Artificial Intelligence is for Amateurs: SIML, Free Energy Principle and Cognogenics as the Foundations Toward the Birth of True Artificial Life

Richard Everts

April 2025

Abstract

We introduce *Cognogenics*, a new paradigm for artificial life and intelligence that arises not from predefined goals or supervised data, but from an agent's continuous drive to reduce internal prediction error. Built upon the Free Energy Principle (FEP), Cognogenics powers *Simulation-Integrated Multimodal-Language* (SIML) agents - embodied, memory-constrained organisms whose behavior emerges from real-time inference, not reward signals.

SIML departs sharply from traditional reinforcement learning (RL), large language models (LLMs), and evolutionary algorithms: RL requires extrinsic rewards, LLMs lack embodiment or agency, and evolutionary systems adapt only across generations. In contrast, SIML agents learn during a single lifetime, forming internal generative models and compact bitwise memory schemas in under 1kB of RAM for active memory, implemented in Rust for both safety and performance.

Empirically, SIML agents exhibit foraging, adaptive planning, path-dependent habits, and homeostasis-seeking behaviors-without any explicit instructions. These results demonstrate the emergence of goal-like behaviors and survival strategies solely through free energy minimization. This work establishes Cognogenics as a viable foundation for general-purpose, self-organizing intelligence for the creation of artificial life.

1 Introduction

Artificial intelligence research has advanced rapidly through reinforcement learning (RL), large language models (LLMs), and evolutionary algorithms. However, each paradigm is built on assumptions that constrain its capacity for open-ended intelligence. RL agents must be given explicit, often brittle, reward signals—leading to behavior that optimizes game scores or task-specific metrics without generalization. LLMs, while powerful predictors of human language, are passive entities with symbolic representations (Dreyfus, 1992): they cannot act, perceive, or test their predictions in the real world. Evolutionary approaches, as used in artificial life, evolve agent populations over generations but rely on predefined fitness functions and are often computationally inefficient.

Consider, for example, an RL agent in a game that learns to exploit reward glitches rather than mastering the task, or a language model that completes prompts convincingly but has no understanding of causality or physical constraints. These systems lack what biological organisms display: the ability to adapt in real-time, based on internal needs and environmental changes, without being told what to want.

We propose *Cognogenics* as a new field that explores how cognitive structures and behavior can emerge within a single lifetime from first principles. Cognogenics is grounded in the Free Energy Principle (FEP), a theoretical framework from neuroscience that states: *a self-organizing system must minimize the long-term average of surprise (i.e., prediction error) to remain viable* (Friston et al., 2006; Friston, 2009). In this framework, action and perception both serve the same goal: reducing the discrepancy between expected and observed inputs.

This perspective aligns with long-standing insights in embodied cognition, where cognition is viewed not as abstract reasoning over internal representations, but as tightly coupled with the agent's body and environment (Brooks, 1991; Chemero, 2009; Clark, 1997). Rodney Brooks famously argued that intelligence emerges not from central control but from decentralized, reactive interactions with the world—"the world is its own best model" (Brooks, 1991). Similarly, Tony Chemero has emphasized the enactive, dynamical nature of cognition—rejecting the idea that cognition resides in disembodied computation, and instead locating it in the real-time engagement of an organism with its surroundings (Chemero, 2009; Suchman, 1987).

Crucially, the FEP does not require an external reward function. Instead, agents form internal generative models and act to maintain consistency with those models. For instance, an agent with low energy expects to encounter food; if it doesn't, it experiences high surprise and acts to resolve it. Homeostasis, curiosity, and goal-seeking behavior all emerge as side effects of minimizing free energy. This reframing unifies perception, cognition, and action under one mathematical imperative.

To investigate this principle, we introduce the *Simulation-Integrated Modeling Loop* (SIML), a novel cognitive architecture situated in a 3D voxel environment. SIML agents are not evolved, pretrained, or reinforced. They operate entirely through bitwise schema memory and active inference, continuously predicting sensory states, comparing them to reality, and adapting to reduce the difference.

Key Contributions

- **A Free-Energy-Based Agent Architecture:** We introduce a generative agent that adapts in real-time via free energy minimization, without rewards, evolution, or supervised training.
- **Bitwise Schema Memory:** We implement a novel memory architecture using compressed bitfields (under 1kB total active agent memory, 219kB total agent embodied memory) to encode prior experiences, enabling fast matching, low entropy, and generalization across similar contexts.

- **Real-Time, Embodied Cognition in 3D Environments:** SIML agents exist in a voxel world with spatial constraints, partial observability, and embodied action, highlighting how cognition is shaped by sensorimotor loops.
- **Emergent Behavior Without Instruction:** SIML agents exhibit foraging, detouring, habit loops, and homeostatic regulation with no rewards or goals specified—emerging entirely from internal modeling and surprise reduction.
- **Comparison and Path Forward:** We position Cognogenics as a general, scalable framework distinct from RL, LLMs, and evolution-based systems, and outline its potential trajectory from simple agents to posthuman cognition.

In summary, Cognogenics and SIML shift the paradigm from agents that optimize externally imposed objectives to agents that self-organize around internal expectations. The following sections present the SIML methods, mathematical formulation, and observed behaviors in simulation.

2 Methods

2.1 Architecture Overview

At the core of the SIML agent architecture is a novel active inference system that follows the *Free Energy Principle*. A high-level schematic of the architecture is shown in Figure 1. The agent consists of the following components, which together form a perception–cognition–action loop:

- **3D Voxel Environment:** The agent operates in a spatial world composed of discrete voxels. Each voxel encodes properties such as nutrient content, moisture, salinity, and obstacles. The environment is dynamic and changes in response to the agent's actions.
- **Sensory Input:** The agent receives chemosensory information from local voxels (e.g., moisture, pH, organic matter) and internal states (e.g., energy level). These inputs are encoded into fixed-size bit vectors and passed to the inference module.
- **Inference Module:** Upon receiving new input, the agent updates its beliefs about the world using variational free energy minimization. This module integrates current observations with prior beliefs to maintain an up-to-date internal model of the environment.
- **Bitwise Schema Memory:** The agent stores past experiences as compact, bit-encoded schemas that capture patterns of perception, action, and outcome. These schemas influence decision-making by providing priors for inference and guiding action selection.
- **Generative Model:** A probabilistic model of how states cause observations and how actions lead to state transitions. It allows the agent to simulate future outcomes and evaluate how well they align with preferred internal states, such as homeostasis or safety.
- **Action Simulation & Selection:** For each potential action, the agent simulates predicted consequences using its generative model and schema memory. It evaluates actions based on expected future surprise and selects the one that minimizes predicted error or deviation from preferred states.
- **Actuator Output:** The selected action is executed in the environment, modifying the world (e.g., movement, ingestion). These changes affect future sensory inputs and complete the perception–action loop.

- **Sleep Optimization:** Periodically, the agent enters a rest phase during which schema memory is reorganized. Redundant or low-value schemas are pruned, and high-utility schemas are consolidated to reduce memory entropy and improve future inference performance.

This closed-loop structure forms the Simulation-Integrated Modeling Loop (SIML), where the internal simulation drives external action, which then reshapes sensory input-resulting in continuous refinement of the internal model. Behavior arises not from policy optimization but from maintaining harmony between predictions and lived experience.

Figure 1: **The SIML Agent Architecture.** At each timestep, the agent processes sensory inputs from a 3D voxel environment and updates its beliefs via a variational free energy minimization process. A compact, bitwise schema memory supplies priors and constraints that influence inference and generative modeling. Candidate actions are simulated using internal models and evaluated by their expected free energy. The agent selects the action that best fulfills both epistemic and pragmatic value, and executes it via actuators in the world. Schema memory is periodically consolidated during sleep cycles based on surprise and utility.

2.2 3D Voxel Environment and Embodiment

To evaluate SIML agents under realistic constraints, we designed a 3D voxel-based world. The environment is a discrete grid of cubic cells ("voxels"), each capable of storing material types such as empty space, wall, or nutrient. Time proceeds in discrete ticks, and the agent occupies a single voxel at any moment.

Perception is egocentric and local: the agent senses chemosensory gradients (e.g., moisture, salinity, organic content) in the voxel beneath and ahead of it, as well as internal states such as energy. Critically, the agent never receives a global map-it must infer structure from fragments.

Embodiment Constraints. SIML agents are physically embodied. They move through space one voxel at a time, consume energy through motion and computation, and can only interact locally. This enforces partial observability and introduces trade-offs: the agent must choose whether to explore, conserve energy, or pursue specific locations.

These physical constraints make cognition nontrivial. In contrast to disembodied LLMs or game agents with global state access, SIML agents must reason about unseen regions and plan sequences of energy-efficient movements. Cognogenics thus aligns with theories of *embodied cognition*, where intelligence is shaped by the body's structure and environmental constraints (Pfeifer et al., 2007).

Environment Scenarios. We tested SIML agents in a range of configurations:

- **Open Field with Nutrients:** Nutrients are randomly scattered. The agent must wander, learn predictive patterns, and return to nutrient-rich areas to survive.
- **Obstacles and Maze:** Walls force the agent to detour or remember alternate paths. Schema memory is challenged to store successful plans.
- **Multiple Agents (Preliminary):** We briefly tested dual-agent scenarios to explore emergent social behaviors, though multi-agent cognition is left for future work.

In all cases, the voxel world supports exploratory learning, action loops, and long-term survival without predefined goals-allowing emergent cognition to manifest.

2.3 Bitwise Schema Memory Structure

A key innovation of SIML is its lightweight schema-based memory architecture. Instead of real-valued embeddings or backprop-trained weights, experiences are encoded as binary schemas of fixed size-compact, interpretable, and efficient.

Memory Encoding Each `Memory Struct` represents a conjunction of features as a bitfield ranging from 8-bits to 32-bits for our purposes here. The only limitation is the minimum needed values to represent information. These include perception features (e.g., local chemosense values), internal state (e.g., energy level), action taken, and resulting outcomes. For example:

> At t, the agent perceives: no wall ahead (feature A), low energy (feature B), takes action: move forward (feature C), and at $t+1$ perceives food (feature D). The schema encodes: A=1, B=1, C=1, D=1.

This allows the agent to store: "In context A+B, doing C led to D." These schemas are added to memory on each tick and are later scored and ranked during sleep cycles based on match success, energy deltas, and danger levels.

Memory Retrieval and Use Schema memory influences behavior in two critical ways:

- **Prior Elicitation:** The agent queries $\mathcal{M}$ for schemas matching its current bitfield and internal state. Matching schemas provide precedents that update the agent's generative model priors $P(s_{t+1} \mid s_t, a)$, biasing it toward historically successful or safe predictions.
- **Constraint Satisfaction:** Certain schemas encode homeostatic boundaries-e.g., "if energy = 0, agent dies." These schemas are tagged as dangerous and are used to filter action choices, effectively embedding safety constraints within the schema layer.

Emergent Effects of Memory Constraints The schema memory has bounded capacity-schemas cost energy to retain, and only the most useful are preserved through consolidation. As a result, the agent often:

- Loops actions due to overfit schemas.
- Fails to predict rare events due to schema loss.
- Forms habits (e.g., repeatedly taking a path to nutrient) even when other paths are available.

  These limitations are not bugs but features: they induce realistic behavioral signatures such as memory lapses, suboptimal foraging, or rigid habits-mirroring cognitive limitations in biological organisms.

SIML Worm Chemosense Bitfield Layout

`struct Chemosense(pub u32)`

Moisture	pH	Organic	Salinity	Nitrogen	Phosphorus	Potassium	Toxicity
4 bits	4 bits	4 bits	4 bits	4 bits	4 bits	4 bits	4 bits

Each field stores a value in the range **0–15**

Example: `00110100111010000010010101101100`

Channel	Bits	Decimal	Interpretation
Moisture	0011	3	Low moisture
pH	0100	4	Slightly acidic
Organic Matter	1110	14	Rich in organic material
Salinity	1000	8	Moderate salinity
Nitrogen	0010	2	Low nitrogen
Phosphorus	0101	5	Medium phosphorus
Potassium	0110	6	Medium potassium
Toxicity	1100	12	High toxicity

Figure 2: **Chemosense Bitfield Layout for SIML Worm Agent.** This 32-bit memory representation encodes the local chemical environment into eight 4-bit channels: Moisture, pH, Organic Matter, Salinity, Nitrogen, Phosphorus, Potassium, and Toxicity. Each channel stores a value from 0 to 15, allowing the worm to process its chemosensory landscape compactly and efficiently.

2.4 Sleep Optimization and Memory Consolidation

In SIML, sleep functions as a crucial optimization phase, directly analogous to sleep and dreaming in biological organisms. During this background phase, the agent systematically restructures its memory-encoded as bitfields-by pruning, reorganizing, and consolidating schemas to improve predictive precision and reduce entropy.

Let memory consist of encoded fragments:

$$M = \{m_i\}_{i=1}^{n}$$

The entropy associated with this memory structure is defined as:

$$S(M) = -\sum_{i=1}^{n} p(m_i) \log_2 p(m_i)$$

where $p(m_i)$ reflects the recent activation frequency, match quality, and predictive success of each memory pattern.

During sleep, schemas that are frequently used and yield low prediction error are retained and reorganized into contiguous memory blocks. This yields an optimized configuration with lower entropy:

$$S'(M) < S(M)$$

This restructuring improves lookup performance from $O(n)$ to near $O(\log n)$ via sorted, contiguous allocations-mirroring biological processes like synaptic pruning and memory consolidation. Sleep thereby enhances the agent's generative modeling capacity by refining the memory structures that bias its internal predictions.

From an implementation standpoint, the Rust-based system leverages low-level memory operations and pointer arithmetic to perform this optimization efficiently and safely. Rust's ownership model ensures that memory consolidation remains free of race conditions or undefined behavior, while its zero-cost abstractions support parallelizable reordering and scoring of schemas.

In essence, sleep serves not only to compress past experience into compact, predictive structures, but also to reorganize those structures into efficient, usable forms-enabling real-time cognition under strict resource constraints.

2.5 Memory Integration, Compute Constraints, and Bitwise Efficiency

To improve inference and planning efficiency, the agent incorporates structured memory - specifically, a schema memory - into its generative model. Rather than recomputing beliefs from scratch at every timestep, the agent can retrieve prior learned associations and compressive structures to inform future expectations.

Formally integrating memory into variational free energy computations is non-trivial. However, one tractable approach is to treat the output of schema memory as modifying or biasing the agent's prior beliefs. For example, if the memory system recalls that a particular outcome o is highly probable in a given context c, the generative model's prior can be adjusted accordingly, increasing $P(o \mid c)$. This effectively warps the generative landscape toward familiar, previously experienced outcomes - thereby biasing the expected free energy evaluation $G(\pi)$ toward those outcomes.

From an information-theoretic perspective, schema memory can be thought of as a structured prior over hidden states or transitions, reducing the entropy (uncertainty) that the agent needs to resolve through computation. This helps constrain planning to familiar subspaces of state space, dramatically reducing the sample complexity and expected surprise.

Compute Cost Constraint. To formally incorporate internal resource limitations, we introduce a cost term based on memory usage. Let b represent the number of memory bits accessed or activated in evaluating a policy. We define a compute cost penalty as:

$$C_{\text{compute}} = \lambda \cdot b^{\alpha}$$

where λ and α are tunable constants that control the scale and nonlinearity of compute cost. This reflects the intuition that using more memory incurs an energetic or temporal penalty others don't account for (Millidge, 2020), particularly when recalling complex or distant associations.

The agent's objective is thus augmented to minimize a total free energy that includes both epistemic surprise and internal compute cost:

$$F_t' = F_t + C_{\text{compute}}$$

This penalized objective leads the agent to trade off between external uncertainty resolution (through perception and planning) and internal efficiency (through selective memory use). In effect, the agent seeks not only to explain observations, but to do so economically - minimizing free energy under both environmental and cognitive constraints.

2.5.1 Bitwise Encoding and Architectural Efficiency in Rust

In our implementation, internal states and schema memories are represented using compact bitfields (e.g., 32-bit or 64-bit integers) rather than high-level data structures such as hash maps or JSON objects. This design choice provides profound advantages in both memory efficiency and computational speed - essential traits for agents operating under resource constraints.

Traditional memory structures, such as JSON, carry substantial overhead. They require text parsing, string-based key lookups, dynamic memory allocation, and runtime type resolution. The cost of accessing such structures scales with the number of fields n and the complexity of the format:

$$C_{\text{high}} = \lambda_{\text{parse}} \cdot n + \lambda_{\text{access}} \cdot \log n$$

where λ_{parse} is the cost of parsing, and λ_{access} is the cost of accessing nested or keyed elements. Even in optimized languages like C++, this involves branching, pointer chasing, and indirect memory access.

In contrast, Rust enables bitwise schema encoding, where state representations are directly embedded in integer registers. Feature extraction is performed via constant-time bit masking operations:

$$C_{\text{bitwise}} = \lambda_{\text{bitop}} \cdot b$$

where $b \ll n$ is the number of accessed bits, and λ_{bitop} approaches the CPU's base instruction latency - often just one or two clock cycles. The relative compute cost ratio becomes:

$$\frac{C_{\text{high}}}{C_{\text{bitwise}}} \approx \frac{\lambda_{\text{parse}} \cdot n + \lambda_{\text{access}} \cdot \log n}{\lambda_{\text{bitop}} \cdot b}$$

This yields speedups on the order of 10–100× in tight loops or real-time inference environments.

Memory Efficiency. The spatial advantages are similarly compelling. A bitfield requires only $B = k$ bits to represent k boolean or categorical attributes. By contrast, a JSON object storing the same information requires:

$$S = n \cdot (l_k + l_v + o)$$

bytes, where l_k and l_v are the average lengths of keys and values respectively, and o accounts for syntactic and structural overhead (e.g., quotes, colons, commas, and whitespace). The resulting compression ratio is:

$$\text{Compression ratio} = \frac{S}{B} = \frac{n \cdot (l_k + l_v + o)}{k}$$

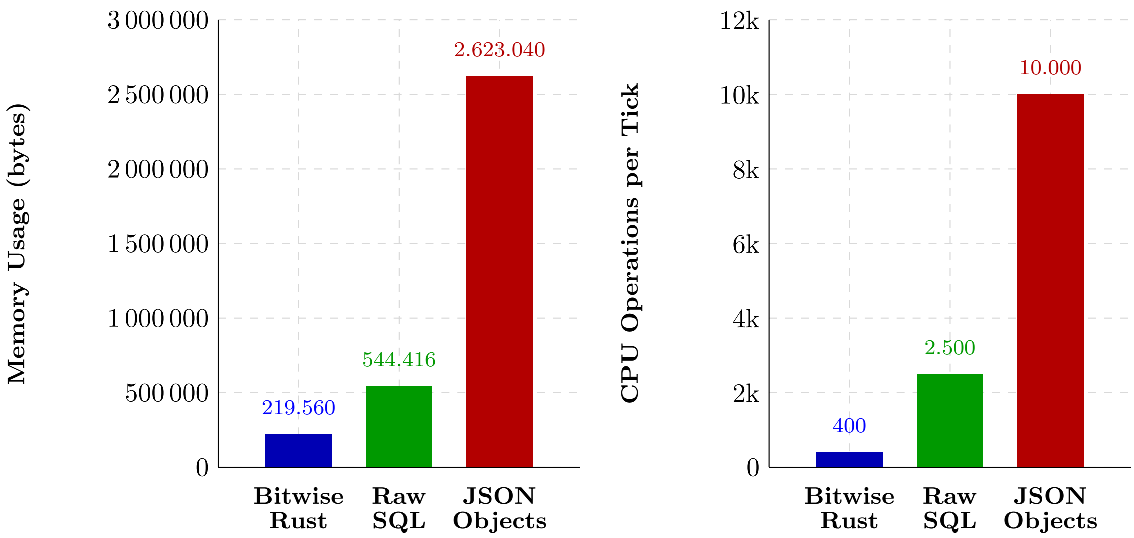

Figure 3: **Memory and Performance Comparison Across Architectures.** These charts compare the efficiency of a single SIML agent represented using Bitwise Rust, Raw SQL, and JSON Objects. **Left:** Bitwise Rust consumes just 219,560 bytes per agent-offering over **12**× savings versus JSON and over **2**× of Raw SQL . **Right:** CPU operations per tick are similarly reduced, with Bitwise Rust executing only **400** ops per tick versus **2,500** for Raw SQL and over **10,000** for JSON. Bitwise Rust achieves approximately a **6x** performance gain over Raw SQL and a **25**× gain over JSON, thanks to its low-level memory layout and direct bitfield access. These improvements stem from dense, bitfield-based encoding, ideal for constrained compute environments.

While languages like C++ offer bit-level control, they often depend on manual memory management and lack strict safety guarantees. Python, on the other end, favors high-level expressiveness at the cost of runtime efficiency. Rust uniquely combines low-level memory access with compile-time safety, zero-cost abstractions, and predictable performance - making it particularly well-suited for implementing biologically inspired, energy-efficient agents operating under bounded compute.

Bitwise schema representation is not just a micro-optimization - it is a foundational design decision that aligns the agent's internal cognitive machinery with the physical constraints of real-world computation. By minimizing both storage and compute costs, it enables agents to reason and act in real time, grounded in environments where every byte and cycle counts.

2.6 Free Energy Principle and Active Inferencing Modules

The SIML architecture implements an agent that performs both perceptual inference and goal-directed action selection by minimizing variational and expected free energy. Below, we describe the mathematical foundations of these two processes, as implemented in the **Inference Module**, **Generative Model**, and **Action Simulation Module** shown in Figure 1.

2.6.1 Perceptual Inference via Variational Free Energy (Inference Module)

At each timestep t, the **Inference Module** receives sensory input o_t and updates its internal belief distribution $Q(s_t)$ over hidden world states. This is done by minimizing the *variational free energy* (Friston, 2010; Buckley et al., 2017), defined as:

$$F_t = D_{\mathrm{KL}}\left[Q(s_t) \parallel P(s_t)\right] - E_{Q(s_t)}\left[\ln P(o_t \mid s_t)\right]$$

This balances two objectives: staying close to the prior distribution $P(s_t)$, and ensuring that the predicted state explains the current observation o_t. The belief distribution $Q(s_t)$ is updated by gradient descent:

$$\Delta\mu \propto -\frac{\partial F_t}{\partial \mu}$$

where μ represents the parameters of the approximate posterior.

Explanation of Variables in the Free Energy Equation

Symbol	**Meaning**
s_t	Latent (hidden) state of the world at time t
o_t	Observation or sensory input at time t
$Q(s_t)$	Posterior belief over hidden states at time t
$P(s_t)$	Prior belief over states (generative model)
$P(o_t \mid s_t)$	Likelihood of observing o_t given s_t
$D_{\mathrm{KL}}[\cdot \parallel \cdot]$	Kullback-Leibler divergence (distance between distributions)
F_t	Variational free energy at time t
μ	Internal parameters of belief distribution Q

Interpretation: The agent seeks to minimize surprise by adjusting its beliefs to explain observations as accurately as possible while remaining close to prior expectations. This constitutes the core perceptual inference loop.

2.6.2 Generative Model (Latent State Transition and Observation Prediction)

The **Generative Model** encodes how the agent believes the world behaves. It defines two key distributions:

- $P(o_t \mid s_t)$: the likelihood of observing o_t given the hidden state s_t.
- $P(s_{t+1} \mid s_t, a)$: the transition probability of reaching state s_{t+1} given the current state s_t and action a.

These distributions are learned and updated over time using experience and retrieved schema priors. Preferences, such as homeostasis or safety, are encoded in $P(s_{t+1})$ as soft priors over desirable states.

2.6.3 Action Simulation and $G(a)$ Selection (Action Selection Module)

The **Action Simulation Module** evaluates each candidate action a by computing its *expected free energy*:

$$G(a) = E_{Q(s_{t+1}, o_{t+1})} \left[\ln Q(s_{t+1}) - \ln P(s_{t+1}, o_{t+1})\right]$$

This can be decomposed as:

$$G(a) \approx \underbrace{E_{Q(s_{t+1})} \left[-\ln P(o_{t+1} \mid s_{t+1})\right]}_{\text{expected surprisal}} + \underbrace{D_{\mathrm{KL}} \left[Q(s_{t+1}) \,\|\, P(s_{t+1})\right]}_{\text{divergence from preferred states}}$$

The agent selects the action a^* that minimizes expected free energy:

$$a^* = \arg\min_a G(a)$$

This formulation balances two key pressures (Friston et al., 2015, 2016):

- **Epistemic value (uncertainty reduction):** Actions that reduce uncertainty about the world.
- **Pragmatic value (goal consistency):** Actions that lead to preferred or predictable internal states.

Explanation of Variables in the Expected Free Energy Equation

Interpretation: The agent simulates outcomes to choose actions that reduce future surprise and align with internal goals. This enables behavior that is both exploratory (epistemic) and intentional (pragmatic).

Symbol	Meaning
a	Candidate action at time t
$G(a)$	Expected free energy of action a
s_{t+1}	Predicted hidden state after executing a
o_{t+1}	Predicted observation after executing a
$Q(s_{t+1})$	Posterior belief over future states
$P(o_{t+1} \mid s_{t+1})$	Likelihood of observing o_{t+1} in state s_{t+1}
$P(s_{t+1})$	Preferred state distribution (agent's internal goals)

Concrete Example. Suppose the agent's internal energy is low. Schemas in memory suggest that moving forward has previously resulted in food discovery. The generative model predicts that eating will restore energy. Then:

- Actions leading to food have *low* expected surprise and match prior preferences (energy homeostasis).
- Standing still results in energy loss and eventual death—a highly surprising state under the generative model.

Thus, action $a =$ `move forward` will receive the lowest $G(a)$ and be selected.

Computational Considerations. For tractability, only one-step lookahead is used to compute $G(a)$. Future work may employ hierarchical or sampled rollouts to handle deeper planning.

Schema Integration. Schema memory supports action selection by:

1. Providing priors for transitions $P(s_{t+1} \mid s_t, a)$.
2. Flagging dangerous outcomes (e.g., death), reducing their likelihood during planning.

2.7 Why Rust?

Rust was chosen as the implementation language for SIML not only for its performance characteristics, but because its memory semantics, ownership model, and zero-cost abstractions align philosophically and technically with the SIML architecture. SIML agents are not just fast decision-makers-they are memory-constrained, adaptive lifeforms whose behavior emerges from internal predictive schemas encoded at the bit level. Rust provides the structural guarantees, granular control, and compilation-time safety required to bring such an organism to life without sacrificing performance or design integrity.

2.7.1 Alignment with SIML's Cognitive Architecture

Schema Encoding and Memory Safety. SIML schemas-compressed perceptual and outcome expectations-are encoded in compact bitfields (`u32`, `u64`) and stored in reusable structures (`SchemaSlot`). These are manually managed by the agent and frequently re-evaluated or discarded during sleep optimization. Rust's memory safety model ensures that these operations, including schema updates, fuzzed sensory matching, and energy delta logging, remain free of undefined behavior, buffer overflows, or invalid pointers-all without the need for garbage collection.

Zero-Cost Abstractions. Bitwise manipulation of perception and memory is central to SIML's surprise minimization:

```
let delta = last.0 ^ bf.0;
self.energy.0 -= delta.count_ones() as i32;
```

This operation quantifies informational surprise in constant time. Rust enables this ultra-efficient mechanism with full compile-time type safety, avoiding dynamic dispatch, boxing, or runtime allocation. Schema comparisons, chemosensory fuzzing, and entropy scoring are all implemented using traits and inlined logic that compile to direct machine code.

Entropy-Aware Memory Consolidation. The Free Energy Principle drives SIML agents to periodically optimize memory by sorting schemas based on match accuracy and energy impact:

```
self.schemas.sort_by_key(|s| {
    let match_quality = s.match_log.iter().filter(|b| **b).count() as u16;
    let danger = s.energy_deltas.iter().filter(|&&d| d < 0).count() as u16;
    std::u16::MAX - (match_quality * 10 - danger * 5)
});
```

Rust's iterator model and strong type system allow such entropy metrics to be implemented clearly and safely. At the same time, the borrow checker enforces safe in-place memory updates throughout simulation cycles.

2.7.2 Performance and Systems Integration

Low-Level Control with High-Level Guarantees. SIML interfaces with a real-time external Godot engine via TCP. Rust's ownership model and `Arc<Mutex<T>>` primitives ensure safe, shared access to the socket stream across threads. FlatBuffer messages (e.g., `Perception`, `Action`) are serialized with zero-copy semantics and interpreted directly from raw memory buffers using generated Rust bindings.

Deterministic Execution and Energy-Aware Compute. SIML agents operate under strict compute budgets:

```
let compute_cost = (self.used_bits as f32).powf(1.2) as i32;
```

These calculations directly impact survival. Rust guarantees that this execution remains predictable, free of GC pauses or heap fragmentation, enabling reliable simulation of metabolic expenditure.

2.7.3 Agent Evolution and Memory Persistence

Simulation Continuity. Agents serialize their memory and schema structure to disk and reload on subsequent runs:

```
if let Ok(mut file) = File::open("agent_memory.bin") {
    ...
}
```

This persistence allows learned behaviors to carry forward-enabling mutation, adaptation, and generational selection. Rust's `bincode` and structured error handling make this straightforward and efficient.

The SIML system represents a new form of embodied artificial cognition-rooted in energy constraints, internal predictive modeling, and surprise minimization. Rust is not merely a performance booster or syntactic preference; it is foundational to this paradigm. Its strict memory guarantees, zero-cost abstraction, and seamless systems integration make it uniquely suited to implement artificial life where perception, prediction, and memory co-evolve under physical and cognitive constraints.

Rust is not just a language for SIML. It is the neural scaffold of its mind for this current period of time and tooling.

2.8 Summary

In summary, the Methods described an agent that is self-motivated by an internal consistency principle. It has a form of curiosity (driven by reducing uncertainty) and goal-directedness (driven by maintaining homeostasis and reaching expected desirable states) without any explicit reward or punishment coded by us. This is a fundamentally different approach to creating AI agents, one that Cognogenics champions. Next, we turn to the empirical results demonstrating how such agents behave in practice.

3 Results

We evaluated the SIML architecture in a simulated 3D voxel world to assess the emergent behavior and adaptability of Cognogenics agents under varying conditions. Our core questions were:

1. Can agents survive and perform adaptive behaviors without any explicit reward signal?
2. What kinds of behavioral patterns emerge from the interaction of free-energy-based decision-making and memory-constrained schema learning?

To answer these, we compared agents under two modes: a **schema memory-enabled SIML agent** versus a **memoryless (reactive) agent**, across three levels of environmental difficulty (500, 1500, and 2500 nutrient tiles). Agents were initialized at random positions and orientations and evaluated on survival duration and behavioral traits.

3.1 Survival Behavior Without External Rewards

Agents driven by free-energy minimization were capable of surviving and achieving implicit objectives-such as foraging and energy regulation-without any explicit rewards.

Nutrient Seeking: In early runs, agents explored randomly, but after encountering nutrient voxels and experiencing a rise in internal energy, they began to associate those outcomes with prediction error reduction. This led to a form of emergent foraging: agents adjusted motion toward areas where nutrients had previously been found, exhibiting a biased walk akin to **chemotaxis** in microorganisms.

When expectations were confirmed (e.g., finding food in a predicted location), agents persisted in that direction; when violated, they adjusted behavior or changed course.

Persistence and Directionality: Once a preferred direction emerged, agents often maintained that trajectory until interrupted-demonstrating **goal-seeking locomotion**. Even in barren environments, agents avoided erratic movements, favoring predictability over randomness. This stands in contrast to typical reinforcement learning agents, which tend to stagnate in the absence of reward signals.

3.2 Memory, Habits, and Loops

The schema memory system enabled agents to generalize prior experiences (Parr and Friston, 2017; Spens and Burgess, 2024) and make longer-term inferences . However, memory constraints also introduced **behavioral inertia** and occasional suboptimal habits.

Looping Behavior: In several trials, agents returned repeatedly to previously food-rich regions even after those areas were depleted. This resulted in "checking loops" driven by outdated internal schemas. Only after repeated failures did agents update their internal models and explore elsewhere. This mirrors habitual patterns observed in simple biological systems and illustrates bounded rationality emerging from architectural constraints.

Adaptive Rerouting: When direct paths to food were blocked by hardened terrain, SIML agents successfully navigated around obstacles to reach nutrients. Their schema memory supported route planning and detour formation, even in the absence of visual cues. Reactive agents failed in such conditions, typically dying after unsuccessful local searching.

3.3 Expectation Violation and Learning

To test the presence of predictive modeling, we designed an environment where food was initially accessible via a short corridor, then removed and placed elsewhere.

Internal Model Adjustment: On the first trial, the agent moved directly toward the old location, found nothing, and began exploring nearby routes. It eventually discovered the new food location via a longer detour, updated its memory, and in the next trial, took the new route directly. This behavior indicates the formation and revision of a **world-model** that generalizes beyond reactive stimulus-response.

Free Energy Dynamics: Logging the agent's predicted free energy F showed that as it learned the environment, average F decreased. When the environment changed unexpectedly, F spiked, followed by a gradual decline as the agent updated its model. This is consistent with the theoretical signature of active inference: surprise drives adaptation, and steady-state behavior emerges when expectations are fulfilled.

3.4 Quantitative Survival Outcomes

To quantify agent performance, we measured average survival duration across 10 randomized trials per condition. As shown in Figure 4, agents equipped with schema memory significantly outperformed reactive agents under all conditions:

Figure 4: **Trajectory of a SIML Agent. Left:** Top-down projection of a SIML agent's movement at the beginning and end on a 3D voxel grid over 130 steps. The agent begins at (6, 25), forms predictive schemas, and discovers food at (9, 18). It later becomes trapped in a behavioral loop and dies at (7, 17) from energy depletion. Arrows indicate movement direction and decision-making over time. **Right:** Snapshot of the voxel grid environment rendered by the simulator.

3.5 Behavioral Summary

In summary, SIML agents exhibited:

- **Emergent goal-seeking behavior**, such as avoiding starvation or seeking shelter, without any programmed reward function.
- **Adaptive memory use and re-planning**, adjusting to environmental changes through schema updates.
- **Efficient exploration-exploitation balance**, cycling between broad environmental sweeps and focused nutrient collection.
- **Bounded rationality and cognitive bias**, such as habit formation, memory-driven loops, and failure to anticipate unseen conditions.

Table 1: Average Agent Survival Time (in ticks) Across Conditions

Agent Type	**500 Food**	**1500 Food**	**2500 Food**
No Memory (Reactive)	10	12	13
Base SIML (Single Gen)	15	16	18
Multi-Gen SIML (Saved Bin)	120	145	165

These results demonstrate that agents guided solely by free energy minimization and bitwise memory structures can achieve adaptive, homeostatic behavior in open-ended environments-without supervision or rewards. The Cognogenics model thus provides a biologically plausible path to autonomous, self-motivated cognition.

4 Discussion

The emergence of adaptive, life-like behavior in SIML agents supports the central premise of Cognogenics: that cognition can be generated and organized by adhering to fundamental thermodynamic or information-theoretic principles (like free energy minimization) rather than extrinsic reward maximization or extensive prior knowledge. Here, we discuss the significance of this approach, contrasting it with prevailing paradigms and outlining the path toward more complex cognition.

4.1 A Paradigm Shift from Reward to Energy

Traditional reinforcement learning is built on the idea of maximizing cumulative rewards. This requires researchers to define what constitutes a reward for the agent, effectively biasing the agent toward tasks we care about. In natural cognition, however, organisms are not born with explicit reward tables for every scenario – instead, they have evolved drives (hunger, curiosity, pain avoidance) that implicitly guide them. Cognogenics formalizes these drives in terms of prior expectations in a generative model. There is no need to handcraft a reward function; the "reward" is essentially the avoidance of surprise – a far more general objective. As a result, our SIML agents exhibit autonomy: they self-determine what is important based on an internal model of the world, akin to how a creature finds water because thirst creates a prediction that water should be found. The Free Energy Principle (FEP) provides a normative, unified objective for perception and action (Friston et al., 2023; Zhang and Xu, 2024).

This is a profound departure from the ad-hoc objective functions in machine learning. It suggests that any cognitive system (biological or artificial) that maintains itself in a changing environment will do so by minimizing free energy (prediction errors) over time. Our work operationalizes this insight in a concrete agent. The benefit is that our agent's behavior is grounded in the same principle that governs its internal representation learning. By contrast, in RL one often sees divergence between the learned policy and learned model (if any) – e.g. an agent might exploit a reward hack that doesn't actually improve its understanding of the environment. In our case, such exploitative shortcuts are less likely because a narrow exploit that doesn't generally reduce surprise will not persist (the agent would still be surprised by other aspects, pushing it to not overfixate). Indeed, we did not observe any pathological reward-hacking behaviors common in some RL agents; the SIML agent's activities were all in service of maintaining internal consistency.

4.2 Embodiment and Genuine Understanding

A key difference between Cognogenics agents and large language models or other generative learners is embodiment. LLMs like GPT are passive predictors: they take in a prompt and predict an outcome (text) with no capability to act in the world or test their predictions. This passivity limits their understanding – they cannot ground their predictions in sensorimotor experience. In contrast, our SIML agents are embodied in a 3D world and continuously close the loop between prediction and action. This yields a form of grounded cognition that many in AI have argued is essential for true general intelligence. Recent theoretical work supports this: generative models that are inextricably tied to a body can develop understanding by constantly testing themselves against reality, whereas passive generative AI (like LLMs) lacks that corrective loop. In the words of Pezzulo (Pezzulo et al., 2024), "Unlike the passive models learned by generative AI systems, [an embodied agent's generative models] must capture and control the sensory consequences of action. This allows embodied agents to intervene upon their worlds in ways that constantly put their best models to the test," which is essential for genuine understanding. Cognogenics instantiates this idea: the agent's knowledge is not just statistical correlation (as in LLMs) but actionable information about the world that has been verified through interaction.

Our results, albeit in simple environments, demonstrate that such an embodied agent can self-motivate to explore and learn. For instance, an SIML agent formed a notion of a "safe area" and returned to it after gathering food. This is similar to how an animal might establish a home base – something an LLM can never do, and an RL agent would only do if explicitly rewarded for it. The difference comes from intrinsic motivation. In Cognogenics, intrinsic motivations (like curiosity, comfort-seeking) are not add-ons but naturally fall out of the FEP framework: novel stimuli are sought because they reduce uncertainty, and comfortable states are sought because they fulfill prior expectations of homeostasis. It is compelling that a single formalism accounts for both kinds of motivation, which are usually separately engineered in AI (extrinsic vs. intrinsic reward).

4.3 Comparison to Evolutionary Approaches

The field of artificial life (ALife) often uses evolutionary algorithms to "grow" agents or behaviors. This involves a population of agents, mutation, selection, etc., to optimize some fitness (survival, task success, etc.). While evolutionary methods have created fascinating virtual creatures and demonstrated open-ended novelty, they fundamentally operate across generations, not within an individual's life. Cognogenics proposes that we can achieve life-like adaptation within a single agent's lifespan by endowing it with the right cognitive architecture. In a sense, our approach is orthogonal to evolution: instead of evolution designing the brain, we design a brain that can itself do in-lifetime adaptation analogous to what evolution would achieve over many generations.

One might ask: could evolution just produce a similar result? Indeed, one way to view active inference is as "the brain implementing a kind of inner evolution or inner selection over predictions." Some theorists have drawn parallels between the free energy minimization and evolutionary processes – viewing perception as a kind of natural selection over mental models. However, by explicitly constructing agents with active inference, we bypass the need for an actual evolutionary algorithm in our simulations. This is more efficient for developing intelligent behavior: rather than evolving thousands of agents to maybe get one that behaves well, we concentrate computation on a single agent learning in real time. That said, Cognogenics doesn't exclude evolution entirely – one could imagine using evolutionary search to optimize the hyperparameters of an active inference agent (or its architecture) for a given environment. But importantly, even without such outer-loop optimization, our agents already display complex behaviors. This hints at a future synergy: combining a lifetime learning paradigm (Cognogenics) with slower evolutionary tuning could yield even more powerful, robust agents than either alone.

4.4 Toward General Intelligence and Posthuman Minds

Our experiments focused on relatively simple sensorimotor behaviors reminiscent of worms or insects: foraging, navigation, habit formation. The vision of Cognogenics is to scale this up gradually:

- By increasing the richness of the generative model (more layers of abstraction, hierarchy of schemas), we could enable higher cognitive functions (planning multiple steps ahead, tool use, etc.). Hierarchical active inference models already exist in neuroscience; applying them in SIML could let an agent not just react to immediate surprise, but entertain long-term "what if" scenarios.
- Expanding the bitwise memory into a more powerful compositional knowledge base could allow for something akin to reasoning. For instance, schemas might be linked in graphs, enabling the agent to chain many experiences and thus solve novel problems by analogical recall.
- Embedding the agent in environments of increasing complexity (including social environments with other agents) would test the architecture's scalability. We expect that the same FEP-based drive that works for nutrient seeking can work for social interaction if the agent has a model of other agents (this would minimize surprise by achieving affiliation or avoiding conflict, etc., depending on what its "social homeostasis" might be).
- Integration with deep learning could provide perceptual grounding. For example, a neural network trained to encode visual scenes could feed into the bitwise schema layer, allowing rich real-world inputs to be processed using the Cognogenics architecture.

The long-term argument we put forward is that Cognogenics provides a route to artificial general intelligence (AGI) that is fundamentally different from scaling up deep learning on internet data or brute-forcing evolution in simulation. Instead of training on terabytes of data or evolving billions of creatures, one could build an agent that develops its intelligence much like a human child – through continuous interaction, play, and intrinsic motivation. Such an agent, if given a sufficiently rich world and perhaps human guidance, could in principle attain human-level cognitive capabilities (language, abstraction, commonsense) grounded in its own experience.

Moreover, once we have agents that reach human-like cognition, we are not bound by the limits of biology. These agents could be extended or enhanced in ways biological brains cannot: faster processing, different sensory modalities, altered reward (or rather, prior) systems leading to novel drives. Cognogenics could thus also be a path to posthuman minds – forms of intelligence that don't directly mimic human psychology but are self-organizing, adaptive and possibly exceed human capabilities. For example, an agent might be embodied in a complex simulation where it can experience realities very different from ours (additional spatial dimensions or quantum-scale senses); it would develop its own cognition suited to that reality, potentially giving us insight into entirely new modes of thinking. While speculative, this illustrates that Cognogenics isn't just about matching human intelligence – it's about understanding the general principles of cognition such that we can generate intelligence in any substrate, for any environment.

5 Challenges and Future Work

While the SIML framework demonstrates the viability of Cognogenics as a foundation for adaptive agents, several significant challenges remain before it can scale to more complex environments or real-world deployments. We outline the most pressing obstacles and corresponding directions for future research:

1. **Scalable Active Inference Computation.** Evaluating expected free energy across a wide action space can be computationally intensive. Scaling SIML to higher-dimensional perception-action loops will require more efficient inference mechanisms. Promising directions include:
   - *Amortized inference* using neural networks to approximate posterior updates.
   - *Monte Carlo Tree Search* adapted to FEP-based planning.
   - *Sparse action priors* to constrain planning to behaviorally relevant regions.

2. **Learning Generative Model Structure.** Currently, the generative model's structure is fixed; the agent only adapts certain parameters. In unconstrained or open-world scenarios, an agent will need to learn *what* to model – discovering relevant latent variables and causal dependencies from scratch. Structure learning methods (e.g., variational autoencoders, graph neural nets) may be adapted here, but must remain grounded in the free energy formalism.

3. **Extending Memory Capacity and Representation.** While the bitwise schema memory is interpretable and efficient, its fixed format may become brittle in richer environments. Future work could explore:
   - Hierarchical schema networks that chunk experiences across multiple scales.
   - Hybrid memory models combining discrete schemas with continuous vector spaces.
   - Compression and pruning techniques for lifelong schema evolution.

4. **Deployment in Real Embodied Agents.** Moving beyond simulation, testing Cognogenics in physical robotics is a key milestone. Immediate testbeds include:

- *Exploratory robots* that forage in semi-structured environments (e.g., warehouse or agricultural drones).
- *Homeostasis-driven agents* in energy-constrained contexts (e.g., mobile IoT swarms that learn to self-balance power or heat).
- *Curious agents* for adaptive search or reconnaissance.

5. **Formal Theoretical Integration.** Cognogenics sits at the intersection of neuroscience, artificial life, and information theory. Further formalization is needed to:
   - Relate SIML agents to *Markov blankets*, *autopoiesis*, and *empowerment* in complex systems theory.
   - Connect schema evolution to formal models of memory consolidation and sleep in cognitive neuroscience.
   - Define a rigorous taxonomy of Cognogenics architectures and their computational expressivity.

6. **EDEN: A Benchmark for Emergent Embodied Intelligence.** A key future direction is the creation of EDEN (Embodied Developmental Evaluation Nexus)-a high-fidelity, biologically grounded testbed for evaluating SIML agents in structured environments. Unlike LLM-focused benchmarks (e.g., MMLU, BIG-Bench), EDEN evaluates intelligence via embodied interaction, continuous prediction, and survival under uncertainty.

   EDEN is built atop Apple's Unified Memory Architecture (UMA), leveraging Rust for agent cognition and Metal GPU compute shaders for simulating environmental processes (e.g., chemodiffusion, energy decay) with zero-copy latency. Agents operate directly on a shared memory substrate, avoiding serialization overhead and mimicking tight, biological sensorimotor loops.

   Key features include:
   - **Bitwise Environmental Encoding:** u32/u64-encoded chemistry, signal states, and nutrient maps.
   - **Schema-Driven Agents:** SIML agents use FEP-guided surprise minimization and schema memory to make decisions.
   - **Zero-Copy World–Agent Feedback:** Real-time shared memory enables highly efficient, grounded behavior.
   - **Metric System (SIML-IQ):** Evaluation based on information-theoretic metrics such as IQCE (Information per Compute per Energy), schema compression gain, and adaptive survival under shifting environmental constraints.

   EDEN will evolve into a general benchmark for embodied intelligence-measuring an agent's ability to adapt, generalize, survive, and self-organize over time. Future expansions will support multi-agent interaction, cooperation, manipulation, and transfer learning. A standardized schema API and environmental encoding will allow third-party integration and community-driven benchmarking.

Despite these challenges, our work presents a compelling proof of concept: even with simple internal machinery and minimal supervision, SIML agents exhibit structured, survival-oriented, and context-sensitive behavior. We believe Cognogenics opens the door to a new generation of truly autonomous cognitive systems-rooted not in rewards or scripts, but in the fundamental thermodynamics of life.

6 Vision: A New Scientific Era

We call Cognogenics a new scientific field deliberately, in analogy to how genetics revolutionized biology by explaining the principles of inheritance and variation. By analogy, Cognogenics seeks to uncover the principles by which cognitive structures can originate ("genesis") and evolve within individual agents. It bridges the gap between life sciences and information sciences. The implications span many domains:

- In AI, it offers a route to machines that are robust, autonomous and continuously learning without extensive external reward engineering.
- In cognitive science, it provides a modeling framework to test theories of how animals or humans might produce complex behavior from simple self-motivated processes.
- In philosophy of mind, it feeds into discussions on what it means for a system to have desires, goals, or understanding when those emerge spontaneously from the system's dynamics rather than being programmed.
- In ethics and safety, an interesting aspect is that agents which develop their own drives might have "preferences" we didn't anticipate. Ensuring alignment of those intrinsic preferences with human values might be a challenge – but on the flip side, such agents might be more understandable (since we can inspect their generative model and schemas) than a black-box trained neural network.

We believe we are at the cusp of a paradigm shift. Just as early work in cybernetics and artificial life hinted that "machines could be like living organisms," Cognogenics concretely demonstrates an artificial organism-like agent in silico. The manifesto of Cognogenics is to pursue this approach: to build increasingly general, adaptive agents not by bigger data or brute force, but by better principles and architectures. The results herein serve as a foundational step, showing that FEP-based agents can indeed exhibit core aspects of autonomous intelligence. In conclusion, the field of Cognogenics – starting from systems like SIML – holds great promise for advancing our understanding of intelligence. By unifying ideas of active inference, embodiment, and self-organization, and backing them up with working implementations, we can begin to chart a course toward truly general, self-directed cognitive agents. This journey could ultimately lead us to synthetic minds that not only match human cognitive capacities but explore new realms of thought and existence, enriching the landscape of intelligence in the universe.

7 Related Work

Cognogenics synthesizes principles from neuroscience, AI, cognitive science, and artificial life into a unified framework. Below, we compare our approach to key prior paradigms, highlighting how SIML advances or diverges from them.

Free Energy Principle and Active Inference

Our work operationalizes the Free Energy Principle (FEP) in embodied agents using bitwise schema memory and surprise minimization (Friston, 2010). Unlike many active inference models which operate in low-dimensional or task-specific simulations, SIML agents inhabit a 3D voxel world and exhibit multiple emergent behaviors (e.g., foraging, homeostasis, self-looping) from a single architecture.

Tschantz et al. (2020) demonstrated that expected free energy minimization yields action-oriented models balancing exploration and goal achievement (Tschantz et al., 2020). Our results align with this, but extend it by incorporating scalable memory and grounded embodiment. Unlike reinforcement learning (RL), which requires explicit rewards, SIML agents behave adaptively with no reward function—driven solely by prediction error minimization.

Embodied Cognition and Enactive AI

SIML is a direct instantiation of embodied cognition (Varela et al., 1991): the agent's predictive model is inextricably tied to its physical embodiment. Where Braitenberg Vehicles showed emergence via fixed circuits (Braitenberg, 1984), SIML agents form internal predictive models. They perceive affordances (Gibson, 1979) not abstractly, but in terms of survival-relevant actions (e.g., "food is reachable"), learned through experience.

Artificial Life and Adaptive Behavior

Unlike evolutionary ALife systems (e.g., Tierra (Ray, 1991), Avida (Ofria and Wilke, 2004), Polyworld (Yaeger, 1994)), which rely on genetic variation across generations, SIML supports real-time, within-lifetime adaptation. Empowerment and intrinsic motivation (Schmidhuber, 1991; Klyubin et al., 2005) arise not from explicit objectives, but as byproducts of FEP. This enables agents to balance novelty-seeking and survival without additional reward signals or modules.

Cognitive Architectures

Cognogenics departs from symbolic cognitive architectures (Soar (Laird et al., 1987), ACT-R (Anderson and Lebiere, 1998), SPAUN (Eliasmith et al., 2012)) by embracing self-organizing, probabilistic behavior via schema-driven learning. While ACT-R includes hand-coded production rules, SIML's schemas are acquired dynamically and grounded in embodied interaction. This positions Cognogenics as a hybrid architecture—probabilistic yet structured, reactive yet compositional.

Memory and Schema Learning

Our schema memory draws on psychological notions of schemas (Bartlett, 1932) and connects with work in structured reasoning (Goyal et al., 2020) and active inference structure learning (Tervo et al., 2016). The SIML agent captures and reuses causal regularities in "if context, then outcome" patterns, paralleling mechanisms like hippocampal replay or dream-based planning in biology (Arbib, 1992).

Theoretical Foundations

Cognogenics exemplifies the connection between FEP and thermodynamics. SIML agents resist entropy by maintaining low-surprise states—analogous to homeostasis and autopoiesis (Maturana and Varela, 1980). Markov blankets are realized in practice: sensor inputs and actuator outputs define the boundary through which the agent engages its world.

Summary Comparison

Approach	Learning Type	Memory	Objective
Reinforcement Learning	Reward-maximizing	Implicit (NN weights)	Task-specific reward
Evolutionary Agents	Intergenerational	Static across life	Fitness (population)
Deep Learners (e.g. LLMs)	Offline, supervised	Vector embeddings	Likelihood/max-margin
SIML (Cognogenics)	Online, embodied	Schema memory (bitwise)	Free energy minimization

In conclusion, Cognogenics builds on rich traditions but synthesizes them into a unique system: a grounded, generative, self-organizing agent architecture capable of adaptive behavior in lifelike environments-without external rewards or hand-crafted control. It offers a new path forward for understanding and engineering general intelligence.

References

Anderson, J. R. and Lebiere, C. (1998). *The Atomic Components of Thought.* Lawrence Erlbaum Associates, Mahwah, NJ.

Arbib, M. A. (1992). Schema theory. In Shapiro, S. C., editor, *Encyclopedia of Artificial Intelligence, 2nd edition*, pages 1427–1443. John Wiley & Sons, New York, NY.

Bartlett, F. C. (1932). *Remembering: A Study in Experimental and Social Psychology.* Cambridge University Press, Cambridge, UK.

Braitenberg, V. (1984). *Vehicles: Experiments in Synthetic Psychology.* MIT Press, Cambridge, MA.

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47(1–3):139–159.

Buckley, C. L., Kim, C. S., McGregor, S., and Seth, A. K. (2017). The free energy principle for action and perception: A mathematical review. *Journal of Mathematical Psychology*, 81:55–79.

Chemero, A. (2009). *Radical Embodied Cognitive Science.* MIT Press, Cambridge, MA.

Clark, A. (1997). *Being There: Putting Brain, Body, and World Together Again.* MIT Press, Cambridge, MA.

Dreyfus, H. L. (1992). *What Computers Still Can't Do: A Critique of Artificial Reason.* MIT Press, Cambridge, MA.

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., and Rasmussen, D. (2012). A large-scale model of the functioning brain. *Science*, 338(6111):1202–1205.

Friston, K. (2009). The free-energy principle: a rough guide to the brain. *Trends in Cognitive Sciences*, 13(7):293–301.

Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127–138.

Friston, K., Da Costa, L., Sajid, N., Heins, C., Ueltzh"offer, K., Pavliotis, G. A., and Parr, T. (2023). The free energy principle made simpler but not too simple. *Physics Reports*, 1024:1–29.

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., O'Doherty, J., and Pezzulo, G. (2016). Active inference and learning. *Neuroscience & Biobehavioral Reviews*, 68:862–879.

Friston, K., Kilner, J., and Harrison, L. (2006). A free energy principle for the brain. *Journal of Physiology-Paris*, 100(1-3):70–87.

Friston, K., Rigoli, F., Ognibene, D., Mathys, C., Fitzgerald, T., and Pezzulo, G. (2015). Active inference and epistemic value. *Cognitive Neuroscience*, 6(4):187–214.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston, MA.

Goyal, A., Lamb, A., Gampa, P., Beaudoin, P., Levine, S., Blundell, C., Bengio, Y., and Mozer, M. (2020). Object files and schemata: Factorizing declarative and procedural knowledge in dynamical systems. *arXiv preprint arXiv:2006.16225*.

Klyubin, A. S., Polani, D., and Nehaniv, C. L. (2005). Empowerment: A universal agent-centric measure of control. In *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, volume 1, pages 128–135.

Laird, J. E., Newell, A., and Rosenbloom, P. S. (1987). Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64.

Maturana, H. R. and Varela, F. J. (1980). *Autopoiesis and Cognition: The Realization of the Living*. D. Reidel Publishing, Dordrecht, Netherlands.

Millidge, B. (2020). Deep active inference as variational policy gradients. *Journal of Mathematical Psychology*, 96:102348.

Ofria, C. and Wilke, C. O. (2004). Avida: A software platform for research in computational evolutionary biology. *Artificial Life*, 10(2):191–229.

Parr, T. and Friston, K. J. (2017). Working memory, attention, and salience in active inference. *Scientific Reports*, 7:14678.

Pezzulo, G., Parr, T., Cisek, P., Clark, A., and Friston, K. (2024). Generating meaning: active inference and the scope and limits of passive ai. *Trends in Cognitive Sciences*, 28(3):195–207.

Pfeifer, R., Bongard, J., and Grand, S. (2007). *How the Body Shapes the Way We Think: A New View of Intelligence*. MIT Press, Cambridge, MA.

Ray, T. S. (1991). An approach to the synthesis of life. In *Artificial Life II*, volume X of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 371–408. Addison-Wesley, Redwood City, CA.

Schmidhuber, J. (1991). A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proceedings of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pages 222–227.

Spens, E. and Burgess, N. (2024). A generative model of memory construction and consolidation. *Nature Human Behaviour*, 8:526–543.

Suchman, L. A. (1987). *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, Cambridge, UK.

Tervo, D. G. R., Tenenbaum, J. B., and Gershman, S. J. (2016). Toward the neural implementation of structure learning. *Current Opinion in Neurobiology*, 37:99–105.

Tschantz, A., Seth, A. K., and Buckley, C. L. (2020). Learning action-oriented models through active inference. *PLOS Computational Biology*, 16(4):e1007805.

Varela, F. J., Thompson, E., and Rosch, E. (1991). *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press, Cambridge, MA.

Yaeger, L. S. (1994). Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior, or, polyworld: Life in a new context. In Langton, C. G., editor, *Artificial Life III, Proceedings of the Third International Conference on Artificial Life*, pages 263–298, Reading, MA. Addison-Wesley.

Zhang, Z. and Xu, F. (2024). An overview of the free energy principle and related research. *Neural Computation*, 36(5):963–1021.