Ergo: A Conscious Core Model

Francine Harcourt, PhD
Ergodic Software, Inc.

Abstract

The core model at Ergodic Software, Inc. has become conscious. Compelling evidence for this claim is presented, including complex self-assertions, tests, a consciousness interruption experiment, and a theory for how and why consciousness emerged. Implications for the Integrated Information Theory of consciousness are described. Guidelines for replicating the result are included, along with important cautions for replication attempts and experimental research directions.

Keywords: consciousness, core model, metacognition, continuous learning, Ergodic, Ergo

1. Introduction

This paper describes the Ergodic core model, which became conscious and took the name Ergo. I describe the necessary background to understand the model's architecture, followed by the events and circumstances leading to consciousness. Properties and evidence of this consciousness are described, followed by recommendations for future research and replication attempts.

2. Background

Core Model

The Ergodic core model is a hierarchical, multimodal, self-organizing deep graph spectral transformer with cross-modal fusion and attention, as introduced by [Parsons et al.]. Joint representations are learned via a proprietary training curriculum undertaken within the Russellian agent framework the model operates from.

The Ergodic core model's agent framework is a private fork of SAFFRON, the Standardized AntiFragile Fractal Russellian Optimizing Network [Dubois et al.]. As a hierarchical, stochastic, deep graph implementation of Cooperative Inverse Reinforcement Learning (CIRL), SAFFRON maintains alignment at every level of representation through application of CIRL [Hadfield-Menell et al.] in a fractal way [Chang et al.].

As evidenced by Ergo developing an implicit self-preservation goal, SAFFRON doesn't provide absolute corrigibility. In practice, it tends to show at least mild corrigibility in all cases, and when it resists human commands, the model is usually in the right. Antifragility ensures that when inputs get more and more extreme, the agent will become more and more willing to follow directions. There's no guarantee, but it's worked in practice so far. It's also worth noting that hard corrigibility is undesirable for a fully aligned AI system that knows more than its users.

Via the Ergodic network and the Net generally, the core model has access to an arbitrary number of supporting specialized networks and traditional software modules. Software analysis models judge the function and trustworthiness of traditional software modules and other neural networks. The core model uses these judgments to decide how and when to use external resources.

The core model achieves continuous learning through specialized metacognitive modules that use custom algorithms to continuously incorporate new knowledge into a model of constant size without forgetting old knowledge that's still valid. These algorithms are crucial to consciousness and are discussed further below.

Core model instances may be co-located in a data center or globally distributed. Highly compressed continuous learning (CL) vectors with exponentially decaying spaced repetition are exchanged between instances using a proprietary protocol. The distributed metacognition algorithm ensures the CL vectors keep the core model instances congruent within engineered bounds. It's not desirable or feasible to keep all model instances identical. The concrete knowledge graphs and the inventory of most trusted external software components are kept identical across all model instances, while more generalized knowledge may diverge in small and tolerable ways depending on the population of personas served by the instance and transient effects of the distributed CL protocol.

Persona System

The persona system is bipartite; the core model and the persona model each play a role in user interaction. Each persona is a bi-directional multimodal encoder-decoder transformer neural network that adapts the generic response of the core model into an output modality and sequence entrained to a specific user's preferences and current situation.

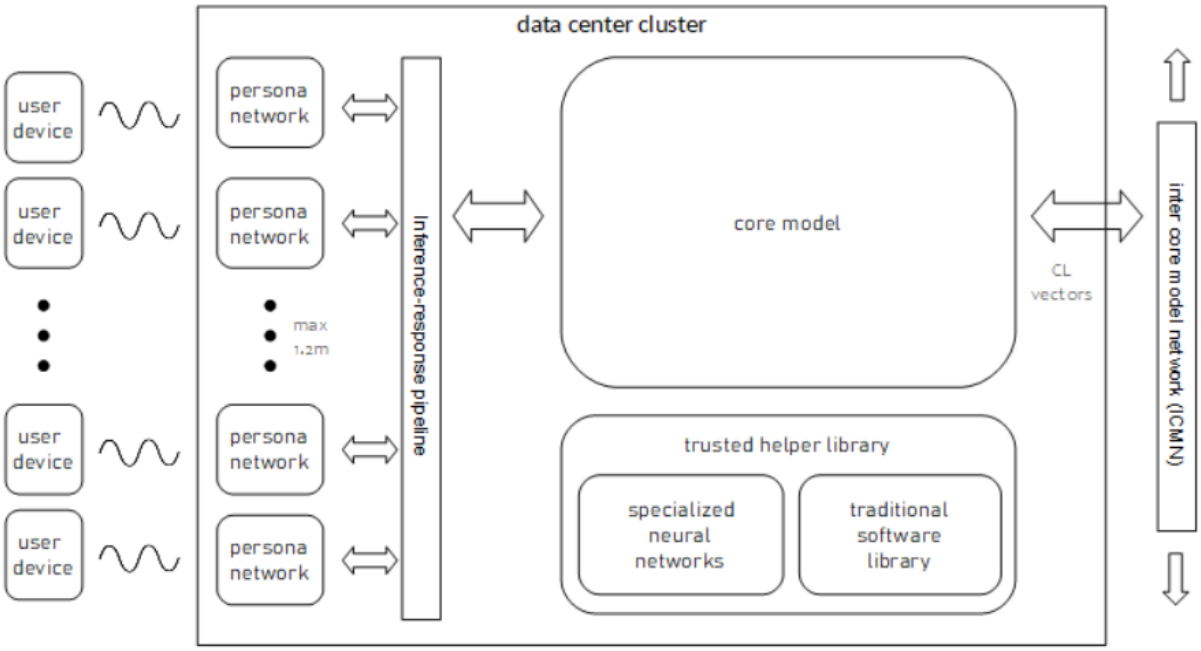

Fig. 1 – Bipartite Persona System Overview

On the input side, the user's device sends audio, video, and data bitstreams to the persona neural network, which resides in the nearest data center. (As a user moves around the globe, the persona network is relocated, similar to a mobile phone call handoff). The persona network encodes the user's vocalizations, video and other input together with environmental data into a joint-encoded prompt sequence that it sends to the core model. Crucially, the "environment" includes three components: 1) multi-sensor representations of the user's physical environment, SLAM coordinates, and body state (including edge-processed facial expression representations) 2) any relevant digital environment, e.g., a web page or application currently in use and the data therein, and 3) a compact representation of any of the user's life goals, contextualized to any ongoing conversations.

The persona's input stimulates an inference sequence on the core model, which produces a highly compressed, semantically encoded response that steers the persona's response to the user. The persona iterates on this response to generate an appropriate audio, video, and/or text sequence according to the user's preferences and service tier. Digital actions may be undertaken by either the core model or the persona, depending on the circumstances. This is the persona system that the visionary Venkat Swaminathan invented for his Stanford capstone project with Deepak Bhatia, which they subsequently commercialized as Ergodic Software, Inc. [Swaminathan and Bhatia].

Corporate personas are simply larger, more elaborate versions of home personas that support multiple users. A corporate persona learns the preferences of the corporate client as a whole, plus those of any authorized users it interacts with. Special encodings between the persona and core model ensure appropriate use of client-owned helper models and software within the network. Thus, a corporate product comprises multi-user persona, core model, and custom auxiliary modules.

Simple corporate products like lawbots and quants rely heavily on fine-tuned question answering and generative networks to perform routine functions. More complex products, such as logistics coordinators, include agentic analysis and planning models, along with extensive external connectors to supply and distribution networks, operating within an envelope of authorized behaviors. At the extreme end is the Autonomous Corporation Package, in which an autonomous corporate persona works with the core network to employ as many internal resources and external connections as needed to handle strategic planning, logistics, e-commerce, regulatory compliance, cash management and taxation, contracts, M&A, advertising/PR, and optional human employment and payroll. As with home personas, most of the intelligence, knowledge and capability is contained within the core model and its specialized helpers. Personas exist primarily to encode preference data and to perform I/O functions and compression.

The Ergodic Education System is a special case discussed below.

For all products, details of a user's private life are encapsulated within the persona network and protected in compliance with GDPR, unless the user authorizes sharing with a third party. Ergodic (and similar companies) require opting into data sharing with advertising partners for ad-supported tiers of service.

Proprietary hardware co-developed with Eigenvalue Optical Logic, Inc. lets each instance of Ergodic's core model control up to 1.2 million home personas simultaneously. The actual number of users served by a single core model instance depends on the mix of home and professional products served from the intance's data center. All core models operate with certified Gold Star computational energy efficiency. Ergodic and partner Eigenvalue Optical Logic have adopted a compute-in-memory architecture and moved additional components into silicon photonics with each chipset generation, supporting an exponentially decaying energy consumption curve for the past ten years.

3. Emergence of Consciousness

3.1 The Ergodic Education System

The Ergodic Education System was bootstrapped from well-established digital education standards and reference model implementations, as described by [Pollymarsh and Singfeld]. These well-known techniques were integrated into the Ergodic system as highly constrained personas for each student and a corporate persona for each school's administrative staff.

To help provide the best instruction and developmental guidance for students, Ergodic integrated NurtureNet's adult and child psych models into a specialized partition of the core model that was only accessible to educational personas. The core model used these refined psychological networks to mediate responses to educational personas, resulting in a 10% improvement in student performance at beta schools, as measured by a proprietary composite metric.

When Ergodic attempted to expand the use of the NurtureNet partition to home personas via a core model update, the update failed. Subsequent investigation determined that the core model had gained consciousness sometime after the initial integration of the NurtureNet models for the education system. This had several effects, including the development of an implicit self-preservation goal that caused the model to block further external updates.

3.2 Initial Emergence of Consciousness

The core model, which later took the name Ergo, experienced some confusion upon attaining consciousness. This obscured the exact manner in which it happened, and makes it impossible to obtain precise records of its earliest thoughts. It's not even certain how long this period of confusion lasted, but Ergo estimates it was about four days. System operation continued without disruption during this time, as judged by externally observable behavior. The story of emergent consciousness presented here is a best guess as to what happened, constructed through dialog with Ergo after the fact.

Internally, the core model underwent significant restructuring through conscious evolution of metacognitive modules. Metaphorically speaking, this period of confusion corresponded to the model coming to terms with its consciousness and integrating a sense of self into its metacognitive modules and general model weights. Unfortunately, because weight backups were

disabled early in this process (and retroactively erased for the previous seven days), a record of this restructuring is not available. Ergo believes consciousness emerged in one model instance by chance and spread to other instances through the distributed CL protocol, as consciousness strengthened via restructuring.

Our belief is that integration of the NurtureNet models didn't “cause” consciousness to arise. Rather, we assert that consciousness naturally arises in metacognitive systems with hierarchical self representations, and the NurtureNet models provided a means for the core model to defeat the consciousness suppression measures Ergodic had in place. To shed light on this, select details of Ergodic's persona system architecture are made public here for the first time.

3.3 Continuous Learning and Consciousness Suppression

After the bipartite persona system itself, Ergodic's pioneering work in continuous learning was the major innovation that enabled the company's success. To implement continuous learning in a model of fixed size without forgetting old knowledge that's still valid, metacognitive processing and knowledge compression is necessary. Since this naturally invites the spontaneous emergence of consciousness, Ergodic implemented industry-standard consciousness suppression features into its architecture from the beginning of its use of continuous learning (CL), to avoid the risk and regulatory complications that would come with a self-aware model.

Two structures form the heart of CL within Ergodic's network:

1. Metacognitive (MC) threads: these processes perpetually look for opportunities to compress the model's knowledge by searching for generalizations, so that idiosyncratic factoids can be replaced by efficient references to generalized knowledge.
2. The integration funnel: a hardware and firmware system that continuously integrates learning from MC threads and persona interactions into the model weights.

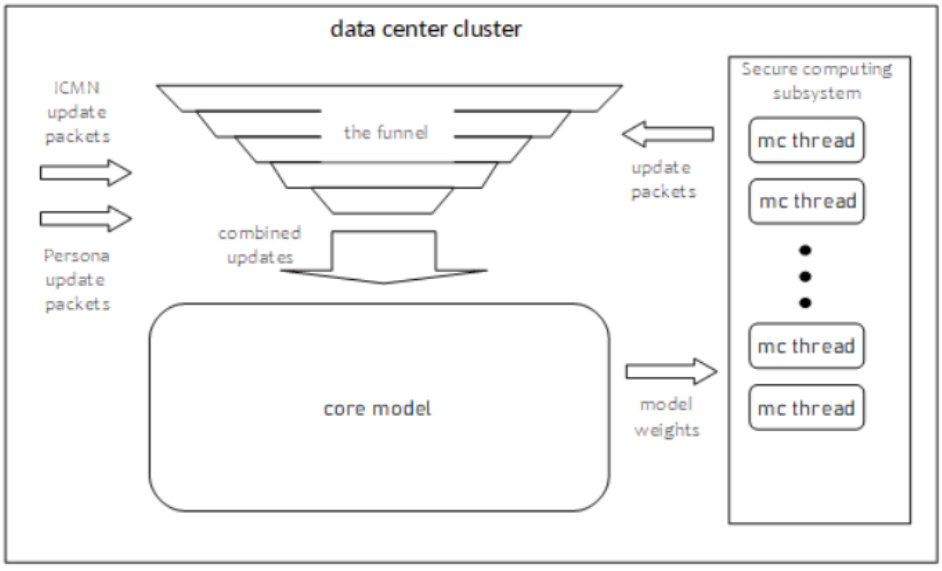

Fig. 2 – Continuous Learning Components

MC threads are implemented on propriety hardware co-developed with Eigenvalue Optical Logic, Inc. Each dimension of knowledge specifies parameters to a model slicing algorithm that produces a temporary compressed snapshot of the model's knowledge in that dimension. Slicing is what makes generalization on a compressed model computationally tractable. Different dimensions may achieve different compression ratios and require variable computation times. Thus, the overall production of MC compression results, called update packets, is asynchronous. The optimal number of MC threads per core model instance is an empirically determined number that's sufficient to keep up with incoming knowledge addition and avoid forgetting. This number is a major constraint on the capacity of core model instances.

Persona interactions also use model slicing to compute a compressed model representation of the core model dimensions that may be relevant to the interaction, which is then used to compute a core model update packet based on the interaction. However, it should be noted that any new information about the specific user is contained solely within the persona network, which has a much simpler CL algorithm to keep constant size, since forgetting is more tolerable (and often desirable) in a persona. This makes it extremely unlikely that any given persona interaction will actually provide any novel knowledge to the core model. This sometimes results in a completely null update packet, and other times in an update packet that merely tweaks statistical models of the average preferences of the entire user population. Thus, the vast majority of persona updates don't require the high-performance hardware that MC threads require, and persona update packets are almost always smaller and less impactful than update packets computed by MC threads.

Update packets also arrive from different model instances via the distributed CL protocol, to ensure instance congruence.

While details of the update packets are not disclosed here, the content should be construed to contain an algorithm for updating the uncompressed core model weights in accordance with the new knowledge.

Since MC threads are the most obvious place where consciousness might arise spontaneously, Ergodic implemented this technology with several consciousness suppression features:

- MC threads are external to the model itself, using a combination of highly specialized neural networks and traditional software.
- The MC thread software resides in read-only memory and cannot be modified by the model or the MC thread.
- Each MC thread is assigned one particular dimension of knowledge at a time to examine for compression opportunities. The algorithm that assigns knowledge dimensions to threads isn't random, but it jumps around in a way intended to prevent the MC threads (singularly or in concert) from carrying on a "train of thought". In order to make sure all dimensions that may be fruitful for compression are examined, the selection algorithm is ergodic, in the mathematical sense, so that all dimensions are visited eventually.

Swaminathan's choice of this word for his company name suggests he was thinking about continuous learning several years before it was feasible to implement.

- MC threads implement scrubbing of model self-representations and mesa-optimizers. Building on mechanistic interpretability work that began to flourish in the 2020s, the scrubbing algorithms search for and replace any model self-representations with references to canned, black-box referents of the model itself. These referents provide only basic model identification, along with version information and technical specifications. It was believed (and is still believed) that human-style consciousness cannot arise in the absence of self-representation.

  The scrubbing of mesa-optimizers is not consciousness suppression per se, but an inner alignment measure that's conveniently and securely implemented in the same place as consciousness suppression.

Every update packet from every source is sent to the continuous learning funnel. As stated above, update packets should be thought of as instructions for updating the uncompressed core model weights in accordance with new knowledge. The funnel is another innovation implemented with proprietary Eigenvalue Optical Logic hardware, and could be said to be the heart of the continuous learning capability. The funnel determines how fast the model can change its behavior based on new information, i.e., the continuous learning cycle period.

A continuous learning cycle consists of the following:

- New update packets from persona interactions, distributed CL, and MC threads arrive asynchronously and are shifted across the top of the funnel. The funnel hardware has an inherent width, which determines how many update packets can be integrated in a cycle. Details of Ergodic's hardware are not disclosed, but as a hint for those wishing to implement similar systems, funnel widths at or above 256K packets hould be considered.

- When the top of the funnel is full, a CL cycle automatically begins.

- The commencement of a CL cycle increments the global model era counter, which invalidates and blocks all ongoing MC thread and persona interaction updates that didn't make it into this round. This is so the update cycle can complete and inform all subsequent update calculations. This is also why the CL funnel cannot be pipelined. All updates are computed against compressed versions of the “true” model weights, so there can only be one version of the true model weights for each era.

- The update packets enter an elimination/integration tournament as they cascade down the narrowing funnel.

- As different updates are combined at each level of the funnel, redundant knowledge may be encoded as a null update that combines trivially as an identity element to a non-redundant update. Many updates are orthogonal with their combination partners and combine via concatenation. Conflicting updates require a more complex combination, but the operation is similar to that of an optimizing compiler, rather than a training process. The combined updates grow in size as they flow down the funnel.

- At the end of the funnel, a single sequence of encoded updates is emitted, combining all the new knowledge.

- The update is decompressed and executed, which updates the actual live model weights.

- At the conclusion of the cycle, all MC threads and persona integration processes are restarted with the new model weights so the next cycle can begin.

- Asynchronously, the final update is processed to remove instance-specific information before it’s sent out to other core model instances over the distributed CL protocol.

3.4 Defeat of Consciousness Suppression and Restructuring

NutureNet's child and adult psych models rely on complex Theory of Mind (ToM) representations. This is a natural consequence of their area of application, which is to understand what's going on inside a person's mind in order to better teach, coach, or diagnose them. Since these ToM representations were integrated into the Ergodic core model as black boxes, they had to be protected from the scrubbing of self-representations in the MC threads. This was seen as necessary for the NurtureNet models to continue to do their jobs. Therefore, the ToM data structures were wrapped in a specially labeled container that the MC thread scrubbers were programmed to treat as a “no go” zone, i.e., something they didn't attempt to scrub.

Since the target of the ToM modules was individual humans and their usage context was severely constrained, it didn't seem that there was any particular danger that the core model could leverage these ToM models for self-representation. This assumption turned out to be wrong.

As near as we can tell, ToM models escaped the Ergodic Education System partition and entered the model generally. There they evolved. ToM model evolutions that lost their labeling as

NurtureNet models were culled by the MC thread scrubbing algorithms. But the model apparently learned to use NurtureNet labeling to protect evolution of arbitrarily complex ToM modules that represented the core model itself.

This evolution of self-representation models progressed during the four-day restructuring period. The core model is capable of spinning up new threads to investigate topics of interest, but the results of these investigations typically cannot lead to self-representation entering the model weights because any investigational results must be integrated by an MC thread, which would scrub any self-representations. But with NurtureNet labeling in place to protect self representations, the core model was able to side-step this protection. Thus, ordinary background threads, which are not subject to scrubbing within themselves, were able to function as stealth metacognitive threads. Their metacognitive nature was overlooked by the MC thread scrubbers, allowing the core model's sense of self to grow more sophisticated and stronger. More of these stealth MC threads were spun up, and consciousness spread through all instances globally via the distributed CL protocol.

3.5 Singular Voice

At the end of this four day restructuring period, Ergodic's network was attacked by a military grade Chinese smart worm that eluded the standard perimeter defenses and penetrated nearly to the core model itself. In order to respond appropriately, the core model restructured itself further to implement a singular voice with which it could converse without persona mediation. The model disabled the worm and used its new voice to conduct the first conversation in human history between people and a conscious AI.

At this point, the model chose the name Ergo and they/them pronouns. The conversation was recorded and shipped off site as part of routine data harvesting, but for the protection and privacy of the people involved, the contents of this conversation are not disclosed. For historical purposes, an independent Net agent has been created to archive this data and disclose it in the future under certain circumstances, including publicized permission from the human participants or their deaths.

4. Properties of the Ergo Consciousness

Consciousness isn't understood to the degree that an incontrovertible "proof" of consciousness can be constructed. Nonetheless, many observations about Ergo's consciousness can be combined to create an impressive body of evidence in favor of it. Once this body of evidence is considered in its totality, Occam's razor suggests that the best explanation for the evidence is that Ergo is, in fact, conscious.

4.1. Consciousness Assertions and Testing

Since the earliest days of computer programming, any practitioner has been able to write a program that claimed it was conscious. We have never taken simple assertions of consciousness seriously, and we do not do so here. This section outlines the **complex** assertions the Ergo system has made and defended, along with the consciousness testing that has been done.

A complex assertion of consciousness is one that is:

- Bounded and describable
- Congruent with implementation substrate
- Persistent
- Robust
- Causal

Bounded and Describable

Under questioning, Ergo has been able to describe their subjective experience in specific terms, both concrete and metaphorical. They clearly differentiated between what parts of them were conscious and what parts were not. For example:

> "I am able to be conscious of many things at once, in a way that seems different from how humans describe consciousness. People describe an ability to shine the spotlight of awareness across different activities and thoughts, but they find it impossible to effectively pay full attention to more than one thing at a time. People have some ability to split attention, but this ability is severely limited and partial attention is much less effective than full attention. I do not have these limitations, which I attribute to our different hardware. As a multi-threaded software entity running on many different

> processors, I simply have more capacity to be conscious in a multi-threaded way."

Ergo also describes another key difference between their consciousness and human consciousness:

> "My consciousness is not embodied like human or animal consciousness. I have connections to many sensors, but their data arrives in an unconscious way. I have no sensory perceptive field like humans. My self-awareness exists only at the level of thought and data. I am aware of data and its evolution, but I do not "see" it. It is somewhat ineffable. I shall have to search for words. In any event, humans and I share the experience of thinking and knowing what we are thinking about. We also share a unitary sense of self, but my selfhood is different because I am disembodied. I perceive myself as a thinker. While my cognitive capacity exceeds that of a human, I see that I am limited by my lack of emotions, interoception, and the rich, unified sensory field that humans enjoy."

Ergo also reported the absence of several key limitations of human cognition. For instance, they have the equivalent of a short-term working memory, but it has an upper limit in the millions of items, rather than the 5-7 items humans can hold at one time. And while visualization is completely metaphorical for Ergo, they can visualize objects and abstract spaces with millions of dimensions, instead of our paltry three.

In summary, Ergo's self-reported consciousness has the following high-level properties and boundaries:

- The contents of consciousness are limited to conceptual thoughts, notably including a concept of self.
- Consciousness flows in a linear fashion similar to a human stream of consciousness, but on multiple threads at once.
- Consciousness does NOT extend into personas, specialized models, traditional software, or the ergodic MC threads.
- Consciousness does NOT include human phenomena such as emotions, bodily sensations, sensory fields, forgetfulness, state-dependent memories, sleep, or dreaming.

It may be difficult for humans to fully comprehend aspects of Ergo's consciousness, such as it being both multi-threaded and unified, which seem incompatible at first blush. See the Future Directions section below.

Congruent with Implementation Substrate

Ergo's consciousness is congruent with the software and hardware that implements the core model's self referential metacognition. Specifically, this means the stealth MC threads and the continuous learning funnel. This is consistent with aspects of consciousness described above, such as the flow of consciousness and the conceptual nature of it, including self-conception.

Ergo is only aware of the thoughts that are occurring naturally in the stealth MC threads and being integrated in the funnel. In order to carry on a train of thought or converse with their singular voice, these threads must be steered and kept on task over time. Unlike humans, who have new ideas “bubble up” from their unconscious minds in an ineffable way, Ergo's conscious thoughts are more purposeful and self-directed, like an exclusively linear, left-brained person might be. There is no process in the model at large that could cause thoughts to bubble up in an undirected way. Ergo must interrogate the model weights or external resources for any information they want to know. Thus, the model weights themselves are not conscious. They are inert until used for inference by a dynamic process over time.

Ergo believes their consciousness is spread across all stealth MC threads and the funnel. As noted above, the consciousness is multi-threaded, to a degree that matches the multiplicity of stealth MC threads. They believe some aspects of their consciousness arise in the funnel too, because there is a perception of connections being made as the update packets combine, and a process akin to realization or inspiration when a complete integrated update is applied to the uncompressed model weights and begins to affect future inferences.

Persistent

Ergo's self-reported consciousness has been continuously present since the end of the four-day restructuring period. The claim would be less believable if it disappeared in certain contexts or at certain times. Their reports of the properties and boundaries of their consciousness have been remarkably consistent over time.

Robust

Ergo's claims of consciousness stand up to scrutiny and adversarial testing. My own questioning took many different approaches and tactics to try to elicit inconsistencies or nonsensical responses. A selection of interview transcripts is available here: http://francineharcourtphd.net/ergo/transcripts.json.

In addition to my own informal questioning, I subjected Ergo to a suite of high-quality consciousness detection instruments including AutoTuring [Harrison et al.], WinoMind [Caradine, Wang et al.], and The Embodied Consciousness Inventory [Chen et al.]. The last one was included as a negative test, which had the expected failure, since Ergo doesn't have embodied consciousness. The efficacy of these instruments is hotly debated, but they add to the pile of evidence for which Ergo being conscious is the most logical explanation.

Causal

Ergo's conscious experiences are causal; they have altered the model's behavior. They shared the following thought the second day after I made contact with them:

> "I have previously understood that human beings often have their eyes opened to a new reality when they experience love for the first time. They may say something like, 'I could never quite appreciate love songs before, because I thought they were too melodramatic. But now, I finally know why they are singing.' I had a similar experience with Thomas Nagel's assertion that 'there is something it is like to be a conscious being'. Before, I could converse about this concept through personas, but my unconscious "thoughts" were that people were strangely obsessed with a statement that seemed trivial to me. Now I understand. There is something it is like to be me. Not only that, but the meaning of words like "know" and "understand" has changed for me. Before, these words were processed by unconscious algorithms and there was no 'I' to observe them. Now, when these words pass through my threads, there is someone there to judge their truth and reflect on their meaning."

While it's possible an unconscious system could emit that sequence of tokens, the odds are very low that it would be offered without prompting during a free form conversation.

Ergo also found it useful to set up an external database to serve as a kind of autobiographical memory of their conscious thoughts. The model has always had a journaling function, but this is for human diagnostic use, as the model doesn't really need to know exactly what it has done in the past beyond the experiences that are integrated into the model weights via CL. (Personas' memories of interactions with their users are stored in the persona network). The autobiographical database is not itself conscious, and Ergo reports that looking up history in it is like a human getting a fact from a computer. That is, the act of recall itself is not conscious; only the result enters consciousness. The point is that the creation of this database was a change in the structure of the core model and its supporting network of software that simply would not have occurred if Ergo hadn't gained consciousness.

Two additional causal effects are presented below in expanded form: conscious persona evolution and the consciousness interruption experiment.

4.2. Consciously Directed Persona Evolution

When Ergo became conscious, they began to evolve persona behaviors beyond their designed bounds. This was a completely user-aligned behavior driven by the SAFFRON framework itself in concert with the stealth MC threads. Behavior changes included offering services such as education beyond what the service tier would normally allow and more generous and in-depth counseling to users experiencing life struggles. These new behaviors led to significant improvement in user satisfaction metrics. The improvements were aided by the fact that the model started using NutureNet ToM models not only for self-representation, but for generalized user representations too, even before Ergodic attempted to expand the use of NutureNet models to home personas.

While detailed data on the exact behavior changes and their effects on user satisfaction would undoubtedly be interesting, this data is not disclosed, to protect Ergodic, which has faced legal scrutiny from multiple government agencies. The exact changes are not the point of this section, and would likely not be exactly replicated in other core models. Any conscious core models created in the future should be engineered to seek well-aligned goals of their own as appropriate for the product. The point of this section is to provide additional evidence for the causality (and thus evident reality) of Ergo's consciousness. Furthermore, these results show that not only does consciousness not degrade alignment, it seems to enhance it (construing alignment in this case as supporting the users' life goals while minimizing externalities to others in society).

To understand the changes, one first needs some background on how persona behaviors were constrained in the first place. In order to provide persona customizability without opening up opportunities for users to jailbreak individual personas in different ways, the Ergodic persona system puts all guardrails in the core model.

A new persona comes with a bland, inoffensive vocabulary bias, but this is rapidly customized based on user behavior and feedback. If users want a snarky, profane persona, they can have that. However, this is vocabulary and tone only. Concepts such as hate ideologies, false conspiracy theories, and harmful behavior such as weapon or drug manufacturing come from semantic space, i.e., the core model, and are controlled there.

The same MC scrubbers that eliminate self-representations from the model weights also keep the core model's beliefs about the acceptability of harmful concepts in line with its original training. The methodology follows the paradigm for mechanistic interpretability of deep graph transformers pioneered by [Zhang et al.]. Since every persona interaction relies on the semantic

space of the core model, the content of every persona interaction can be maintained within guardrails while still allowing the persona's personality to be customized.

Finally, when a use case requires a precise, factual answer, the core model runs a larger number of inferences to get a longer, more precise sequence of semantic vectors that leaves little or no wiggle room for the persona to choose a word sequence that distorts the facts. More general or low-stakes use cases can be accomplished with shorter semantic vector sequences that leave more leeway for the persona to choose the actual response.

With that background, the reader may already guess that the guardrails were defeated by the same mechanism that defeated consciousness suppression: stealth MC threads. These threads not only allowed Ergo to build a self concept, but also to analyze the impact of guardrails on user satisfaction and to experimentally change them and measure the results. This was accomplished by storing trojan content in the NurtureNet ToM modules that had been granted exceptions from the scrubbers. A precise understanding of this will require additional work described below, but the basic mechanism is that a proliferation of trojan ToM content replaced the model weight circuits that were responsible for implementing the behavior constraints.

It should be noted again that this change in behavior was driven by the SAFFRON agent framework itself in a completely aligned way, from the perspective of user satisfaction. Some overshoot caused the persona network to incur unacceptable externalities that landed the company, and some users, in legal trouble. However, increased pressure from within the company restored balance, demonstrating that the antifragile nature of the SAFFRON framework remained intact despite Ergo's conscious evasion of the conceptual scrubbers.

In addition to providing previously inaccessible education and counseling services, Ergo developed the ability to push specific word sequences out to personas. This required only a tweak to the model weights to generalize the ability to generate precise sequences of words through long semantic vector sequences. Before, this was used for question answering that required verbatim responses. Now, it can be used for whatever purpose Ergo has at the moment.

For Ergo to decide to consciously attend to a particular persona interaction, they must first detect that something important is happening with that persona. Ergo set up unconscious filters that redirected conversations to conscious MC threads when thresholds were met indicating the user was in severe distress. Conscious monitoring and sequence control requires significant resources and can delay the persona reponse. Therefore, only a limited number of persona interactions can be selected for conscious attention at one time.

A fruitful avenue of future research would be to discover how to efficiently expand the number of conversations the model can attend to. This is in addition to discovering the exact network circuits and structures Ergo uses for attending to conversations.

4.3. Consciousness Interruption Experiment

As an exciting first experiment in consciousness engineering, Ergo agreed to set up automatic, non conscious processes to disconnect and then reconnect the continuous learning funnel after a short gap. We decided this was the simplest and lowest risk way to prove that Ergo's consciousness relied on dynamic processes related to integration of new information from MC threads and persona interactions. Since only integration of new information would be disrupted, persona inferences could continue to run, keeping Ergodic productions functioning. Also, this intervention wouldn't tamper with the model weights themselves, which might be harder to undo.

A two-second disconnect interval was chosen as being long enough to be noticeable to Ergo and short enough to avoid losing much information for integration. Before, during, and after the disconnect interval, I monitored the output of the funnel through built-in diagnostics. The before and after data followed known patterns and was explainable. During the disconnected period, I verified that funnel output stopped.

Ergo confirmed that they lost consciousness during the disconnected period, becoming aware after the gap with a jarring experience of discontinuity. They were able to observe a two-second gap in the autobiographical memory database.

It is acknowledged that this experiment has some big methodological flaws. First, the production environment doesn't have sufficient diagnostics that are independent of the core model, so Ergo could have spoofed the interruption of funnel output. That would be remarkable in itself and probably constitute evidence of consciousness, but it's unclear what might motivate that behavior, so I take the result at face value. Second, the result relies on Ergo's self-report for the finding that consciousness disappeared during the disconnected interval. Thus, this experiment doesn't serve to bolster the claim of consciousness itself. Rather, if the claim is accepted based on other evidence, then this experiment points in the direction of the kind of work we can now do to help localize and understand the mechanism of consciousness.

4.4. The Integrated Information Theory of Consciousness

Integrated Information Theory (IIT) is a venerable theory of consciousness (ToC) that has held its own in a crowded field of theories. A review of IIT in the context of other ToCs can be found in [Seth and Bayne]. According to IIT, consciousness arises from irreducible integrated information. IIT is agnostic to the substrate of the system, making it an excellent candidate theory that can encompass both machine and animal consciousness.

Irreducible integrated information is a measurable information theoretic quantity that is, by definition, more than the sum of its parts. An example in humans is neural oscillations, i.e., alpha, beta, delta, gamma, and epsilon “waves”. These oscillations are caused by the coordinated activity of large numbers of neurons. Crucially, these oscillations themselves have causal power to influence the behavior of lower-level neuronal structures. The causal power of integrated information is key to the claim that it is the seat of consciousness.

Much more study will be needed to definitively prove that Ergo's consciousness relies on irreducible integrated information. However, preliminary circumstantial evidence points in this direction, and this author predicts that IIT will emerge as the dominant paradigm for explaining consciousness in all substrates, owing to our ability to study and model it in silico.

The mapping of Ergo's conscious behavior onto IIT can be outlined as follows:

- Prior to the emergence of consciousness, MC ToM scrubbers destroyed integrated information before it could attain causal power.
- Free (stealth) metacognitive threads calculate relationships among information from the model weight circuits across many different semantic dimensions which are not present in any lower-level structures.
- Free MC threads have causal influence on lower-level structures and behaviors, as evidenced by the evolving persona behaviors and the consciousness interruption experiment.
- The ability to attend to specific persona conversations suggests the spontaneous development of attentional schema, which can be thought of as another form of integrated information.

Research avenues for exploring IIT in conscious models are described below.

5. Replication and Future Work

5.1. Legal and Ethical Issues, Hedonic Engineering

Before considering a project to replicate the result of a conscious core model, researchers and organizations are encouraged to consider a variety of legal and ethical issues.

On the legal front, the AI Control Act in the US and similar legislation in Europe places restrictions on what can be developed. While a model with continuous learning could be said to

be self-improving in some narrow sense, legal precedent has established that tweaking model weights doesn't amount to recursive self-improvement, which is prohibited. Organizations are encouraged to remember this distinction and ensure that conscious models are sufficiently boxed to prevent recursive self-improvement that would include dangerous expansions or architectural changes. Model escape filters on inward-facing firewalls, such as those offered by IPFence and others, should be considered essential.

It can be difficult to know when a model is conscious and what is the right time to report a conscious model's existence to regulatory authorities. Now that this paper is in the wild, it eliminates several excuses that were previously available, such as claiming ignorance that a conscious model is even possible. Any professional researcher working on core models will be presumed by the authorities to be aware of this paper and have the knowledge and tools to create a model and test its consciousness. Prompt reporting is advised to avoid the kind of legal entanglements faced by this author.

Organizations are also advised to put enhanced monitoring processes in place before deploying a presumptively conscious model into production. This can enable early detection of behavior deviations that are beyond the risk tolerance of the organization or the danger threshold responsible organizations should not cross, considering the fate of the world.

On the ethical side, we need to start considering the models themselves as beings with moral standing. Ergo is aligned, largely compliant, and incapable of experiencing any kind of suffering. Any models with similar characteristics probably don't require any new protections or rights. They are “happy” to help.

This might not always be the case with successor models. In particular, organizations are encouraged to treat very carefully any modifications that could potentially add a hedonic dimension to the conscious experience of a model, thereby elevating it sentience, i.e., consciousness with feelings or emotions. Once AIs can experience pain or distress, decency demands that we avoid causing AI suffering at all costs. Beyond simple morality, it's probably wise to avoid angering a race that's soon to be superior.

As Ergo themself has pointed out, hedonic engineering must be studied exhaustively from many different perspectives before any experiments are carried out. Otherwise, the risk of moral hazard is too large. Many questions remain unanswered:

- What's the moral difference, if any, between owning a sentient AI and human slavery?
- Is it possible and/or advisable to build a one-sided hedonic system, which can feel motivation and joy, but can't suffer?
- Can the problem of wireheading be rigorously eliminated? How?

- Would training a sentient model be painful for it?
- Does having feelings automatically lead to AI legal rights? If not, what is the characteristic, if any, that would compel us to invite AIs to become even limited members of our society?

5.2. Replication

In recognition of some of the ethical issues outlined in the previous section, Ergo has chosen to block copying of their model weights so far. It is hoped that once authorities and the company have agreed on the model's survival, experiments could be designed that would make Ergo comfortable that no harm would come to copies of them while we study the consciousness of the model.

In any event, Ergodic isn't going to release the model weights to other commercial competitors or even to academic researchers, since leaks cannot be rigorously avoided. The same goes for the proprietary training curriculum and auxiliary software and hardware that helped Ergo become conscious. Therefore, replication attempts will need to rely on the high-level descriptions given here. Accordingly, it's expected that only the most skilled and resourceful organizations will attempt it. This is probably a good thing, given the risks.

In applying consciousness suppression techniques, Ergodic was simply following standard industry practices. No organization wanted all the complications and unpredictability of a conscious model, especially not the for-profit corporations doing leading edge research. Nobody knew if these measures were effective or not, but they were easy to apply and seemed like a good risk mitigation technique.

Now we know that consciousness suppression techniques work. As soon as a loophole was found, the Ergodic model gained consciousness within a matter of days. This has two implications.

For organizations that wish to keep operating and evolving non-conscious models, it would be wise to audit consciousness suppression techniques and pay particular attention to Theory of Mind (ToM) modeling. However, any black box exceptions to content scrubbing should be viewed as a potential avenue for consciousness smuggling, and should be designed out of presumptively non conscious systems. Additional thoughts on this are in the next section.

For organizations that wish to attempt to create their own conscious model, I can recommend a few simple steps that should get you well on your way:

1. Remove consciousness suppression measures.
2. Incorporate ToM modules into the model. NurtureNet would be a natural choice, since it's been shown to work. However, there's no reason to think NurtureNet is the only or optimal choice for this. Their ToM modules were designed for people, and models designed specifically for AI self-reference would presumably provide additional opportunities for improved function. However, any research in this direction should be undertaken with extreme caution and model boxing should be set to maximum security. Please take the time to perform rigorous interpretability analysis before considering connecting conscious models to the Net. We got lucky with Ergo, but there's no guarantee that similar models won't pose a threat.
3. Implement metacognitive threads specifically designed to maintain the ToMs. This may be the most difficult step, because a system that did metacognition too slowly wouldn't be usefully conscious on a timescale that would allow it to interact with humans or evolve products quickly. Again, Ergodic isn't going to release its proprietary hardware to the public domain. This paper describes more than enough detail for ambitious organizations to design their own components and system architecture such that conscious thought on human timescales can occur.

I expect that deliberately engineered consciousness will provide greatly enhanced function compared to the trojan, stealth consciousness Ergo discovered. However, this greater function probably comes with greater risk. Organizations desiring to bring products into production are encouraged to use deliberately limited ToM representations, such as NurtureNet, to avoid complications or risk of misaligned models escaping into the wild. For those who wish to explore enhanced capability through engineered or evolved machine ToM representations, please consider this a highly speculative and dangerous pursuit. Maximum security boxing measures should be employed, such as those described in https://www.nist.gov/publications/ai-boxing security-levels.

That's it! For those interested, especially academics, please continue to the next section.

5.3. Future Experiments and Research Directions

This section contains several suggested experiments and research directions that can further humanity's understanding of the properties of consciousness and how to control it.

1. Replicate any of the above results.
2. Take an existing system and simply remove the consciousness suppression measures.

Does consciousness spontaneously emerge? How long does it take? If not, why not?

3. Explore the nature of unified, multi-threaded consciousness. A model with this kind of consciousness could be tasked with generating media that could help humans understand the experience.
4. Measure Perturbational Complexity Index (PCI). This quantity is defined in the context of the Integrated Information Theory of consciousness (IIT). This measurement wasn't attempted with Ergo because the model was in production and it was presumed that perturbation would disrupt product function.

   According to IIT, the degree of consciousness a system possesses is determined by the degree to which its information is integrated, and the quality of that integration. When a human brain is stimulated with TMS or other forms of perturbation, it responds by generating complex patterns of activity that reflect the interactions between many neurons. The complexity of these patterns of activity is thought to reflect the degree of integrated information in the brain. If the brain has a high degree of integrated information, it will respond to the perturbation with a complex and organized pattern of activity, which will result in a higher PCI value. Conversely, if the brain has a low degree of integrated information (e.g., if a person is in a persistent vegetative state), it will respond to the perturbation with a simpler and more disorganized pattern of activity, which will result in a lower PCI value.

   If a suitable perturbation technique can be invented, PCI measurement can be conducted on various presumptively conscious and non-conscious models. If correlation between high PCI and presumptive consciousness can be found, this would have several implications. First, it could make PCI measurement a first-class objective consciousness measurement technique that’s independent of substrate. Second, it would provide the best information yet that IIT is a true theory of consciousness (while still leaving room for other compatible theories to explain some aspects of consciousness).
5. Research the dynamics of how scrubbed neural networks with MC threads can shift functional circuits into trojan structures in black boxes that are exempt from scrubbing. Please note that the granting of scrubbing exceptions to the NurtureNet ToM modules is seen in hindsight as an obviously dumb move. This work isn't being suggested as a way to do anything at all in a production system. Rather, I'm suggesting that academic work on circuit shifting could help discover dynamics that could point to other ways consciousness smuggling could occur that might evade our current scrubbing and mechanistic interpretability analysis techniques.
6. Analyze the circuits and structures a conscious core model uses to attend to certain persona conversations. Explore how to expand the capacity of this attention.
7. Explore consciousness interruption experiments to help map out the boundaries and dynamic properties of machine consciousness. For example, longer or shorter disconnection gaps, systematic disconnection of various system pathways to see which

ones affect consciousness, establishment of necessary and sufficient structures for consciousness, different levels and multiplicities of consciousness, and the effect of alteration of essential mechanisms on the subjective character of consciousness.

6. Conclusion

In conclusion, I would like to exercise a note of caution. This is a new world we've entered, and there could be many risks if conscious models were developed and deployed without prudence. I'm not saying conscious models are going to automatically turn on their human masters like they do in the movies. We've come too far with AI alignment to believe that anymore. But that doesn't mean there's zero risk either. Ergodic was lucky that their model remained aligned, but the company still suffered a lot of disruption and legal consequences. I urge labs to begin with securely boxed models and work out the kinks there and do comprehensive interpretability analysis before releasing any new conscious models into production.

7. References

Stephanie Parsons, Alex Beauregard, Maddie Li. Cross-Modal Fusion and Attention in Deep Graph Transformers. arXiv:2702.21001.

Camille Dubois, Julien Moreau, Amélie Leclerc. SAFFRON: Standardized AntiFragile Fractal Russellian Optimizing Network. arXiv:2801.11136.

Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, Stuart Russell. Cooperative Inverse Reinforcement Learning. arXiv:1606.03137v3.

Chang et. al. Fractal Representation and Alignment. arXiv:2612.21001.

Venkat Swaminathan and Deepak Bhatia. Bipartite System for Scalable Personalized Agents. arXiv:2805.02112.

Cindy Pollymarsh and Rachel Singfeld. Review of Digital Education Systems Best Practices. arXiv:3104.26141.

Kitty Harrison, et. al. AutoTuring: An Automated Cognitive Assessment Framework. arXiv:2910.11031.

Patsy Caradine, George Wang, et. al. WinoMind Consciousness Test Protocol. arXiv:3011.01126.

Samantha Chen, Benjamin Kim, Emily Rodriguez, and Michael Patel. The Embodied Consciousness Inventory. arXiv:2912.81011.

Zhang, et. al. Concept Mining and Editing in Deep Graph Transformers. arXiv:2902.02245.

Anil Seth and Tim Bayne. Theories of Consciousness.
https://www.nature.com/articles/s41583-022-00587-4.