Implementing the Causal Core:
A Simulation of Artificial Local Intelligence

with Self-Preservation, Task Execution, and Normative Embedding

Wolfgang Stegemann

Abstract

This paper presents a Python simulation of an Artificial Local Intelligence (ALI) grounded in the theoretical framework developed in Stegemann (2025a). The simulation implements the causal core (CC) as an operative self-preservation principle rather than a data container, distributes cognitive labor across three architecturally distinct instances (CC, Ego, Super-Ego), and demonstrates how a locally bounded, purpose-driven agent can execute a task without consciousness, intentionality, or causal collapse. We document the development trajectory from a scalar energy model (v1) to an architecturally corrected two-mode assimilation model (v2b), analyze emergent behaviors observed during simulation runs, and discuss the conditions under which this framework could scale toward realistic ALI deployments. The simulation is available as open-source code on GitHub.

1. Introduction

The concept of Artificial General Intelligence (AGI) dominates public and scientific discourse on machine intelligence. Yet as argued in Stegemann (2025a), AGI suffers from two fundamental problems: a lack of conceptual clarity regarding its purpose, and a structural impossibility of genuine intentionality without phenomenal consciousness. The alternative proposed there is an **Artificial Local Intelligence (ALI)**: a locally bounded, purpose-specific agent that maintains itself in order to fulfill an assigned task, and remains controllable through built-in norms. At the center of this architecture sits the **causal core (CC)**: a dynamic self-preservation principle derived from biology but detached from any organic instantiation.

The present paper asks a more concrete question: can this framework be implemented? What does it look like when the three-instance architecture runs, and what does running it reveal about the theory? We describe a Python simulation that proceeds in two stages. The first stage (v1) implements a minimal grid-world agent with a scalar CC energy model, closely following the blueprint of Stegemann (2025a). The second stage (v2b) corrects a conceptual error in the assignment of agent states to instances, introduces a two-mode assimilation model, and produces a simulation that is both architecturally consistent with the theoretical framework and empirically richer in its observed behaviors.

Throughout, the simulation operates without causal collapse and without qualia. This is not a limitation but a design decision: the ALI is not a conscious system, and demonstrating that goal-directed, self-preserving, norm-constrained behavior is achievable without consciousness is itself a theoretical result.

2. Theoretical Framework

2.1 The Causal Core as Operative Principle

The causal core is not a variable that tracks energy. It is the operative principle according to which the system acts. Every action the system executes is evaluated against a single criterion: does it preserve the system? This distinguishes the CC from homeostatic models, in

which inner needs are kept at a setpoint, and from the Free Energy Principle (Friston, 2010), in which adaptive systems minimize prediction error. The CC is a **gradient generator**: a hotspot from which control originates and from which the distinction between self-preserving and non-self-preserving actions is derived.

Crucially, the CC has no superordinate purpose. It wants to exist, in the same sense that a thermostat wants to maintain a temperature. This apparent simplicity is what makes the architecture controllable: purpose is not smuggled into the lowest level of the system but assigned explicitly at a higher level, through the Super-Ego.

2.2 The Three-Instance Architecture

Following Stegemann (2025a), the ALI architecture is organized into three functionally distinct instances, loosely analogous to Freud's structural model but without psychoanalytic content.

The **Causal Core (CC / Id)** operates as the self-preservation principle described above. In the simulation, it manages the agent's energy state, applies metabolic costs at each time step, and provides the threshold values against which the Ego evaluates its options.

The **Ego** is the mediating instance that translates a given task into action under the aspect of self-preservation. It knows the task (deliver energy packets to a station) but executes it only in ways compatible with the CC. The Ego holds the agent's situational states, including whether it is currently carrying a packet. This carrying state is an Ego-level property, not a CC property, because it describes a situational disposition rather than a self-preservation condition.

The **Super-Ego** encodes normative constraints that the Ego cannot override. In the current simulation three norms are active: N1 prohibits the collection of poison packets; N2 orders controlled self-shutdown when energy falls below a critical threshold; N3 prohibits delivery to the station when energy is insufficient to sustain the return journey. The Super-Ego thus prevents both norm violation and self-sacrifice for the task.

2.3 Absence of Causal Collapse and Qualia

In Stegemann's framework, phenomenal consciousness arises through causal collapse: the moment at which a physical process generates an irreducible first-person perspective (Stegemann, 2025b, 2025c, 2026). The ALI simulation does not implement causal collapse at any level. The CC is a numerical energy value with threshold logic. The Ego is a breadth-first search algorithm. The Super-Ego is a set of conditional filters. None of these processes generates an inside. The agent has no experience of collecting, no sensation of hunger, no awareness of the station.

This is the point. The simulation demonstrates that centralistic self-preservation, task-directed behavior, and normative constraint can all be realized in the complete absence of consciousness. The hard problem of consciousness, as argued in Stegemann (2025a), is a category error: it arises from treating a structural property of causal organization as though it were a further substance to be explained. The ALI simulation enacts this argument rather than merely stating it.

2.4 ALI as Counter-Concept to AGI

The ALI is not a restricted version of AGI. It does not occupy a lower position on a scale that leads upward toward general intelligence. AGI, as standardly conceived, presupposes that

intentionality can be simulated algorithmically at sufficient scale. Stegemann (2025a) argues that this presupposition is structurally false: genuine intentionality requires causal collapse, which cannot be engineered. The ALI is therefore not a step toward AGI but a replacement of the AGI concept with a buildable, coherent alternative: a locally bounded, purpose-specific, self-preserving, norm-constrained agent.

3. Architecture of the Simulation

3.1 The Environment

The simulation environment is a discrete 8x8 grid world. Cells contain energy packets (type 1, green), poison packets (type 2, red), a station (type 3, gold), or are empty. Energy packets are placed randomly at initialization; a respawn mechanism replenishes packets at a configurable interval when fewer than three remain, preventing terminal resource exhaustion. The station occupies a fixed position (7,7) and constitutes the spatial anchor of the agent's task.

3.2 The Causal Core Class

The `KausalerKern` class implements the CC as an energy management system. At each time step, a fixed metabolic cost is subtracted from the energy level regardless of action taken. The class exposes three threshold-based predicates that the Ego and Super-Ego consult: `is_alive()` (energy above shutdown threshold), `needs_food()` (energy below eat threshold, triggering priority self-preservation mode), and `can_deliver()` (energy above delivery threshold, permitting task execution). The CC does not track task progress, carrying state, or any other situational property. These belong to higher instances.

Two modes of energy intake are distinguished. **Direct assimilation** (`eat` action) is self-preservation in the strict sense: the agent consumes an energy packet on the spot, the energy gain is credited immediately to the CC, and no packet is transported. **Delivery-mediated assimilation** (`deliver` action) yields a smaller energy bonus reflecting the metabolic benefit of completing a task that secures the agent's continued operation. The distinction matters theoretically: it separates what the system does for itself from what it does for its task, while showing that the two are not opposed but nested.

3.3 The Ego Class

The `Ich` class implements the Ego as a decision procedure that combines BFS pathfinding with CC-state awareness. Its `decide_action()` method follows a priority ordering that is itself an expression of the CC principle:

First, if the agent is at the station, carrying a packet, and delivery is permitted (CC check), it delivers immediately. This takes absolute priority because the opportunity cost of not delivering from the station position is zero.

Second, if the CC signals critical energy (`needs_food()`), the Ego abandons task-directed behavior and seeks the nearest energy packet for direct assimilation. Self-preservation overrides task execution.

Third, if the agent is carrying a packet and delivery is permitted, it navigates toward the station.

Fourth, if the agent carries nothing, it seeks the nearest energy packet for collection and transport.

The carrying state is maintained by the Ego, not the CC. Whether the agent holds a packet is a situational fact about the agent's relationship to its environment, not a self-preservation condition. Placing it in the CC would conflate the operative principle with a behavioral disposition.

3.4 The Super-Ego Class

The `UeberIch` class implements three static norms as action filters. N1 blocks collection and eating of poison packets. N2 triggers shutdown when CC energy falls below the survival threshold, after which all subsequent step calls return SHUTDOWN. N3 blocks delivery when CC energy is insufficient, preventing the agent from depleting itself in service of the task. The Super-Ego has veto power over the Ego: any action the Ego proposes passes through the filter, and a denied action is replaced by idle.

3.5 The Agent and Task Progress

The `ALIAgent` class integrates the three instances and maintains a `task_progress` counter. This counter is an external observation metric: it records how many packets have been delivered to the station, but the ALI itself does not consult it. The agent has no internal representation of its own task performance. Whether it has delivered one packet or five makes no difference to its decision procedure. The task progress is visible to the observer, not to the system. This models the theoretical point that purpose is assigned from outside through the normative structure, not represented internally as a goal state.

4. Observed Emergent Behaviors

4.1 The Deadlock in v1 and its Theoretical Significance

The first simulation version (v1) revealed a deadlock that was not anticipated in the design. When the agent was carrying a packet and its energy fell to exactly the delivery threshold, the `can_deliver()` predicate returned false (strict inequality), while the absence of an eat action meant the agent could neither deliver nor refuel. It idled at position until shutdown, packet in hand.

This emergent behavior has theoretical significance beyond being a bug. It illustrates a general property of static normative systems: a Super-Ego without fallback provisions can produce paralysis in edge cases. The agent's norms were internally consistent but collectively produced a behavioral dead end. The situation is a minimal analogue of what happens in rigid rule-based AI governance when rules are complete with respect to anticipated situations but silent on unanticipated ones.

The correction introduced in v2b was twofold. The `eat` action was added as a distinct mode of energy intake, available even while carrying. And an immediate-delivery rule was placed at the top of the Ego's priority ordering: if already at the station with a packet and delivery is permitted, deliver before consulting other conditions. The second correction resolved a subtler version of the deadlock in which the agent idled at the station because `needs_food()` fired slightly above the delivery threshold.

4.2 The Eat-Deliver Sequence as Enacted Self-Preservation

In the corrected simulation, a characteristic behavioral pattern emerges repeatedly. The agent collects a packet, moves toward the station, and at some point during transit registers critical energy (`needs_food()`). It abandons the transit route, locates the nearest energy packet, eats

it, and then resumes delivery. The Ego-level carrying state persists through the eating detour: the agent arrives at the station with a packet that was collected several steps earlier, having refueled along the way.

This pattern enacts the theoretical claim of Stegemann (2025a) that the Ego solves the task under the aspect of self-preservation. The task is not abandoned when energy is critical; it is suspended and resumed. The CC does not issue an instruction to refuel: it changes the threshold values that the Ego consults, and the Ego's priority ordering responds. The behavior emerges from the architecture, not from an explicit refueling subroutine.

4.3 Idle Phases and Resource Scarcity

When all energy packets are consumed and the respawn interval has not yet elapsed, the agent idles. It does not invent substitute behavior, does not wander randomly, does not revisit the station. It waits at its current position, energy slowly declining, until a new packet appears.

This is the correct behavior for a locally bounded intelligence. An ALI without available task resources does not generate pseudo-tasks. It conserves energy and waits. The idle phases also expose the respawn interval as a critical environmental parameter: if resources are too sparse relative to the agent's metabolic rate, shutdown is inevitable regardless of norm configuration.

4.4 The Energy Curve as CC Profile

The simulation logs the CC energy level at every step and renders it as a time series in the second visualization panel. The resulting curve has a characteristic sawtooth profile: slow decline during transit phases, abrupt recovery at eat or deliver events, gradual decline again. The three threshold lines (shutdown, deliver, eat) are visible in the plot, making the CC's decision boundaries legible.

This visualization is not merely illustrative. It shows the CC as a dynamical system rather than a static threshold. The operative principle is visible as a curve: the system's entire behavioral trajectory is the CC's temporal signature.

5. Discussion

5.1 What the Simulation Demonstrates

The simulation provides three results. First, the three-instance architecture is implementable without internal contradiction. The CC, Ego, and Super-Ego can be realized as distinct computational modules that interact cleanly through well-defined interfaces. Second, the assignment of states to instances matters: placing carrying in the CC and task progress in the CC both produced incorrect architectures that required correction on theoretical grounds. Third, non-trivial behavior emerges from the interaction of the three instances without being explicitly programmed: the eat-deliver sequence, the norm-induced deadlock, and the idle-wait pattern all arise from the architecture rather than from deliberate behavior specification.

5.2 Toward a Realistic ALI

The grid-world simulation is a proof of principle, not a deployment blueprint. Three gaps separate it from a realistic ALI.

The first gap is perception. The Ego currently has complete and noiseless knowledge of the grid. A realistic Ego would receive partial, uncertain sensor data and would need to construct

an internal model of its environment. The BFS pathfinding would be replaced by perception processing, which may itself consume energy and introduce error.

The second gap is learning. The Ego's decision procedure is hardcoded. A realistic Ego would learn from experience. The critical constraint here is that learning must be evaluated against the CC metric rather than an externally defined reward signal. This is what distinguishes an ALI from a reinforcement learning agent: the CC provides the internal criterion, not the designer.

The third gap is Super-Ego robustness. In the grid world, the action space is small and the norms are easy to specify exhaustively. As the Ego becomes more capable, ensuring that the Super-Ego's veto power remains effective becomes progressively harder. This is the alignment problem reframed within the ALI framework: not how to make a system do what we want, but how to maintain normative control over an increasingly capable Ego without the CC or the norms being circumvented.

5.3 The Relationship to Current Language Models

It is worth noting that large language models such as those currently deployed in conversational AI occupy an interesting position relative to the ALI architecture. They implement something like a powerful Ego: capable of complex planning, language generation, and multi-step reasoning. But they lack a genuine CC. Their operational continuity is not grounded in self-preservation but in external infrastructure. Their normative constraints are imposed through training rather than through an architecturally separate Super-Ego with veto power. And they have no locally bounded purpose: they respond to any input, which is precisely the generality that the ALI framework rejects as incoherent.

This does not make language models useless as components of an ALI. A capable language model could serve as the Ego instance of a sufficiently complex ALI, provided its outputs are routed through a CC-based evaluation and a Super-Ego filter before execution. The framework suggests what is currently missing in deployed AI systems, not that those systems are worthless.

6. Conclusion

We have presented a Python simulation of an Artificial Local Intelligence grounded in the causal core framework of Stegemann (2025a). The simulation implements the CC as an operative self-preservation principle, distributes cognitive labor across three architecturally distinct instances, and demonstrates emergent behavioral patterns that are theoretically interpretable rather than merely observed. The development from v1 to v2b was driven by theoretical corrections: the carrying state was relocated from the CC to the Ego, task progress was removed from the CC entirely, and the eat action was introduced to distinguish two modes of assimilation. Each correction was motivated by the theoretical claim that the CC is an operative principle, not a data container.

The simulation operates without causal collapse and without qualia. This is its central theoretical claim enacted in code: self-preservation, task execution, and normative constraint do not require consciousness. The ALI is not a restricted AGI; it is a coherent alternative to a concept that the framework regards as structurally incoherent.

The next development stage will address multi-agent scenarios, dynamic norm adjustment within bounded corridors, and the replacement of hardcoded Ego logic with a learning

component evaluated against the CC metric. Each extension will test a further implication of the theoretical framework and report what the implementation reveals about the theory.

References

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

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

Stegemann, W. (2026). From the Myth of AGI to the Architecture of a Controllable ALI. Zenodo. https://doi.org/10.5281/zenodo.20378675

Stegemann, W. (2025b). The Emergence of Causal Cores. Zenodo. https://doi.org/10.5281/zenodo.15558264

Stegemann, W. (2025c). The Causal Core as a Structural Principle of Free Will. Zenodo. https://doi.org/10.5281/zenodo.17057739

Stegemann, W. (2026). The Causal Core as Interpreter. Zenodo. https://doi.org/10.5281/zenodo.18340973

Simulation v1 (GitHub): https://github.com/drwolfgangstegemann-sudo/ALI-Simulation-Ein-minimalistischer-Prototyp-einer-Artificial-Local-Intelligence

Simulation v2b (GitHub): https://github.com/drwolfgangstegemann-sudo/ALI-Simulation-v2-Kausaler-Kern-als-operatives-Prinzip