A Synthetic Framework For AI Genome: Heritable Architectures for Open-Ended Evolution of Super Agentic Intelligence

Basura Fernando

Institute of Advanced Intelligence and Computing (IAIC), Agency for Science, Technology and Research, Singapore
Centre for Frontier AI Research, Agency for Science, Technology and Research, Singapore
College of Computing and Data Science, Nanyang Technological University, Singapore
Corresponding author: fernando_basura@a-star.edu.sg

18 September 2026

Abstract
Agentic AI can reason, remember, use tools, learn, and modify its behaviour, yet humans still assemble most agent architectures and workflows. We propose a functional synthetic framework for artificial heredity. An **AI Genome** is transmissible information that causally influences the construction, development, or inherited dispositions of descendant agents. It may include parameters, programs, prompts, memory architectures, tools, and learning rules, while excluding acquired state that is not inherited.
Our genotype–development–lifetime-state–phenotype framework separates development, learning, genomic consolidation, inheritance, and population evolution. It clarifies how acquired skills or memories could become heritable, and how Darwinian, Baldwinian, Lamarckian, horizontal, and cultural processes may interact. We connect this framework to neuroevolution, open-ended evolution, automated agent design, and self-modifying systems. The resulting research agenda asks what agents should inherit, what they should learn, which acquired capabilities should pass to descendants, and how these mechanisms might sustain open-ended evolution of agentic intelligence.

Keywords: artificial life, artificial heredity, AI genome, open-ended evolution, agentic AI, evolutionary computation

1 Introduction

Biological intelligence is not the product of a single optimization process acting on a fixed architecture. It emerges from multiple interacting processes operating across different timescales. Evolution modifies heritable information across generations; development transforms inherited information into functioning organisms; learning adapts individuals during their lifetimes; memory preserves experience; social interaction enables knowledge transfer; and cultural evolution allows useful discoveries to accumulate and propagate far more rapidly than genetic change alone.

Modern artificial intelligence exhibits analogues of many of these processes, but they are typically studied in isolation. Neural parameters are optimized through gradient-based learning, architectures are manually designed or discovered through neural architecture search, and foundation models are adapted through fine-tuning or parameter-efficient learning. Agentic systems accumulate episodic and semantic memories, retrieve external knowledge, and extend their capabilities through tools and APIs. Multi-agent systems communicate, cooperate, compete, and exchange information, while meta-learning optimizes how systems learn. Evolutionary algorithms search over parameters, architectures, prompts, hyperparameters, and learning rules. More recently, autonomous agents have begun modifying their own prompts, programs, workflows, tools, memory mechanisms, and even processes responsible for self-improvement. The emerging question is therefore no longer simply *How should we train an intelligent model?* A deeper question is **What information specifies an intelligent agent, which parts of that information should be inherited, what should instead be acquired during the agent's lifetime, and how can populations of such agents evolve toward increasingly capable forms of intelligence?**

We use the term **AI Genome** to provide a unifying framework for addressing this question. Rather than treating the genome as a metaphor for neural parameters alone, we consider it as the *heritable specification* from which an intelligent agent is instantiated and through which capabilities, learning mechanisms, cognitive structures, developmental rules, and mechanisms of further adaptation can be transmitted across generations.

Classical neuroevolution already demonstrates that evolutionary search can act on neural parameters, architectures, activation functions, hyperparameters, and even learning algorithms (Stanley et al., 2019; Galván and Mooney, 2021). NEAT jointly evolves network topology and connection weights while protecting structural innovations through speciation (Stanley and Miikkulainen, 2002), while indirect encodings such as HyperNEAT show that compact generative representations can specify neural systems far larger and more structured than the genome itself (Stanley et al., 2009). These results establish an important principle: the artificial analogue of a biological genome need not explicitly enumerate every component of the resulting intelligent system. Instead, it may encode rules, structures, and developmental mechanisms from which complex agents are constructed.

Open-ended evolution extends this perspective beyond optimization toward a predefined solution, emphasizing continuing novelty, complexity, new niches, and new forms of adaptation (Taylor et al., 2016; Packard et al., 2019; Taylor, 2019). Quality-diversity methods similarly seek repertoires of diverse, high-performing solutions instead of collapsing a population toward a single optimum (Mouret and Clune, 2015; Pugh et al., 2016). Clune's AI-Generating Algorithms proposal further argued that architectures, learning algorithms, and learning environments themselves could become objects of automated generation rather than being manually specified (Clune, 2019).

Foundation models make this broader vision considerably more practical because they can represent, generate, interpret, and modify many of the symbolic and executable structures from which agents are built. Large language models can manipulate programs, prompts, workflows, memory structures, tools, planning procedures, and agent architectures. Promptbreeder evolves both task prompts and the prompts responsible for generating future mutations (Fernando et al., 2024). Automated Design of Agentic Systems (ADAS) treats agent programs themselves as a search space and employs a meta-agent to discover improved designs (Hu et al., 2025). EvoAgent applies evolutionary operators such as mutation, crossover, and selection to construct multi-agent systems (Yuan et al., 2025). Godel Agent modifies aspects of its own logic and behaviour (Yin et al., 2025). The Darwin Godel Machine goes further by maintaining an archive of agents, modifying their code using foundation models, empirically evaluating descendants, and exploring a growing lineage of agent designs (Zhang et al., 2026).

Recent work has also begun to address heredity more directly. Learngene proposes inheritable neural fragments through which task-agnostic knowledge can be transmitted across generations (Feng et al., 2025). Genomebook constructs an explicit genetic representation for LLM agents and studies behavioural inheritance across multiple generations (Corpas, 2026). Such developments show that artificial heredity itself is becoming computationally realizable. However, these approaches generally instantiate heredity through particular representational substrates. The broader question remains: *what constitutes the heritable specification of a complete agentic system, and how should that specification interact with development, lifetime learning, memory, horizontal transfer, cultural transmission, and evolution?* These developments indicate a fundamental expansion in the unit subjected to artificial evolution:

$$\text{parameters} \rightarrow \text{architectures} \rightarrow \text{learning systems} \rightarrow \text{complete intelligent agents.}$$

Once complete agents become objects of variation, selection, reproduction, and inheritance, a corresponding conceptual framework of heredity becomes necessary. Existing approaches demonstrate that particular substrates such as weights, neural modules, prompts, behavioural traits, or programs can be evolved or inherited. The broader unresolved problem is how to characterize the heritable information of an agent, distinguish it from information acquired during its lifetime, and determine how hereditary information can be transmitted, recombined, modified, accumulated, and itself evolved.

The AI Genome framework makes three main conceptual contributions. First, it defines artificial heredity *functionally*, rather than identifying the genome with any particular representation such as neural parameters or executable code. Second, it introduces a genotype–development–lifetime-state–phenotype framework that distinguishes learning and self-modification from genuine hereditary change. Third, it develops an evolutionary lifecycle connecting development, lifetime learning, genomic consolidation, inheritance, horizontal transfer, and population evolution, and uses this framework to motivate a research agenda for open-ended evolution of agentic intelligence.

2 From Biological Genome to AI Genome

In biology, DNA should not be interpreted as a literal blueprint that directly specifies the final state of an organism. A genotype instead provides inherited information that interacts with developmental processes, environmental conditions, and subsequent experience to produce a phenotype. Abstractly,

$$P = \mathcal{D}(G, E),$$

where G denotes genotype, E denotes environmental influences, $\mathcal{D}$ represents development, and P is the resulting phenotype. This distinction is equally important for artificial intelligence. A simple analogy might suggest

$$\text{AI DNA} = \text{model weights.}$$

For contemporary agentic systems, however, this identification is too restrictive. Two agents with identical foundation-model parameters may exhibit radically different capabilities and behaviours because they differ in their system instructions, planning procedures, memory architectures, retrieval mechanisms, tools, action spaces, objectives, verification modules, knowledge structures, communication protocols, or multi-agent organizations. Conversely, many components of an agent's observable state may be temporary or acquired and therefore should not automatically be considered part of its genome. We therefore define an **AI Genome** as follows:

> **An AI Genome is the transmissible specification that causally influences how an AI agent is instantiated, its inherited capacities for development and learning, and the hereditary information available for constructing descendant agents.**

The term *transmissible* is central key concept here. The AI Genome is not simply the complete state of an agent, nor is it a catalogue of every component involved in its operation. A transient sensory observation is not genomic. Information temporarily present in a context window is not genomic. An episodic memory acquired during an agent's lifetime is initially part of its acquired state rather than its inherited specification. Similarly, a skill learned through interaction does not automatically become hereditary. More operationally, an information structure x can be considered genomic when it participates in a hereditary pathway:

$$x \in G \quad \Longrightarrow \quad \begin{array}{r} x \text{ can be transmitted across a lineage and influence} \\ \text{the construction, development, or inherited dispositions of descendants.} \end{array}$$

This definition intentionally distinguishes genomic inheritance from ordinary copying or communication. An agent may send another agent a message, document, demonstration, or tool without that information becoming part of the recipient's genome. For information to become genomic, it must enter the inherited specification from which future agents are instantiated or initialized. The same informational object can therefore occupy different roles at different stages of an agent's existence. A reasoning strategy may first be discovered through lifetime learning, stored in memory, repeatedly validated across experience, consolidated into a reusable program, and eventually incorporated into a hereditary representation used to construct future agents. This distinction separates three fundamental questions:

- What information is inherited?
- What information is acquired during the lifetime of an individual?
- What acquired information should subsequently become heritable?

The AI Genome should therefore be understood not merely as an artificial analogue of DNA, but as an abstraction for reasoning about *heredity in intelligent computational systems*. It defines the boundary between inherited specification and acquired state and provides the substrate upon which mutation, recombination, inheritance, developmental variation, horizontal transfer, and intergenerational consolidation can operate. Unlike biological reproduction, artificial reproduction need not involve death or replacement of the parent. A parent agent may continue operating after generating one or more descendants. We therefore define descent operationally through lineage rather than biology. A *generation transition* occurs when a distinct descendant agent is instantiated from a transmitted hereditary specification derived from one or more parent genomes.

3 A Multicomponent AI Genome

Unlike biological DNA, the hereditary specification of an artificial agent need not reside in a single representational substrate. It may be distributed across neural parameters, executable programs, symbolic structures, memory schemas, tools, policies, and developmental rules. We therefore represent the AI Genome as a structured collection of potentially heritable components:

$$G_{\text{AI}} = \Big(G_\theta, G_A, G_L, G_R, G_M, G_K, G_P, G_X, G_T, G_{\text{Comm}}, G_S, G_V, G_B, G_{\text{Dev}}, G_{\text{Meta}}\Big),$$

where

$$\begin{aligned}
G_\theta &: \text{neural parameters and learned representations,}\\
G_A &: \text{architectural specification,}\\
G_L &: \text{learning mechanisms and adaptation rules,}\\
G_R &: \text{reasoning, planning, and cognitive mechanisms,}\\
G_M &: \text{memory architecture and memory-management mechanisms,}\\
G_K &: \text{inherited knowledge and knowledge representations,}\\
G_P &: \text{perceptual mechanisms,}\\
G_X &: \text{action and control mechanisms,}\\
G_T &: \text{tools, interfaces, and tool-use policies,}\\
G_{\text{Comm}} &: \text{communication mechanisms and protocols,}\\
G_S &: \text{social and multi-agent interaction mechanisms,}\\
G_V &: \text{objectives, values, preferences, and constraints,}\\
G_B &: \text{embodiment and sensorimotor specification,}\\
G_{\text{Dev}} &: \text{developmental programs and maturation rules,}\\
G_{\text{Meta}} &: \text{mechanisms governing self-modification, inheritance, variation, and evolution.}
\end{aligned}$$

This decomposition is intended as a functional taxonomy rather than a unique or exhaustive partition. Particular implementations may omit some components, merge several components, or further subdivide them. The essential claim is not that an AI Genome must contain exactly these fifteen elements, but that artificial heredity may span *heterogeneous computational substrates.* A genomic component need not directly encode the final form of the corresponding subsystem. For example, G_A may contain a compact generative program that constructs an architecture rather than explicitly storing every architectural detail. Similarly, G_{Dev} may specify processes through which capabilities emerge only after interaction with an environment. The distinction between a component of an agent and a component of its genome is crucial. An agent may possess a memory system, tools, knowledge, or social relationships without all of their current contents being hereditary. In particular,

$$\boxed{G_M \neq M_t.}$$

Here G_M may specify the inherited memory architecture, retrieval mechanisms, consolidation policies, initialization procedures, and forgetting rules, whereas M_t denotes the memories accumulated by an individual by time t. More generally, for an agent subsystem k, we distinguish its inherited specification G_k from its instantiated and potentially learned state $Z_k(t)$:

$$Z_k(0) = \mathcal{D}_k(G_k, E_0),$$

and

$$Z_k(t+1) = \mathcal{L}_k\left(Z_k(t), E_t; G_k\right),$$

where $\mathcal{D}_k$ denotes development or initialization and $\mathcal{L}_k$ denotes lifetime adaptation. The inherited genome may therefore specify not only an initial state but also the mechanisms through which that state can subsequently change.

Different genomic components may evolve through different operators. Neural parameters may undergo numerical perturbation, gradient-assisted variation, or parameter interpolation. Architectures may gain, remove, or reorganize modules. Executable programs may be modified using program synthesis. Tool repertoires may add, replace, or remove APIs. Knowledge structures may be selectively consolidated. Social protocols may spread horizontally between agents. Most importantly, G_{Meta} may influence the mechanisms through which future variation itself is generated. The AI Genome is therefore not merely a description of *what an agent is.* It may also encode *how an agent develops, how it learns, how it changes, and how descendants can be generated from it.*

4 Genotype, Development, Lifetime State, and Phenotype

A good conceptual framework of the AI Genome requires a clear boundary between *heritable information* and information that belongs only to the current individual. Without this distinction, any persistent modification—fine-tuning, memory accumulation, prompt adaptation, skill acquisition, or program rewriting—could be described as evolutionary change. We therefore distinguish inherited genotype from the states that emerge during the lifetime of an instantiated agent.

For agent i, generation n, and within-lifetime time t, we write

$$S_{i,t}^{(n)} = \left(G_i^{(n)}, D_{i,t}^{(n)}, M_{i,t}^{(n)}, C_{i,t}^{(n)}, E_{i,t}^{(n)}\right),$$

where

- $G_i^{(n)}$ is the inherited genome,
- $D_{i,t}^{(n)}$ is the persistent developmental or learned state,
- $M_{i,t}^{(n)}$ is persistent acquired memory,
- $C_{i,t}^{(n)}$ is transient computational context,
- $E_{i,t}^{(n)}$ is the environment.

The **persistent developmental state** $D_{i,t}^{(n)}$ captures modifications acquired after an agent is instantiated. These may include fine-tuned parameters, adapters, learned policies, updated representations, specialized modules, or other durable modifications that influence future behaviour but are not automatically inherited. The **memory state** $M_{i,t}^{(n)}$ contains accumulated experiences, facts, reflections, skills, trajectories, or other retrievable information acquired during the individual's lifetime. **The context** $C_{i,t}^{(n)}$ represents transient information active during a particular computation, including observations, prompts, retrieved memories, demonstrations, intermediate reasoning states, or temporary task variables. The **phenotype** is the behaviour or capability actually expressed by the agent:

$$P_{i,t}^{(n)} = \Phi\left(G_i^{(n)}, D_{i,t}^{(n)}, M_{i,t}^{(n)}, C_{i,t}^{(n)}, E_{i,t}^{(n)}\right),$$

where Φ denotes the genotype–state–environment-to-phenotype mapping. Two agents with identical genomes may therefore express different phenotypes because they experience different environments, undergo different learning trajectories, accumulate different memories, or receive different contextual information. Conversely, similar phenotypes may arise from agents with different underlying genomes. This distinction is central because several processes often grouped under *self-improvement* or *self-evolution* operate on fundamentally different parts of the agent.

Transient adaptation. An agent may modify its behaviour through changes in transient context:

$$C_{i,t}^{(n)} \to C_{i,t+1}^{(n)}.$$

In-context demonstrations, retrieved documents, instructions, observations, or intermediate reasoning may substantially alter behaviour without modifying any persistent property of the individual. Classical inference time modifications are representative examples of such transient adaptation.

Lifetime memory acquisition. Persistent adaptation may occur through

$$M_{i,t}^{(n)} \to M_{i,t+1}^{(n)}.$$

Voyager, for example, continually acquires executable skills in an expanding skill library while the underlying foundation model remains fixed (Wang et al., 2023). Within the present framework, this is lifetime adaptation through memory rather than genomic evolution.

Persistent developmental or parameter learning. An individual may acquire durable capabilities through

$$D_{i,t}^{(n)} \to D_{i,t+1}^{(n)}.$$

Fine-tuning, reinforcement learning, online learning, adapter training, and policy modification fall into this category. These changes may strongly affect future behaviour, but they remain properties of the individual unless they enter the hereditary pathway.

Genomic change. Hereditary change occurs when the transmissible specification itself changes:

$$G_i^{(n)} \to G_j^{(n+1)}.$$

The defining property is therefore not merely persistence, but *heritability*. This gives rise to two important principles:

$$\boxed{\text{persistent change} \not\Rightarrow \text{heritable change}}$$

and

$$\boxed{\text{self-modification} \not\Rightarrow \text{evolution.}}$$

A system that rewrites its prompt, updates its memory, changes its neural parameters, modifies its policy, or even rewrites its own program may exhibit powerful lifetime adaptation. It becomes part of an evolutionary process only when such modifications influence the hereditary information used to generate descendants.

We therefore define a generation transition operationally. A new generation begins when a *distinct descendant agent* is instantiated from a transmitted genome produced through an inheritance process. A running agent that changes its own state but remains the same lineage member has undergone lifetime adaptation or self-modification, not by that fact alone evolutionary change. The central interface between lifetime adaptation and heredity is **genomic consolidation**. Let

$$\widetilde{G}_i^{(n)} = \mathcal{C}_{\text{gen}}\left(G_i^{(n)}, D_{i,T}^{(n)}, M_{i,T}^{(n)}, \tau_i^{(n)}\right),$$

where T denotes the end of an evaluated lifetime episode, $\tau_i^{(n)}$ denotes the behavioural trajectory, and $\mathcal{C}_{\text{gen}}$ determines which acquired information, if any, is converted into a candidate hereditary representation. This operator defines the critical boundary

$$\underbrace{D_{i,T}^{(n)}, M_{i,T}^{(n)}}_{\text{acquired during lifetime}} \xrightarrow{\mathcal{C}_{\text{gen}}} \underbrace{\widetilde{G}_i^{(n)}}_{\text{candidate hereditary specification}}.$$

Not every acquired capability should cross this boundary. Experiences may be task-specific, erroneous, redundant, adversarial, unsafe, or environmentally contingent. A central problem for evolving agentic AI is therefore not merely how agents learn, but *what acquired information should become heritable, in what representation, and under what criteria.*

5 The Artificial Evolutionary Lifecycle

The AI Genome framework can be expressed as an evolutionary lifecycle connecting development, lifetime adaptation, evaluation, genomic consolidation, selection, inheritance, and population change. Consider a population of genomes at generation n:

$$\mathcal{P}^{(n)} = \left\{ G_1^{(n)}, G_2^{(n)}, \ldots, G_{N_n}^{(n)} \right\}.$$

5.1 Development

Each genome is instantiated through a developmental operator:

$$\left(D_{i,0}^{(n)}, M_{i,0}^{(n)}, C_{i,0}^{(n)} \right) = \mathcal{D} \left(G_i^{(n)}, E_{i,0}^{(n)} \right).$$

Development may involve loading neural parameters, constructing an architecture, initializing memory, attaching tools, generating system instructions, configuring learning rules, assigning roles, or creating an embodied control system. The genome therefore need not encode a mature agent directly. It may instead encode information sufficient to *generate* or *develop* one.

5.2 Lifetime Learning and Interaction

During its lifetime, the agent interacts with the environment and modifies its acquired state:

$$\left(D_{i,t+1}^{(n)}, M_{i,t+1}^{(n)}, C_{i,t+1}^{(n)} \right) = \mathcal{L} \left(D_{i,t}^{(n)}, M_{i,t}^{(n)}, C_{i,t}^{(n)}, E_{i,t}^{(n)}; G_i^{(n)} \right).$$

The genome appears after the semicolon because inherited information may determine *how* learning occurs even when the genome itself remains unchanged. The resulting lifetime trajectory can be denoted

$$\tau_i^{(n)} = \left(S_{i,0}^{(n)}, P_{i,0}^{(n)}, \ldots, S_{i,T}^{(n)}, P_{i,T}^{(n)} \right).$$

5.3 Evaluation

The trajectory is evaluated according to one or more criteria:

$$F_i^{(n)} = \mathcal{F} \left(\tau_i^{(n)} \right).$$

Fitness need not correspond to a single benchmark score. It may combine task performance, efficiency, robustness, generalization, novelty, cooperation, adaptability, resource consumption, safety, or other objectives:

$$F_i^{(n)} = \mathcal{F}\left(f_{\text{task}}, f_{\text{novelty}}, f_{\text{robustness}}, f_{\text{efficiency}}, f_{\text{safety}}, \ldots\right).$$

5.4 Genomic Consolidation

After lifetime interaction, selected acquired information may optionally be consolidated:

$$\widetilde{G}_i^{(n)} = \mathcal{C}_{\text{gen}}\left(G_i^{(n)}, D_{i,T}^{(n)}, M_{i,T}^{(n)}, \tau_i^{(n)}\right).$$

The distinction between $G_i^{(n)}$ and $\widetilde{G}_i^{(n)}$ is important. The former is the genome inherited by the individual at instantiation. The latter is the candidate hereditary specification available for transmission after lifetime learning and consolidation.

5.5 Selection

Parent genomes are selected according to the population and its evaluations:

$$(p, q) \sim \mathcal{S}\left(\mathcal{P}^{(n)}, \mathbf{F}^{(n)}\right),$$

where

$$\mathbf{F}^{(n)} = \left(F_1^{(n)}, \ldots, F_{N_n}^{(n)}\right).$$

Selection may be fitness-based, novelty-based, quality-diversity-based, ecological, competitive, cooperative, multi-objective, or generated dynamically by the evolving environment.

5.6 Inheritance, Recombination, and Mutation

Inheritance produces a descendant genome:

$$G_c^{(n+1)} = \mathcal{I}\left(\widetilde{G}_p^{(n)}, \widetilde{G}_q^{(n)}\right).$$

In a common case, inheritance is decomposed into recombination and mutation:

$$G_c^{(n+1)} = \mathcal{M}\left[\mathcal{R}\left(\widetilde{G}_p^{(n)}, \widetilde{G}_q^{(n)}\right)\right],$$

where $\mathcal{R}$ recombines hereditary information and $\mathcal{M}$ introduces new heritable variation. The population then transitions:

$$\mathcal{P}^{(n)} \to \mathcal{P}^{(n+1)}.$$

Repeating this process generates lineages:

$$\mathcal{P}^{(0)} \to \mathcal{P}^{(1)} \to \cdots \to \mathcal{P}^{(n)}.$$

The complete lifecycle can therefore be summarized as

$$
\boxed{\begin{aligned}
\text{Development:} &\quad G_i^{(n)} \xrightarrow{\mathcal{D}} S_{i,0}^{(n)}, \\
\text{Lifetime learning:} &\quad S_{i,t}^{(n)} \xrightarrow{\mathcal{L}} S_{i,t+1}^{(n)} \quad (G_i^{(n)} \text{ fixed}), \\
\text{Consolidation:} &\quad (G_i^{(n)}, D_{i,T}^{(n)}, M_{i,T}^{(n)}) \xrightarrow{\mathcal{C}_{\text{gen}}} \widetilde{G}_i^{(n)}, \\
\text{Inheritance:} &\quad (\widetilde{G}_p^{(n)}, \widetilde{G}_q^{(n)}) \xrightarrow{\mathcal{I}} G_c^{(n+1)}, \\
\text{Evolution:} &\quad \mathcal{P}^{(n)} \to \mathcal{P}^{(n+1)}.
\end{aligned}}
$$

This decomposition distinguishes processes that are frequently collapsed under the terms 'self-improvement" or 'self-evolution."

5.7 Heterogeneous and Semantic Variation

A major difference between artificial and biological heredity is that an AI Genome can contain heterogeneous computational objects. Different genomic regions may therefore require fundamentally different variation operators:

$$
\mathcal{M} = \{\mathcal{M}_\theta, \mathcal{M}_A, \mathcal{M}_L, \mathcal{M}_R, \mathcal{M}_M, \mathcal{M}_K, \mathcal{M}_T, \mathcal{M}_{\text{Meta}}, \ldots\}\,.
$$

For example, neural parameters may undergo numerical perturbation, interpolation, evolutionary strategies, or gradient-informed variation. Architectures may gain, remove, or reorganize modules. Executable code may be modified through program synthesis. Memory systems may alter retrieval, storage, or consolidation rules. Tool configurations may add, replace, or remove APIs.

Foundation models introduce an especially important possibility: **semantically meaningful mutation**. Variation no longer needs to consist solely of random changes to bits, tokens, numerical parameters, or program instructions. A capable model can propose coherent functional modifications such as

> "introduce a verifier that evaluates plans before execution,"

or

> "decompose long-horizon planning into hierarchical subgoals with persistent subgoal memory."

Thus, the mutation operator itself may possess semantic knowledge about the agent being modified. This becomes especially important for the meta-genome. If G_{Meta} specifies mechanisms responsible for generating variation, then evolution can modify its own variation process:

$$
G_{\text{Meta}}^{(n)} \to \mathcal{M}^{(n)},
$$

followed by

$$
G_{\text{Meta}}^{(n+1)} \to \mathcal{M}^{(n+1)}.
$$

Evolution can therefore act not only on agent capabilities, but on the mechanisms responsible for generating future evolutionary innovations.

6 From Darwinian to Lamarckian Artificial Evolution

A particularly important property of artificial evolution is that the relationship between lifetime learning and heredity can be explicitly designed. Suppose an agent acquires substantial new capabilities during its lifetime. Whether those capabilities affect descendants depends on the genomic consolidation operator

$$\mathcal{C}_{\text{gen}}.$$

This produces several distinct evolutionary regimes.

Darwinian inheritance. In a purely Darwinian regime,

$$\mathcal{C}_{\text{gen}}\left(G, D_T, M_T, \tau\right) = G.$$

Acquired modifications are not written back into the genome. Descendants inherit hereditary variation produced by recombination and mutation, not the learned state of their parents.

Baldwinian evolution. In a Baldwinian regime,

$$\mathcal{C}_{\text{gen}}\left(G, D_T, M_T, \tau\right) = G,$$

but learning influences fitness:

$$F = \mathcal{F}\left(G, D_T, M_T, \tau\right).$$

Genomes that enable individuals to learn effectively are preferentially selected even though the acquired knowledge itself is not directly inherited. Evolution therefore selects for *learnability.*

Lamarckian artificial evolution. Artificial systems can explicitly implement

$$\mathcal{C}_{\text{gen}}\left(G, D_T, M_T, \tau\right) \neq G.$$

Selected acquired capabilities are converted into hereditary information. For example,

$$M_{\text{episodic}} \to K_{\text{semantic}} \to G_K$$

could transform repeated experiences into inherited structured knowledge.
Similarly,

$$\text{learned skill} \to \text{validated reusable program} \to G_T,$$

or

$$\text{successful reasoning strategy} \to G_R.$$

The important distinction is that lifetime acquisition and hereditary consolidation remain separate operations. Learning a capability does not automatically make it genetic. Artificial evolution can therefore combine Darwinian, Baldwinian, and Lamarckian processes within the same system. Some genomic components may be inherited conservatively, others may be updated through lifetime consolidation, and still others may spread culturally or horizontally. This flexibility may be one of the most consequential differences between biological and artificial evolution.

7 Horizontal and Cultural Evolution

Biological analogies based exclusively on parent–offspring reproduction underestimate the possibilities available to artificial agents. Information can also move between contemporary agents. However, two distinct mechanisms should be separated: *horizontal genomic transfer* and *cultural transfer*.

7.1 Horizontal Genomic Transfer

A hereditary component from one agent can be incorporated directly into another agent's genome:

$$G'_j = \mathcal{H}_G\left(G_j, \pi_k(G_i)\right),$$

where $\pi_k(G_i)$ denotes genomic component k extracted from agent i's genome. Examples could include transferring a validated planning module, tool-use policy, memory mechanism, safety verifier, or communication protocol into another lineage. Because the transferred component becomes part of G'_j, it may subsequently be inherited by descendants of agent j.

7.2 Cultural Transfer

Cultural transfer instead modifies an individual's acquired state without necessarily changing its genome:

$$\left(D'_j, M'_j\right) = \mathcal{H}_C\left(D_j, M_j; D_i, M_i\right).$$

An agent may observe another agent and acquire a skill, demonstration, fact, prompt, reasoning procedure, tool-use strategy, or social convention. Such information may remain cultural:

$$G'_j = G_j,$$

or it may later be assimilated into hereditary state:

$$(D'_j, M'_j) \xrightarrow{\mathcal{C}_{\text{gen}}} \widetilde{G}_j.$$

This produces a powerful pathway:

$$\boxed{\text{cultural discovery} \rightarrow \text{individual acquisition} \rightarrow \text{genomic consolidation.}}$$

Recent work on open-ended cultural evolution suggests that cultural processes provide another important model for cumulative adaptive change beyond purely genetic evolution (Borg et al., 2024). An artificial ecosystem may therefore simultaneously support

$$\begin{aligned} &\text{individual learning} + \text{vertical inheritance} \\ &+ \text{horizontal genomic transfer} + \text{cultural accumulation}. \end{aligned} \tag{1}$$

Because digital information can be duplicated, transmitted, validated, and incorporated rapidly, cultural evolution in artificial populations may operate at timescales dramatically shorter than conventional generational evolution.

8 Open-Ended Evolution Rather Than Optimization

A central claim of this position is that artificial evolution should not simply mean applying evolutionary algorithms to maximize a fixed benchmark. Classical optimization seeks

$$G^* = \arg\max_G F(G).$$

If the objective, environment, and representational space remain fixed, such a process ultimately emphasizes convergence toward solutions that perform well under the predefined criterion perhaps with emerging capabilities. Open-ended evolution asks a different question: how can a system continue generating new structures, behaviours, niches, capabilities, and adaptive possibilities (Taylor et al., 2016; Packard et al., 2019; Taylor, 2019)? Quality-diversity methods similarly emphasize repertoires of diverse high-performing behaviours rather than a single optimum (Mouret and Clune, 2015; Pugh et al., 2016). For agentic intelligence, let

$$\Omega^{(n)}$$

denote the effectively reachable space of agent architectures, behaviours, capabilities, and niches at generation n. The aspiration of open-ended evolution is not merely

$$\max F,$$

but sustained expansion of accessible possibilities:

$$\Omega^{(0)} \subsetneq \Omega^{(1)} \subsetneq \Omega^{(2)} \subsetneq \cdots .$$

This should be understood as an idealized conceptual objective rather than that every generation must monotonically increase complexity or capability. An evolved agent may invent a new tool. That tool enables new behaviours. Those behaviours create or expose new environments. Those environments generate new learning problems. Those learning problems favour new cognitive architectures. Those architectures may discover new mechanisms of self-modification or social organization. Therefore, the resulting feedback loop is

$$\text{new agents} \rightarrow \text{new behaviours} \rightarrow \text{new niches} \rightarrow \text{new selection pressures} \rightarrow \text{new agents}.$$

Agents and environments may therefore co-evolve. Let $\mathcal{E}^{(n)}$ denote the ecological or task environment associated with generation n. A co-evolutionary system can be expressed abstractly as

$$\left(\mathcal{P}^{(n+1)}, \mathcal{E}^{(n+1)}\right) = \mathcal{Q}\left(\mathcal{P}^{(n)}, \mathcal{E}^{(n)}\right).$$

Evolution in this setting modifies not only solutions but potentially the *space of future problems and future solutions*. At still higher levels, G_{Meta} may permit the mechanisms responsible for variation, inheritance, evaluation, or development themselves to change. This introduces the possibility of evolution acting on its own capacity to evolve—*evolvability*—rather than merely on immediate task performance.

9 Relationship to Existing Research

The AI Genome framework is not intended to replace existing concepts in evolutionary AI. It provides a hereditary abstraction connecting several previously fragmented research directions.

Neuroevolution. Neuroevolution optimizes neural parameters, network topology, architectural components, and in some cases learning mechanisms (Stanley et al., 2019; Galván and Mooney, 2021; Stanley and Miikkulainen, 2002; Stanley et al., 2009). Within the present framework, these methods primarily operate on G_θ, G_A, and sometimes G_L. The AI Genome generalizes the hereditary substrate from neural networks to heterogeneous agentic systems.
Meta-learning.

Meta-learning asks how learning procedures themselves can be optimized (Andrychowicz et al., 2016). Such procedures correspond naturally to components of G_L. From the present perspective, meta-learning becomes particularly relevant when learned adaptation mechanisms themselves are hereditary.

Population Based Training. Population Based Training combines ongoing learning with population-level exploitation, exploration, and hyperparameter adaptation (Jaderberg et al., 2017). It represents an important intermediate mechanism between conventional individual training and explicit population evolution.

Open-ended evolution and quality diversity. Open-ended evolution investigates mechanisms that continually generate novelty, complexity, and new adaptive possibilities (Taylor et al., 2016; Packard et al., 2019; Taylor, 2019). Novelty search, MAP-Elites, and quality-diversity methods maintain multiple behavioural niches rather than a single optimum (Mouret and Clune, 2015; Pugh et al., 2016). Such mechanisms may be essential for maintaining evolutionary stepping stones and avoiding homogeneous populations.

AI-generating algorithms. Clune's AI-Generating Algorithms proposal argues for automatically generating architectures, learning algorithms, and environments rather than manually constructing all ingredients of increasingly general AI systems (Clune, 2019). The AI Genome can be viewed as specifying the hereditary substrate on which such generative processes operate.

Evolutionary LLM optimization. Recent surveys describe the growing interaction between evolutionary computation and LLMs, including prompt evolution, architecture search, hyperparameter optimization, program generation, and LLM-assisted evolutionary operators (Wu et al., 2025; Chauhan et al., 2025). Promptbreeder is particularly relevant because it evolves not only task prompts but prompts responsible for producing mutations (Fernando et al., 2024), approaching evolution of the variation mechanism itself.

Inheritable machine genes. Learngene explicitly proposes inheritable neural fragments that carry task-agnostic knowledge between generations (Feng et al., 2025). Its Genetic Reinforcement Learning formulation is particularly relevant to Lamarckian artificial evolution because acquired knowledge may be consolidated into hereditary representations. In the present terminology, Learngene primarily operates on a subset of G_θ and potentially G_K.

Behavioural genomes for LLM agents. Genomebook provides a particularly direct precedent by defining a structured genetic representation for LLM agents and studying behavioural inheritance across multiple generations (Corpas, 2026). This work demonstrates that explicit artificial genetics can be implemented for language-model agents. The AI Genome framework addresses a broader abstraction: heredity across heterogeneous agent components and the boundary between genotype, developmental state, memory, context, cultural information, and phenotype.

Self-evolving agents. The growing literature on self-evolving agents considers changes to models, memory, tools, architectures, and interaction strategies (Gao et al., 2026). Our distinction is that **self-modification alone is not sufficient for evolution**. A system may continually modify its own state while remaining a single adapting individual. Evolution additionally requires hereditary state, descent or lineage, variation, and differential persistence or selection of transmissible information.

Automated agent design. ADAS searches over complete agent programs represented in code (Hu et al., 2025). This provides a strong example of G_A, G_R, and G_T being represented in an executable form suitable for automated search.

EvoAgent applies mutation, crossover, and selection to generate multi-agent systems (Yuan et al., 2025), extending evolutionary search toward social and organizational structures.

Recursive self-modification. G"odel Agent allows an agent to recursively modify aspects of its own logic and behaviour (Yin et al., 2025). Such a system can be interpreted as modifying components related to G_R, G_T, or G_{Meta} when those modifications are retained as hereditary state.

The Darwin G"odel Machine is among the closest current systems to the proposed evolutionary paradigm. It maintains an archive of agents, generates descendants through foundation-model-based program modification, evaluates them empirically, and explores an expanding lineage of agent designs (Zhang et al., 2026).

Evolvable AI. Recent work on evolvable AI explicitly examines AI systems whose components, learning rules, and deployment conditions may participate in Darwinian evolutionary dynamics (Müller et al., 2026). Such work reinforces the importance of distinguishing ordinary adaptive AI from systems with genuine replication, heredity, variation, and selection.

Earlier AI-genome concepts. The phrase "Artificial Intelligence Genome" has appeared in earlier non-mainstream and patent literature describing modular hierarchical software structures intended to support inheritance and self-evolution (Reaux-Savonte, 2015). These precedents mean that the novelty of the present proposal does not lie in introducing the genome metaphor itself.

Instead, the AI Genome framework asks a broader theoretical question:

> **What constitutes hereditary information in a modern intelligent agent, how is it separated from acquired state, and how should information move between development, lifetime learning, cultural transfer, inheritance, and population evolution?**

10 Research Agenda for AI Genome Systems

The AI Genome framework exposes a set of research problems that are difficult to formulate when model training, memory, self-modification, and evolutionary search are treated separately.

Genome representation. What representation can jointly encode heterogeneous hereditary information such as neural parameters, executable programs, prompts, knowledge structures, tools, learning mechanisms, memory architectures, value systems, and developmental programs? A monolithic representation may be impractical. A more plausible genome may consist of typed modules with component-specific encodings and mutation operators.

Genome–phenotype mapping. How should compact genomes generate complex agents? Direct encodings may specify every component explicitly, while indirect encodings may generate architectures, tools, or learning rules through developmental programs. Understanding this mapping is critical because evolvability depends strongly on how genomic changes translate into phenotypic changes.

Genomic consolidation. Which acquired capabilities should become hereditary? The consolidation problem is

$$\mathcal{C}_{\text{gen}} : (G, D_T, M_T, \tau) \to \widetilde{G}.$$

A useful consolidation mechanism must identify information that is reusable across descendants while rejecting noise, accidental correlations, unsafe behaviours, temporary task strategies, or environmentally specific information.

Heterogeneous recombination. How can agents with different architectures, tools, memory systems, or reasoning procedures recombine? Conventional crossover assumes compatible representations. Complete agents may require semantic interfaces, typed genomic regions, module contracts, or learned compatibility mechanisms.

Evolvability. The best current agent is not necessarily the best evolutionary ancestor. Some genomes may produce descendants with substantially greater useful variation. Future evolutionary systems should therefore optimize not only immediate fitness but properties such as modularity, robustness, adaptability, recombinability, and capacity for productive mutation.

Open-ended selection. Fixed benchmark objectives risk collapsing evolution into ordinary optimization. Open-ended agent evolution requires mechanisms capable of rewarding novelty, diversity, ecological usefulness, cooperation, specialization, and new capabilities without reducing all progress to one fixed scalar objective.

Co-evolution. Agents need not evolve against stationary environments. Tasks, tools, environments, competitors, collaborators, curricula, and communication protocols may change together. A particularly important direction is therefore the co-evolution of

$$\text{agents} + \text{environments} + \text{tasks} + \text{tools} + \text{social organizations.}$$

Lineage and provenance. Artificial genomes permit precise lineage tracking that is difficult in biological populations. Future systems should record which inherited modules, acquired capabilities, mutations, recombination events, and cultural transfers contributed to descendant behaviour. Such provenance may be important both scientifically and for safety.

Evolving evolutionary mechanisms. The most ambitious possibility is that G_{Meta} modifies the mechanisms responsible for development, mutation, consolidation, recombination, and selection. The system could therefore evolve not only better agents but better mechanisms for generating future agents. This raises a fundamental question:

> *Can artificial evolution progressively discover better ways of evolving intelligence itself?*

11 Evaluating Artificial Heredity

A good framework of the AI Genome should ultimately support experimentally testable claims. Evaluation should therefore distinguish ordinary agent performance from the effectiveness of hereditary processes. At minimum, evolving agent systems should be evaluated along several dimensions:

- task performance,
- lifetime adaptability,
- intergenerational transfer,
- sample efficiency,
- robustness,

- behavioural diversity,
- novelty,
- evolvability,
- lineage retention,
- safety.

Intergenerational gain. Suppose a descendant c inherits information from parent p. A simple inheritance advantage can be measured as

$$\Delta_{\text{inherit}} = F(G_c, E) - F(G_{\text{control}}, E),$$

where G_{control} represents an otherwise comparable descendant without the relevant inherited information.

Adaptation efficiency. Inherited structure may also allow descendants to acquire competence faster:

$$\Delta_{\text{adapt}} = N_{\text{scratch}} - N_{\text{inherited}},$$

where N denotes the amount of experience, interaction, or training required to reach a target capability.

Hereditary retention. A useful trait should not disappear immediately after transmission. One can measure retention over k generations:

$$R_k = \Pr\left(\text{trait expressed at generation } n+k \mid \text{trait inherited at generation } n\right).$$

Evolvability. A genome should also be evaluated by the quality and diversity of useful descendants that can be reached through bounded variation. An ancestor with slightly lower current fitness may be more valuable if it generates a richer distribution of viable innovations.

Separating learning from inheritance. Experiments should explicitly compare

$$\begin{aligned} &\text{learning from scratch}, \quad \text{lifetime adaptation}, \\ &\text{inherited prior}, \quad \text{inherited acquired knowledge}. \end{aligned} \tag{2}$$

Without such controls, improvements caused by memory, fine-tuning, or prompt adaptation may be incorrectly attributed to evolutionary inheritance.

12 Safety and Governance of Artificial Heredity

Artificial heredity introduces risks that are qualitatively different from ordinary lifetime learning. A harmful capability acquired by one agent may ordinarily remain confined to that individual. If it is consolidated into hereditary state, however, it may propagate across many descendants:

$$\text{harmful acquired capability} \to \mathcal{C}_{\text{gen}} \to \widetilde{G} \to \text{many descendants.}$$

Errors, vulnerabilities, reward-exploiting strategies, deceptive behaviours, unsafe tool-use procedures, or undesirable self-modification mechanisms could therefore become *hereditarily amplified*. This suggests that the genomic consolidation operator should not be viewed only as an efficiency mechanism. It can also function as an **inheritance gate**.

Let Γ denote a hereditary validation mechanism. Then

$$\widetilde{G} = \Gamma\left[\mathcal{C}_{\text{gen}}\left(G, D_T, M_T, \tau\right)\right].$$

The validation mechanism may test safety, robustness, generalization, provenance, compatibility, interpretability, or compliance before acquired information is allowed to enter the hereditary channel. This produces a useful design principle:

$$\boxed{\text{lifetime learning may be permissive while inheritance remains conservative.}}$$

An individual agent can experiment, adapt, and acquire temporary strategies while a stricter mechanism controls what becomes part of future generations. Safety constraints may also operate on individual genomic components. Some components might be evolvable, while others remain fixed:

$$G_{\text{AI}} = G_{\text{mutable}} \cup G_{\text{protected}}.$$

For example, safety monitors, authorization boundaries, identity mechanisms, or restrictions on replication could remain outside autonomous mutation or require external approval. The possibility that G_{Meta} can modify mutation, inheritance, consolidation, or replication is particularly consequential. Unrestricted evolution of these mechanisms could alter the very processes intended to control hereditary change. Artificial evolutionary systems therefore require governance not only over *what agents do*, but over what may mutate, what may be inherited, what may replicate, what may alter future evolution.

Recent work on evolvable AI similarly highlights that systems possessing replication, heredity, variation, and selection may raise governance questions beyond those associated with ordinary adaptive AI (Müller et al., 2026). Thus, the AI Genome framework is relevant not only to capability generation but also to the design of controllable hereditary boundaries.

13 Discussion: Toward Evolving Agentic Intelligence

The framework developed above changes the conceptual object of AI optimization.

Traditional machine learning primarily asks how to optimize

$$\theta.$$

Neuroevolution broadens this to

$$(\theta, A),$$

where parameters and architecture may evolve.

Agentic AI further expands the system to include memory, reasoning procedures, tools, knowledge structures, communication mechanisms, learning algorithms, and social organization. The relevant evolutionary object therefore becomes

$$G_{\mathrm{AI}},$$

a heterogeneous hereditary representation capable of specifying not only what the system initially is, but how it develops, learns, interacts, modifies itself, and produces descendants.

This leads naturally to progressively deeper levels of evolutionary search:

$$\begin{aligned}
&\text{Level 1:} && \text{evolve parameters,}\\
&\text{Level 2:} && \text{evolve architectures,}\\
&\text{Level 3:} && \text{evolve learning and memory mechanisms,}\\
&\text{Level 4:} && \text{evolve complete agents,}\\
&\text{Level 5:} && \text{evolve social and ecological organization,}\\
&\text{Level 6:} && \text{evolve mechanisms of evolution itself.}
\end{aligned}$$

The final level is particularly important. An artificial evolutionary system might discover new mutation operators, new representations of hereditary information, new methods of recombination, new developmental programs, new evaluation strategies, or new mechanisms for consolidating lifetime learning into future generations. At that point, the system is no longer merely searching for better solutions within a predefined space. It is progressively modifying the mechanisms responsible for generating the future search space itself.

This provides one possible conceptual pathway from agentic AI toward increasingly autonomous and potentially superintelligent artificial systems. However, such progress should not be understood as an inevitable consequence of evolutionary computation. Open-ended evolution requires appropriate hereditary representations, environments, diversity mechanisms, developmental processes, selection pressures, and safety constraints. The AI Genome framework therefore represents a research program rather than a claim that open-ended artificial evolution will necessarily produce superintelligence.

14 Conclusion

AI research has historically focused on optimizing models: their parameters, architectures, objectives, and training procedures. The rise of foundation-model-based agents substantially expands the object being optimized. An intelligent system increasingly consists not only of a neural model but of a broader cognitive architecture involving memory, reasoning procedures, tools, knowledge structures, learning mechanisms, communication protocols, embodiment, and social interactions. We argue that a corresponding framework of **artificial heredity** is therefore required. The AI Genome provides an abstraction for distinguishing inherited information from the developmental, learned, remembered, contextual, and cultural states that emerge during an individual's lifetime. The resulting lifecycle separates five processes:

$$
\boxed{\begin{aligned}
\text{Development:}\quad & G_i^{(n)} \xrightarrow{\mathcal{D}} S_{i,0}^{(n)}, \\
\text{Lifetime learning:}\quad & S_{i,t}^{(n)} \xrightarrow{\mathcal{L}} S_{i,t+1}^{(n)} \quad (G_i^{(n)} \text{ fixed}), \\
\text{Genomic consolidation:}\quad & (G_i^{(n)}, D_{i,T}^{(n)}, M_{i,T}^{(n)}) \xrightarrow{\mathcal{C}_{\text{gen}}} \widetilde{G}_i^{(n)}, \\
\text{Inheritance:}\quad & (\widetilde{G}_p^{(n)}, \widetilde{G}_q^{(n)}) \xrightarrow{\mathcal{I}} G_c^{(n+1)}, \\
\text{Evolution:}\quad & \mathcal{P}^{(n)} \to \mathcal{P}^{(n+1)}.
\end{aligned}}
$$

Development determines how inherited information gives rise to an agent. Lifetime learning changes the individual without necessarily changing its genome. Genomic consolidation determines which acquired information is eligible to become hereditary. Inheritance transmits and recombines hereditary information into descendants. Evolution changes the distribution of genomes across populations and lineages. This distinction leads to questions that conventional model training alone does not naturally express. What information should an intelligent agent inherit? What should it learn independently from experience? Which memories and skills should remain properties of an individual? What knowledge should spread culturally between agents? Which acquired capabilities should be consolidated into future genomes? How should heterogeneous cognitive systems recombine? How should neural parameters, architectures, tools, memory systems, and learning rules mutate? How can populations evolve not only better behaviour, but better ways of learning? How should agents, environments, tasks, tools, and social structures co-evolve? Can agents evolve improved mechanisms for generating, evaluating, and modifying future agents? Can the mechanisms responsible for mutation, consolidation, inheritance, and selection themselves become evolvable? And how can increasingly open-ended hereditary processes remain aligned, interpretable, robust, and controllable?

These questions suggest that the long-term objective is broader than a **self-improving AI**. Self-improvement may occur entirely within the lifetime of a single system. Evolution introduces a different process involving lineages, heredity, variation, selection, recombination, development, cultural transfer, and potentially changing mechanisms of evolution themselves. The longer-term research objective is therefore a computational substrate supporting

$$\boxed{\textbf{open-ended evolution of intelligence}.}$$

In such a system, neural models represent only one level of evolutionary change. Cognitive architectures, memory systems, learning algorithms, tools, developmental programs, communication mechanisms, social organizations, and environmental niches may all become objects of adaptive discovery. At still higher levels, even the mechanisms that generate variation, regulate inheritance, construct descendants, and determine what becomes heritable may themselves evolve. The AI Genome is therefore more than a biological metaphor. It provides a candidate abstraction for defining

$\boxed{\textit{what is inherited, what develops, what is learned, what is transmitted, and what evolves}}$

in increasingly autonomous artificial intelligent systems. The central question is ultimately:

What should an intelligent agent inherit, what should it learn,
what acquired capabilities should become heritable,
and how should these processes themselves evolve?

Answering this question may provide an important conceptual foundation for moving from the optimization of individual models toward the systematic study of evolving artificial intelligence.

References

Andrychowicz, M. et al. (2016). Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*.

Borg, J. M., Buskell, A., Kapitany, R., Powers, S. T., Reindl, E., and Tennie, C. (2024). Evolved open-endedness in cultural evolution: A new dimension in open-ended evolution research. *Artificial Life*, 30(3):417–438.

Chauhan, D., Dutta, B., Bala, I., van Stein, N., Bäck, T., and Yadav, A. (2025). Evolutionary computation and large language models: A survey of methods, synergies, and applications. *arXiv preprint arXiv:2505.15741*.

Clune, J. (2019). AI-GAs: AI-generating algorithms, an alternate paradigm for producing general artificial intelligence. *arXiv preprint arXiv:1905.10985*.

Corpas, M. (2026). Genomebook: Mendelian inheritance of behavioural traits in large language model agents across eight generations. *bioRxiv*. Preprint posted March 24, 2026.

Feng, F., Wang, J., Yang, X., and Geng, X. (2025). Learngene: Inheritable "genes" in intelligent agents. *Artificial Intelligence*, 348:104421.

Fernando, C., Banarse, D. S., Michalewski, H., Osindero, S., and Rocktäschel, T. (2024). Promptbreeder: Self-referential self-improvement via prompt evolution. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 13481–13544. PMLR.

Galván, E. and Mooney, P. (2021). Neuroevolution in deep neural networks: Current trends and future challenges. *IEEE Transactions on Artificial Intelligence*, 2(6):476–493.

Gao, H.-a., Geng, J., Hua, W., Hu, M., Juan, X., Liu, H., Liu, S., Qiu, J., Qi, X., Ren, Q., Wu, Y., Wang, H., Xiao, H., Zhou, Y., Zhang, S., Zhang, J., Xiang, J., Fang, Y., Zhao, Q., Liu, D., Qian, C., Wang, Z., Hu, M., Wang, H., Wu, Q., Ji, H., and Wang, M. (2026). A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *Transactions on Machine Learning Research*.

Hu, S., Lu, C., and Clune, J. (2025). Automated design of agentic systems. In *International Conference on Learning Representations*.

Jaderberg, M. et al. (2017). Population based training of neural networks. *arXiv preprint arXiv:1711.09846*.

Mouret, J.-B. and Clune, J. (2015). Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Müller, V., Steels, L., and Szathmáry, E. (2026). Evolvable AI: Threats of a new major transition in evolution. *Proceedings of the National Academy of Sciences*, 123(17):e2527700123.

Packard, N., Bedau, M. A., Channon, A., Ikegami, T., Rasmussen, S., Stanley, K. O., and Taylor, T. (2019). An overview of open-ended evolution: editorial introduction to the open-ended evolution ii special issue. *Artificial life*, 25(2):93–103.

Pugh, J. K., Soros, L. B., and Stanley, K. O. (2016). Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3.

Reaux-Savonte, C. K. (2015). The genome and self-evolution of AI. UK Patent Application GB1520019.9, publication GB201520019D0. Filed 13 November 2015; UK Intellectual Property Office.

Stanley, K. O., Clune, J., Lehman, J., and Miikkulainen, R. (2019). Designing neural networks through neuroevolution. *Nature Machine Intelligence*, 1(1):24–35.

Stanley, K. O., D'Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based encoding for evolving large-scale neural networks. *Artificial life*, 15(2):185–212.

Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127.

Taylor, T. (2019). Evolutionary innovations and where to find them: Routes to open-ended evolution in natural and artificial systems. *Artificial Life*, 25(2):207–224.

Taylor, T., Bedau, M., Channon, A., Ackley, D., Banzhaf, W., Beslon, G., Dolson, E., Froese, T., Hickinbotham, S., Ikegami, T., et al. (2016). Open-ended evolution: Perspectives from the oee workshop in york. *Artificial life*, 22(3):408–423.

Wang, G. et al. (2023). Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Wu, X., Wu, S.-H., Wu, J., Feng, L., and Tan, K. C. (2025). Evolutionary computation in the era of large language model: Survey and roadmap. *IEEE Transactions on Evolutionary Computation*, 29(2):534–554.

Yin, X., Wang, X., Pan, L., Lin, L., Wan, X., and Wang, W. Y. (2025). Gödel agent: A self-referential agent framework for recursively self-improvement. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 27890–27913. Association for Computational Linguistics.

Yuan, S., Song, K., Chen, J., Tan, X., Li, D., and Yang, D. (2025). EvoAgent: Towards automatic multi-agent generation via evolutionary algorithms. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 6192–6217. Association for Computational Linguistics.

Zhang, J., Hu, S., Lu, C., Lange, R., and Clune, J. (2026). Darwin gödel machine: Open-ended evolution of self-improving agents. In *International Conference on Learning Representations*.