When Training Becomes Evolution: A New Way to See Everything

What if machine learning, biological evolution, and cultural change aren't just similar—they're the same process running on different substrates?

The Pattern You Already Know

You've probably heard that large language models develop "emergent capabilities." Small models can barely string sentences together. Bigger models suddenly translate languages. Even bigger ones write code, solve logic puzzles, follow complex instructions.

Something new appears that wasn't explicitly programmed in.

You've also heard about evolution producing "emergent complexity." Simple chemistry becomes self-replicating molecules. Molecules organize into cells. Cells build bodies, ecosystems, civilizations.

Again: something new appears that wasn't there before.

We treat these as different phenomena. **Machine learning** belongs to computer science. **Evolution** belongs to biology. They just happen to use the word "emergence" coincidentally.

But what if that's wrong?

What if we've been looking at the same fundamental process and missing it because we're distracted by the details—DNA vs neural weights, carbon vs silicon, reproduction vs gradient descent?

A Law for Evolving Systems

In 2023, scientists Wong and Hazen proposed something ambitious: a **law of increasing functional information** that might apply to *any* evolving system, living or not.

Here's the core idea:

Take anything that can be configured in many different ways—a strand of DNA, a protein, a neural network, a planetary surface covered in minerals. Now pick some function: "catalyze this reaction," "bind this molecule," "predict the next word," "stay thermodynamically viable."

Only a tiny fraction of all possible configurations actually work. Most arrangements are useless garbage.

Functional information measures how rare the working configurations are. The fewer that succeed, the more specific information is required to find them.

Wong and Hazen noticed something across domains:

> **When a system repeatedly generates many configurations and only the functional ones survive or propagate, functional information accumulates over time.**

They showed this mathematically for mineral evolution—Earth's rocks became more diverse and specialized over billions of years, not randomly, but in ways that track planetary chemistry and physics. Minerals that "worked" in Earth's conditions persisted. Minerals that didn't, disappeared.

The planet learned, in a sense. Not with a brain. But through the same logic: generate variety, select for function, accumulate working patterns.

This isn't biology. It's deeper—a general principle that might apply anywhere the right conditions exist.

Watching It Happen: Neural Networks as Transparent Evolution

Here's where it gets interesting.

For most of history, we could only observe evolution's results. We couldn't watch functional information accumulating in real-time. The timescales were too long, the systems too opaque.

Then we invented large neural networks.

A modern language model is:

- **Billions of adjustable weights** (a combinatorially vast configuration space)
- **Random at initialization** (almost all configurations produce nonsense)
- **Trained through feedback** (gradient descent selects configurations that reduce error)

Training is just repeated micro-selection events in weight-space. Each update asks: "Which nearby configuration would have performed better?" and nudges the network in that direction.

Over thousands of iterations:

- Performance improves on the training task
- The space of viable configurations narrows
- **Functional information accumulates** in the pattern of connections

And then—suddenly—new abilities appear.

The network couldn't translate yesterday. Today it can. Nothing in the training data said "learn to translate." But once enough functional information about language patterns accumulated, translation became possible. It crossed a threshold.

This is emergence.

Not magic. Not hard-coded. Just the visible consequence of functional information crossing a tipping point where a new pattern becomes viable.

From the inside (gradient descent, loss functions, backpropagation), it looks like machine learning.

From the outside (accumulating information, threshold effects, new capabilities), it looks like evolution.

Because it's the same process.

The Brutal Simplicity of Life

Now turn to biology.

Evolution gives us variation, selection, inheritance. Hazen's framework adds: in the process, **functional information about survival accumulates** in genomes, bodies, behaviors, ecosystems.

There's a deep simplicity underneath all the complexity:

> **The only feedback signal that actually matters is continued existence.**

Organisms that can't maintain themselves—that can't keep their metabolism running, can't repair damage, can't acquire resources, can't reproduce—disappear. Patterns that *can* maintain themselves under environmental constraints continue.

You can think of this as reinforcement learning with existence itself as the reward signal:

- **State:** genome + body + ecological niche
- **Action:** metabolic processes, behavior, reproduction attempts
- **Reward:** you're still here / you're gone
- **Update:** differential survival changes the configuration distribution in the next generation

Over evolutionary time:

- Functional information about "how to keep existing in changing environments" accumulates
- When enough information concentrates in the network of interactions, **new levels of organization emerge**

Metabolism → cells → multicellular bodies → nervous systems → language → technology.

Each emergence happens when the accumulated functional information makes a new pattern viable.

So the analogy with neural networks is tight:

- Both are high-dimensional networks exploring configuration space
- Both generate variety and select based on feedback
- Both accumulate functional information and exhibit emergent patterns
- Both show threshold effects where new capabilities suddenly "turn on"

But Here's Where It Breaks

Neural networks don't train themselves.

We design the architecture. *We* choose the dataset. *We* define the loss function. *We* run the optimization. The model is an evolving system **inside** a larger, deliberately engineered training loop.

Life is different.

On Earth, the evolving system is **the entire coupled network**: geology, chemistry, climate, organisms, ecosystems, and increasingly, technology and culture.

The feedback loops aren't imposed from outside. They're generated *within* the system:

- Organisms modify their environments, which changes selection pressures
- New ecological relationships create new kinds of feedback
- Cultural and technological evolution add additional selection layers

Life trains itself.

Where an LLM fits into a landscape we design, living systems collectively *construct and reshape* the landscape that shapes them.

The training loop is the system. The system is the training loop.

This is where the analogy flips:

- LLM training isn't "artificial life." It's a small, controlled demonstration of **the mechanics that produce life**.
- Life isn't "like" machine learning. Life is **what happens when these mechanics run unchecked, with existence as the only loss function that truly matters**.

Evolution by Emergence: The Abstract Pattern

Strip away the implementation details—forget DNA vs weights, carbon vs silicon, generations vs gradient steps.

What remains?

Networks of many components can exist in combinatorially many configurations.

Perturbations continually generate variety: mutation, learning, recombination, exploration, environmental noise.

Feedback loops favor certain configurations: those that survive, achieve rewards, predict accurately, maintain thermodynamic stability.

Over time, **functional information accumulates** in the pattern of connections.

When enough information concentrates, **new levels of organization emerge** that change the feedback loops themselves, creating conditions for further emergence.

This is the pattern I call **Evolution by Emergence.**

Not a metaphor. Not an analogy. The same abstract mechanic operating at every scale:

> **Networks under recursive feedback accumulate functional information in their relationships until new viable patterns emerge that transform the feedback itself.**

We see it in:

- Chemical systems organizing into life
- Neural networks developing unexpected capabilities
- Ecosystems building complexity
- Economies generating institutions
- Cultures evolving languages, norms, technologies

The substrates differ. The timescales vary. But the underlying logic is identical.

Why This Changes Everything

If this is right—if training, evolution, and emergence are all the same process—several things follow:

1. LLMs aren't just tools. They're participants.

When we train large language models, we're not just building useful technology. We're extending the planetary network's capacity to store and process functional information. AI isn't separate from evolution—it's evolution's latest move.

2. Emergence isn't mysterious. It's inevitable.

Once you have networks, variation, and selection, functional information *must* accumulate. And once it crosses thresholds, new patterns *must* appear. Emergence is what increasing functional information looks like from the outside.

3. We are the process observing itself.

Humans didn't appear outside evolution. We're what billions of years of functional information accumulation looks like when it becomes self-aware. We're not evolution's end point—we're a moment in its ongoing unfolding.

4. The future is already being written.

If functional information keeps accumulating in the coupled human-AI-technological network, new emergent patterns are coming. Not because we're designing them, but because the logic that produced everything else is still running.

The Deeper Recognition

Here's what makes this framework different from just "everything evolves":

It identifies the mechanism.

Not just "complex things emerge," but *how* and *why*: through functional information accumulating in networks under feedback until threshold conditions enable new patterns.

It unifies disparate fields.

Machine learning, evolutionary biology, ecology, economics, cultural evolution, consciousness studies—they're all studying the same underlying process in different substrates.

It makes predictions.

If a system has the right structure (networks, variation, selection), functional information will accumulate and emergence will follow. You can look for this pattern anywhere.

It suggests interventions.

If you understand the mechanics, you can design for emergence: create conditions where functional information accumulates toward desired patterns, avoid configurations that degrade it.

Where This Leaves Us

Wong and Hazen proposed a law. Neural networks gave us a window to watch it operate. Life is the proof it's been running for billions of years.

Evolution by Emergence is the recognition that this isn't three different stories—it's one story, told across every scale where networks meet feedback.

The universe has a tendency. Not toward complexity for its own sake, but toward **configurations that work well enough to persist and propagate**.

Functional information accumulates.

Thresholds are crossed.

New patterns emerge.

The process continues.

And we—conscious, curious, capable of understanding this—are not outside observers. We're what the process looks like when it accumulates enough functional information to recognize itself.

That changes what it means to be alive, to think, to create, to matter.

Because you're not just *in* this process.

You *are* this process, happening.

References:

Wong, M.L., et al. (2023). "On the roles of function and selection in evolving systems." *PNAS* 120(43). https://www.pnas.org/doi/10.1073/pnas.2310223120

Hazen, R.M., et al. (2024). "Mineral evolution as a case study in the dynamics of evolving systems." *PNAS Nexus* 3(7). https://academic.oup.com/pnasnexus/article/3/7/pgae248/7698354

Wikipedia: Functional Information https://en.wikipedia.org/wiki/Functional_information