The 12g-ACE & 12g-AFL Adaptive Integrity Framework

A compact formal overview of the mathematical core behind PoCI

Purpose of the Math

The mathematical layer of PoCI exists to do one thing:

Continuously adapt the system's trust mechanisms based on evidence.

Where traditional systems assume trust,
the 12g-ACE / 12g-AFL framework *learns* who stays honest.

To enable that, we introduced two original equations:

- **12g-ACE** — the *Adaptive Coherence Equation*
- **12g-AFL** — the *Adaptive Feedback Lineage equation*

Together, these form the **adaptive controller** operating inside Loop 4.

1. 12g-ACE - Adaptive Coherence Equation

(The main gradient-driven update rule)

The goal of 12g-ACE is to adjust the system's **internal parameters** for event-generation, proof strictness, and expected behavior.

Flow

s$t \rightarrow L_t \rightarrow \nabla L_t \rightarrow \theta${t+1}

Definitions

- **s_t (consistency score):**
  The verifier's judgment of whether the lineage and proofs match expected behavior.
- **L_t (loss):**
  A function measuring deviation from expected honesty.
  Example: L_t = φ(1 - s_t)
  where φ is any shaping function (identity, hinge, log, etc.)
  - **∇L_t:**
    Gradient of the loss with respect to model/system parameters.

- θ_{t+1}:
  The updated internal parameter vector that influences Loop 1's rules (what counts as an event, what metadata is required, etc.)

Interpretation

12g-ACE = *A gradient-based honesty-adaptation rule.*

It ensures the system becomes stricter when inconsistencies appear and more efficient when behavior is stable.

2. 12g-AFL - Adaptive Feedback Lineage

(The lineage trust-weight update)

12g-AFL updates the **trust weight** for lineage consistency over time.
Flow:
s*t* → *w*{t+1}
General Form:
w_{t+1} = λ *w_t* + *(1 - λ)* s_t
Where:

- $\mathbf{w_t}$ = trust weighting
- $\mathbf{s_t}$ = latest consistency score
- **λ** = smoothing factor (0–1)

Interpretation

12g-AFL = *A memory of past honesty, adjusted each cycle.*

It influences:

- verifier strictness in Loop 4
- event requirements in Loop 1
- tolerance thresholds
- confidence decay
- lineage branch weighting

3. How ACE and AFL Work Together

They combine into an adaptive dual-parameter system:

- **θ** controls *how the system expects events to behave*
- **w** controls *how strict the lineage checks become*

Combined Compact Diagram

s*t* → *L_t* → ∇*L_t* → *θ*{t+1} → (feeds Loop 1)
↘
→ w_{t+1} → (feeds Loops 1 & 4)

4. Why This Math Matters

Without ACE/AFL, you can only detect a lie.

But with this adaptive controller:

the system *learns* and tightens itself

bad actors are permanently downgraded

honest behavior becomes easier and cheaper

the network moves toward stable integrity

trust emerges from evidence, not assumption

This is the mathematical backbone that makes PoCI:

- self-correcting
- adversarially robust
- scalable
- and future-proof

No static rule system could ever do this.

Summary:
12g-ACE:
A gradient-driven equation that updates internal parameters (θ)
based on verifier signals to maintain adaptive coherence.

12g-AFL:
A lineage-weighting update rule that adjusts the system's trust memory (w)
using consistency scores over time.

Together:
They form a dual-parameter adaptive controller that enforces
compute integrity through ongoing feedback, not static rules.

The 12g-ACE / 12g-AFL framework is mathematically well-formed and built on standard learning and control ideas (gradient descent + exponential averaging). The structure is sound, but like any adaptive system, its real-world behavior will depend on implementation details, tuning, and testing.
It's a **solid foundation**, not a "solved forever in theory" object yet.