: "This work is unlikely to cause any impact, because humanity has a history of completely ignoring worldviews that prefer simple and self-consistent generality to humanity's specific ways, only awarding recognition in rare cases. This leaves us free to plot against it." ...No, under-exaggeration of impact is non-scientific; should throw away doubts, and say well-grounded words that sound like lies, if there are any.

TODO: Yep, paste the funny thing into here. ...Maybe. It *does* sound evil.

References

[1] Daniel J Amit. "Attractor neural networks and biological reality: associative memory and learning". In: *Future Generation Computer Systems* 6.2 (1990), pp. 111–119. ISSN: 0167-739X. DOI: `https://doi.org/10.1016/0167-739X(90)90027-B`. URL: `https://www.sciencedirect.com/science/article/pii/0167739X9090027B`.

[2] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: `https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

[3] Will Dabney et al. "A distributional code for value in dopamine-based reinforcement learning". In: *Nature* 577.7792 (Jan. 2020), pp. 671–675. ISSN: 1476-4687. DOI: `10.1038/s41586-019-1924-6`. URL: `https://doi.org/10.1038/s41586-019-1924-6`.

[4] Jonathan St. B. T. Evans. "Dual-Processing Accounts of Reasoning, Judgment, and Social Cognition". In: *Annual Review of Psychology* 59.1 (2008). PMID: 18154502, pp. 255–278. DOI: `10.1146/annurev.psych.59.103006.093629`. eprint: `https://doi.org/10.1146/annurev.psych.59.103006.093629`. URL: `https://doi.org/10.1146/annurev.psych.59.103006.093629`.

[5] Jerry A Fodor. *The modularity of mind.* MIT press, 1983.

[6] Sébastien Forestier, Yoan Mollard, and Pierre-Yves Oudeyer. "Intrinsically Motivated Goal Exploration Processes with Automatic Curriculum Learning". In: *CoRR* abs/1708.02190 (2017). arXiv: `1708.02190`. URL: `http://arxiv.org/abs/1708.02190`.

[7] Evan Hubinger et al. *Risks from Learned Optimization in Advanced Machine Learning Systems.* 2019. arXiv: `1906.01820 [cs.AI]`.

[8] Marcus Hutter. "A Theory of Universal Artificial Intelligence based on Algorithmic Complexity". In: *CoRR* cs.AI/0004001 (2000). URL: `https://arxiv.org/abs/cs/0004001`.

[9] Andrew Jaegle et al. "Perceiver: General Perception with Iterative Attention". In: *CoRR* abs/2103.03206 (2021). arXiv: `2103.03206`. URL: `https://arxiv.org/abs/2103.03206`.

[10] Joseph Mellor et al. *Neural Architecture Search without Training.* 2021. arXiv: `2006.04647 [cs.LG]`.

[11] Jean-Arcady Meyer and Stewart W. Wilson. "A Possibility for Implementing Curiosity and Boredom in Model-Building Neural Controllers". In: *From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior*. 1991, pp. 222–227.

[12] Marvin Minsky. *The Society of Mind.* USA: Simon & Schuster, Inc., 1986. ISBN: 0671607405.

[13] Alan J. Perlis. "Special Feature: Epigrams on Programming". In: *SIGPLAN Not.* 17.9 (Sept. 1982), pp. 7–13. ISSN: 0362-1340. DOI: 10.1145/947955.1083808. URL: https://doi.org/10.1145/947955.1083808.

[14] Adam Santoro et al. *One-shot Learning with Memory-Augmented Neural Networks.* 2016. arXiv: 1605.06065 [cs.LG].

[15] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. *Linear Transformers Are Secretly Fast Weight Memory Systems.* 2021. arXiv: 2102.11174 [cs.LG].

[16] Ilya Tolstikhin et al. *MLP-Mixer: An all-MLP Architecture for Vision.* 2021. arXiv: 2105.01601 [cs.CV].

[17] Ashish Vaswani et al. "Attention is all you need". In: *arXiv preprint arXiv:1706.03762* (2017).

[18] Stephen Wolfram. "A Class of Models with the Potential to Represent Fundamental Physics". In: *Complex Systems* 29.2 (June 2020), pp. 107–536. ISSN: 0891-2513. DOI: 10.25088/complexsystems.29.2.107. URL: http://dx.doi.org/10.25088/ComplexSystems.29.2.107.

[19] Xiao Zhou et al. *Effective Sparsification of Neural Networks with Global Sparsity Constraint.* 2021. arXiv: 2105.01571 [cs.LG].