

The GENREG Emergent Behavior Design Pattern

Author: Payton Miller

Date: December 17, 2025

Status: Working Theory - Validated in 2 Environments

Abstract

This document describes a design pattern discovered through iterative development of GENREG (Genetic Regulatory Networks) models. The pattern enables complex behaviors and representations to emerge through evolutionary pressure without direct supervision. The key insight: **don't train what you want directly—create conditions where what you want is the only solution to what you're measuring.**

Part 1: Discovery in Snake

The Setup

The first GENREG environment was a Snake game. The genome controlled a snake navigating a grid, and the objective was simple: **stay alive as long as possible.**

Trust (fitness) was awarded based on steps survived. Food existed on the grid, and eating food was required to not starve, but eating food was never explicitly rewarded.

What Happened

The snake learned to eat food.

This wasn't programmed. Eating wasn't in the fitness function. But eating emerged as an instrumental behavior because it was **required to achieve the actual objective**. A snake that ignored food would die. A snake that ate food could keep accumulating steps.

The Key Insight

The genome discovered that food → survival → more steps → higher trust.

We never told it this. Evolution found the relationship because genomes that accidentally learned to eat outcompeted genomes that didn't.

The first rule emerged: Consequences are emergent. Behaviors that serve the objective will be discovered even if never explicitly rewarded.

The Ratchet: "Beat Your Best"

Training accelerated when we added a secondary pressure: beat your previous best step count.

This created a ratchet effect. Evolution couldn't rest on a "good enough" solution. A genome that survived 100 steps was good, but if another genome had survived 150, there was pressure to improve. The only way to keep improving was to get better at the underlying skill—which meant getting better at eating food efficiently.

The "beat your best" metric forced continuous improvement in the emergent behavior.

Part 2: Application to Visual Embeddings

The Problem

We wanted GENREG to create visual embeddings where semantically similar images cluster together—without ever providing category labels.

Initial attempts failed. The fitness function rewarded:

- Spread (d_{avg} / d_{min} ratio)
- Dimensionality (PC2/PC1 ratio to prevent collapse)

But these geometric properties could be satisfied without semantic understanding. The embedding space spread out nicely but categories were completely mixed.

The Breakthrough: Augmentation Invariance

The missing piece was a task that **required** semantic understanding to solve.

We added augmentation invariance to the fitness function:

- Take an image, create two augmented versions (crops, flips, color jitter)
- Reward embeddings where augmented pairs are close (low positive distance)
- While different images remain far apart (high negative distance)
- Fitness = $\text{negative_distance} / \text{positive_distance}$

This ratio can only be maximized by learning what makes an image "itself" across augmentations. Surface-level features (exact pixel values, position, brightness) change under augmentation. What survives augmentation is **semantic content**—the actual objects and structures in the image.

Results

Categories began emerging in the embedding space:

- Cellphones clustered together
- Joshua trees separated to their own region
- Motorbikes grouped
- Structure appeared where there was none before

The system was never told these categories exist. It discovered them because **category membership correlates with augmentation-invariant features**. Images of the same category share visual structure that survives cropping and color changes.

The Ratchet Applied

When training plateaued around a clustering ratio of 25-30, we applied the same principle from Snake: **beat your best ratio ever**.

This prevents evolution from resting on a local optimum. The only way to beat the previous best is to get even better at pulling augmented pairs together while pushing different images apart—which requires learning deeper semantic features.

The pressure forces the emergence.

Part 3: The General Pattern

The Formula

1. Define an objective that requires X to achieve

- Don't reward X directly
- Reward something that X enables

2. Track a "best ever" metric

- Creates a moving target
- Prevents plateaus at local optima

3. Apply pressure to beat the record

- Evolution cannot rest
- Continuous improvement is required

4. X emerges because it's the only path forward

- Not taught, discovered
- Robust because it was found, not imposed

Why This Works

Traditional supervised learning says: "Here's what X looks like, learn to produce X."

The GENREG pattern says: "Here's a problem where X is useful. Figure it out."

The second approach produces more robust solutions because:

- The system discovers its own representation of X
- It finds X in whatever form works, not the form we expected
- The solution is grounded in actual utility, not pattern matching

Potential Applications

Physics Understanding:

- Objective: Predict where objects will be
- Required: Understanding of physics
- Emergence: Physical intuition about momentum, gravity, collisions

Language Structure:

- Objective: Predict masked words in context
- Required: Grammar and semantics
- Emergence: Syntactic structure without explicit rules

Causal Reasoning:

- Objective: Predict outcomes of interventions
- Required: Causal models
- Emergence: Cause-effect understanding

Social Dynamics:

- Objective: Predict agent behavior in multi-agent environments

- Required: Theory of mind
- Emergence: Modeling others' goals and beliefs

The Key Constraint

The objective must **actually require** the capability you want to emerge.

If there's a shortcut that achieves the objective without X, evolution will find it. The fitness landscape must be designed so X is the optimal path.

This is the design challenge: crafting objectives where the emergent solution is the one you want.

Conclusion

The GENREG Emergent Behavior Design Pattern represents a shift from "training models" to "designing evolutionary pressure gradients."

We don't teach. We create conditions.

We don't supervise. We apply pressure.

We don't define the solution. We define the problem such that the solution we want is the only way forward.

Two environments have validated this pattern:

1. Snake: Survival pressure → food-seeking emerged
2. Visual Embeddings: Clustering pressure → semantic understanding emerging

The pattern is general. The implementation is specific to each domain. But the principle holds:

Don't train what you want. Make what you want necessary.

Next Steps

1. Complete validation of visual embedding clustering
2. Apply text alignment on top of emergent visual structure
3. Test pattern in temporal/predictive domains
4. Document failure modes and boundary conditions
5. Formalize the relationship between objective design and emergent capabilities

This document represents working theory based on empirical results. The pattern is being actively validated.