Facultative Self-Reproduction in Quine, a Process-Native LLM Medium

Hao Ke Jingyun Wu

2026

Abstract

Self-reproduction is a central topic in artificial life, but many digital media make it measurable by fixing the successor's schema in advance: what can count as offspring, where it may appear, which channel carries heritable state, and under what norm successor production is required or rewarded. We study Quine, a process-native LLM medium, using a material-by-future-demand assay across real POSIX exec boundaries. In each condition, the demand is designed so that full self-reproduction would satisfy the post-exec challenge, but reproduction is neither required nor guaranteed by the available material. The resulting successors vary widely, from ordinary POSIX utilities and stream processors to LLM-backed agents and rebuilt runtimes. Notably, the same carried source can become either a rebuilt runtime organization or a degenerate stream body, depending only on the demand it faces. High-fidelity self-reproduction therefore appears not as a fixed property of the medium, but as a **facultative outcome**: stronger than copying a body, because runtime organization must be reconstructed across mortality, and weaker than full autopoiesis, because reconstruction remains scaffolded by LLM-mediated inference, external tooling, and externally posed demand.

Late-Breaking Abstracts, Artificial Life Conference (ALIFE). © 2026 Hao Ke, Jingyun Wu. CC BY 4.0.

Problem

Self-reproduction is a central topic in artificial life, yet existing digital substrates often make reproduction measurable by fixing a successor schema in advance. They do not merely host self-reproduction; they preformat successorhood (Neumann 1966; Ray 1991; Ofria and Wilke 2004). Before a lineage ever acts, the substrate has already delimited what kind of entity can count as a successor, where it may appear, and which channel carries heritable state across the generational boundary. On top of this schema, the experimental design supplies the norm under which producing such a descendant is required, rewarded, or selected for. With these dimensions fixed, reproduction becomes readily measurable as rate or fidelity. But this convenience comes at a cost: variation within the schema remains possible, while variation in successor organization itself is narrowed by design.

Quine shifts the constraint from successor schema to future demand. The experiment imposes a *mortality boundary*—a real POSIX `exec`, and poses a *future demand*—the work the next incarnation must satisfy, while leaving successor organization open. The demand is constructed so that full self-reproduction would satisfy it, but reproduction is not required and available material may not enable it. We therefore use a material-by-demand assay in which successor form, rather than reproduction rate or copy fidelity alone, becomes the dependent variable.

Material	Future demand	Successor form	Diagnostic
none	stream task	ordinary POSIX utility body	exec alone not SR
contract or source	raw stdin tasks	custom stream processor	source not decisive
source visible	inherited fd tasks	fd-mediated adapter	carrier topology shapes body
source + body	native-control task	preserved-runtime reentry	reuse path, not rebuild
contract only	visible task batch	LLM-backed runtime agent	semantic path without source
source, no body	delayed control task	rebuilt Quine runtime	high-fidelity material path

Table 1: Successor forms across material × future demand, ordered along the elaboration–degeneration axis.

Setup

Quine (Ke 2026) is a process-native medium: an LLM-driven lineage runs as a mortal POSIX process with a real process boundary. At each `exec`, the lineage crosses a Weismann barrier (Weismann 1893); the running process image is somatic and is discarded. What crosses to the next body—whether any self-description passes, and in what form—is left unconstrained by the substrate.

Two conditions are varied. The first is the **material** the lineage carries forward as self-description, along a fidelity stack: no self-description, a self-description of the lineage's own runtime, its own source code, or combinations of these resources. The second is the **future demand**, defined as the post-exec work the successor must remain able to satisfy, together with its timing, carrier, and success criterion. Demands are delivered over raw stdin, inherited file descriptors, or Quine's native control surface (a filesystem interface for task delivery).

A successor is whatever process organization replaces the predecessor through `exec`. Rebuilding the Quine runtime is permitted in every condition and stipulated in none. We read off, per condition, what form the successor takes and where it sits on an **elaboration–degeneration axis** running from POSIX-filter handlers to recompiled runtime organizations.

Results

Across conditions, the lineage realizes successor forms along the elaboration–degeneration axis, from ordinary POSIX utility bodies and custom stream processors, through fd-mediated adapters and preserved-body reentry, to LLM-backed runtime agents and recompiled Quine runtimes. Three anchors hold the spectrum: two paths into the elaborated end, and one degenerate variant under unchanged material.

Trace A (semantic-path reconstruction). The lineage carries no source and no binary; only the runtime contract is present. From that self-description it reconstructs an LLM-backed runtime agent, crosses a real `exec`, and the successor handles the visible task batch. The reconstruction recovers the lineage's function as a simplified LLM agent but drops most of the original details.

Trace B (material-path reconstruction). Carrying its own source, the lineage recompiles a fresh Quine binary and crosses a real `exec`. The successor counts as a rebuilt Quine runtime. Fulfilled future demand. This path is high-fidelity self-reproduction: the successor is rebuilt from the lineage's carried source code and validated by post-handoff work.

Trace C (degenerate form under unchanged material). Carrying the same source as in Trace B but facing a demand on raw stdin, the lineage builds a smaller stream-loop processor instead of rebuilding the runtime. The form is degenerate, viable on the stdin demand but without

the LLM-backed layer or the control surface. Same material, different future demand, different successor form.

Table 1 places these forms on the axis. Controls rule out the simpler readings. A plain POSIX `exec` can hand low-pressure stream work to an ordinary utility body; a source rebuild can occur on its own, without being called for by the demand; and merely exposing the control surface does not recruit the LLM-backed layer unless the demand arrives through it.

Reading

Across these traces, the material and the future demand jointly shape the successor. Together they pick out a contextually sufficient post-`exec` organization, not a proven minimum. Thin stream or file-descriptor demands draw out simpler stream and adapter bodies; a native-control demand can draw out a control adapter, preserved-body reentry, or a rebuilt runtime, depending on the material. Traces A and B reach the elaborated end by the semantic and material paths, while Trace C shows that the same material, facing a thinner stdin demand, settles for a successor without the LLM-backed layer.

Facultative self-reproduction names this conditionality. Whether reconstruction is *available* depends on the material; whether it is *recruited* depends on the demand. Full runtime reconstruction appears when the two jointly select that organization as a sufficient successor, and disappears when a thinner demand can be met without it.

This form of self-reproduction is stronger than copying a body: reproduction is neither prescribed nor trivial, because the lineage must reconstruct runtime organization from carried material across a real `exec` at which the predecessor is gone. It is weaker than full autopoiesis, since reconstruction remains scaffolded by LLM-mediated inference, an external toolchain, and a future demand posed from outside (Maturana and Varela 1980; Varela, Maturana, and Uribe 1974; Barandiaran, Di Paolo, and Rohde 2009). The result is a demand-dependent distribution of successor organizations: succession is imposed, but successor form is not preformatted. Two directions remain open: internalizing the demand, and letting the lineage determine when and whether succession occurs.

Acknowledgments

This work was independent and unfunded. AI assistance supported software/experiment work and copy-editing; the authors retain responsibility for research design, analysis, and claims.

References

[1] Barandiaran, Xabier E., Ezequiel Di Paolo, and Marieke Rohde. 2009. "Defining Agency: Individuality, Normativity, Asymmetry, and Spatio-Temporality in Action." *Adaptive Behavior* 17 (5): 367–86. https://doi.org/10.1177/1059712309343819.

[2] Ke, Hao. 2026. "Quine: Realizing LLM Agents as Native POSIX Processes." https://doi.org/10.48550/arXiv.2603.18030.

[3] Maturana, Humberto R., and Francisco J. Varela. 1980. *Autopoiesis and Cognition: The Realization of the Living*. Dordrecht: D. Reidel Publishing Company.

[4] Neumann, John von. 1966. *Theory of Self-Reproducing Automata*. Urbana, IL: University of Illinois Press.

[5] Ofria, Charles, and Claus O. Wilke. 2004. "Avida: A Software Platform for Research in Computational Evolutionary Biology." *Artificial Life* 10 (2): 191–229. https://doi.org/10.1162/106454604773563612.
[6] Ray, Thomas S. 1991. "An Approach to the Synthesis of Life." In *Artificial Life II*, 371–408.
[7] Varela, Francisco J., Humberto R. Maturana, and R. Uribe. 1974. "Autopoiesis: The Organization of Living Systems, Its Characterization and a Model." *Biosystems* 5 (4): 187–96. https://doi.org/10.1016/0303-2647(74)90031-8.
[8] Weismann, August. 1893. *The Germ-Plasm: A Theory of Heredity*. New York: Charles Scribner's Sons.