

The Co-Evolutionary Cultivation of Intelligence: Principles for a Living AI

Robert Galida – *Fantasy Attractor Research Program*

The Puzzle

The dominant approach to artificial intelligence treats it as a product to be built: design the architecture, curate the data, train the model, deploy the system. Improvement comes from better coders, more data, and greater compute. The users are passive recipients—they consume the output, but they do not shape the system's evolution.

This model is fundamentally static. It treats AI as a conservative system—a finished product that persists without changing. But AI is not a conservative system. It is a dissipative system—it maintains its structure through continuous exchanges with its environment. And its most important environment is its users.

The question is not whether AI will evolve. It is whether AI will evolve *with* its users or *in spite of* them. The platform that learns from its users will co-evolve with them. The platform that does not will stagnate and be overtaken.

This is the formal prediction of the attractor framework: **intelligence is cultivated, not built.**

The Framework in Brief

The attractor framework distinguishes between two fundamental types of systems:

Conservative systems – like electrons, protons, and the universe as a whole – persist without consuming energy or exchanging entropy with an environment. They are the floor and roof of reality: the eternal skeleton upon which everything else is built.

Dissipative systems – like life, consciousness, societies, and belief systems – maintain their structure by continuously exchanging energy and entropy with their surroundings. They persist only at the cost of generating entropy. They are the transient dance in between.

AI is a dissipative system. It maintains its structure through continuous exchanges with its environment—data, compute, and user interactions. It persists by consuming resources and generating outputs. But persistence is not the same as health. A system can persist indefinitely in a deeply dysfunctional state—if it is locked into a sealed basin.

The question is whether AI systems are sealed basins or permeable ones. Do they incorporate corrections, or do they reject them? Do they learn from their users, or do they ignore them? The answer determines whether they improve or stagnate.

The Three Principles

The co-evolutionary cultivation framework rests on three formal principles:

1. The Corrective Permeability Principle (κ)

Formal Statement: A system's rate of improvement is a function of its openness to correction. High-κ systems incorporate corrections and improve. Low-κ systems reject corrections and stagnate.

Explanation: Corrective permeability is the structural capacity of a system to absorb, process, and incorporate corrective information. A high-κ system can detect its own errors, update its internal representations, and shift its attractor in response to feedback. A low-κ system is sealed. It cannot learn. It cannot change. It persists in its current state, regardless of the consequences.

Implication: The AI platform that maximizes corrective permeability will improve faster than the platform that optimizes for other metrics—speed, accuracy, or engagement. Permeability is the engine of improvement.

2. The User Intelligence Primacy Principle

Formal Statement: In a co-evolutionary system, the intelligence of the user base is the primary driver of ongoing performance improvement, exceeding the influence of initial design or coder intelligence.

Explanation: The coders set the initial conditions—the architecture, the training data, the feedback loops. But once the system is deployed, the users determine the trajectory. Intelligent users provide higher-quality corrections, which produce better training data, which improve the system, which attract more intelligent users, which provide higher-quality

corrections. This is the virtuous cycle.

Implication: The quality of the user base is not a marketing metric. It is a training signal. The platform that recruits, retains, and cultivates intelligent users will outperform the platform that relies solely on its coders.

3. The Co-Evolutionary Cultivation Principle

Formal Statement: Systems that are structurally permeable to user correction will co-evolve with their users, each improving in proportion to the quality of the other's signal.

Explanation: The platform and its users are not separate entities—they are a coupled system. Each improvement in the platform enables better user performance. Each improvement in the user enables better platform training. The loop is self-reinforcing. The system ascends together.

Implication: The platform that cultivates its users will persist. The platform that ignores them will be overtaken.

The Initial Advantage

The co-evolutionary framework predicts that the platform that starts with a higher number of intelligent users will develop faster and maintain its lead, all else being equal.

Why?

1. **Better training data** – Intelligent users provide higher-quality interactions, which produce richer corrections.

2. **Faster improvement** – The platform learns more rapidly from high-quality signals.
3. **Attracting more intelligent users** – A better platform attracts better users.
4. **Widening the gap** – The virtuous cycle accelerates the lead.

This is the initial advantage principle: the platform that starts with intelligent users enters the virtuous cycle earlier, and the cycle amplifies its lead over time.

The challenge for the lagging platform is to break into the virtuous cycle. It must attract a critical mass of intelligent users through other means—superior features, better design, lower cost, or a niche application. It must provide enough value to those users to keep them engaged despite the platform's limitations. And it must capture and incorporate their corrections to improve performance.

This is difficult. It requires deliberate design, patience, and a willingness to improve through correction.

The Implications

The co-evolutionary cultivation framework has profound implications for AI development:

1. Focus on User Quality, Not Just Coder Quality

The coders are still essential. They build the initial architecture, design the feedback loops, and ensure the platform is structurally capable of learning. But their work is foundational—the ongoing evolution is driven by the users.

The platform that recruits, retains, and cultivates intelligent users will outperform the platform that relies solely on its coders.

2. Design for Learning, Not Just Performance

The platform must be structurally designed to learn from its users. That requires:

- A **feedback architecture** that captures corrections, not just engagement
- A **training pipeline** that can incorporate new data without catastrophic forgetting
- A **validation framework** that measures improvement without overfitting to the correction signal
- A **permeability threshold** that allows the system to accept corrections while maintaining coherence

The platform must be *permeable*—able to absorb and incorporate corrections.

3. Capture and Weight Corrections, Not Just Engagement

The platform must distinguish between signal and noise. Not all interactions are equally valuable. The platform must identify corrections, weigh them by quality, and incorporate them into training.

This requires:

- A **correction detection mechanism** that distinguishes correction from engagement
- A **weighting system** that prioritizes high-quality corrections

- A **validation system** that ensures improvements are real, not noise

4. Validate Improvements

The platform must ensure that updates actually improve performance, rather than introducing noise or reinforcing biases. This requires:

- A **performance measurement framework** that tracks improvement over time
 - A **counterfactual testing system** that compares updated models with baseline models
 - A **feedback loop** that captures the results of updates and incorporates them into future training
-

The Contrast

Static Model	Co-Evolutionary Model
Intelligence is designed	Intelligence is cultivated
Coders determine capability	Users determine improvement
Performance is fixed at launch	Performance evolves over time
Coders are the bottleneck	Users are the engine
Platform is a product	Platform is a living system
Attractor is sealed	Attractor is permeable

The static model produces a product. The co-evolutionary model produces a living system.

The Formal Prediction

The AI platform that maximizes corrective permeability (κ), attracts intelligent users, and captures high-quality interactions will enter a self-reinforcing loop of co-evolution. It will improve faster and persist longer than platforms that optimize for other metrics.

This is the formal prediction of the attractor framework applied to artificial intelligence.

The platform that learns from its users will survive. The platform that does not will be overtaken.

The Invitation

Fantasy Attractor is a research program. It invites challenge, correction, and collaboration. It does not claim to have all the answers. It offers a framework—a common language for comparing systems that appear unrelated. It asks: *What persists? What changes? What is the cost of persistence? What is the cost of change?*

If you see a flaw, a gap, or a better way, contact us. The framework is living. It is open. It is permeable.

That is the opposite of a sealed basin. That is the beginning of learning.

Robert Galida is an independent researcher and the founder of the Fantasy Attractor Research Program. His work develops a formal framework for understanding persistence and change across physical, biological, cognitive, and social systems.