Verification-Feedback Inference Function Evolution

by Nick Clark | Published April 25, 2026

What Falsifiable Inference Specifies

When a behavior-inferred intent observation is produced, it is paired with a temporal commitment: the inference predicts a specific behavior pattern within a specified window. 'Vehicle B is preparing to merge' commits to a window of N seconds and a specific behavior pattern (lateral motion in the merge direction, signaling, gap-acceptance).

When the window expires, the actual behavior is observed and compared to the prediction. The agreement is a credentialed verification observation against the originating inference. The verification observation propagates through the mesh; aggregated verification observations modulate the inference function's parameters.

Why Frozen-At-Training Inference Functions Fail in Adversarial Domains

Adversaries adapt. The behavior patterns that distinguished hostile-intent six months ago do not match adversaries' current patterns. The classifier that detected last year's drone-swarm formations does not detect this year's. The fraud-detection model trained on last quarter's transaction patterns misses this quarter's.

Verification-feedback closes the loop. The architectural primitive makes every inference structurally falsifiable; the falsifications accumulate; the inference function evolves toward higher accuracy under continuous evaluation.

How the Closed Loop Operates

Each inference function publishes its predictions as credentialed observations with temporal commitments. The mesh records the predictions; when temporal windows expire, observed behavior is compared and verification observations are signed by the verifying agents. The verification observations propagate back to the inference function's authoring authority.

The authoring authority aggregates verification observations across deployments, identifies inference functions whose verification track record is degrading, and proposes parameter updates or replacement functions. Updates are governance-credentialed.

What This Enables for Adversarial-Aware Architectures

Defense classification systems, fraud detection, anti-money-laundering, intrusion detection, and counter-drone systems all face the same adversarial co-evolution problem. Verification-feedback evolution provides the architectural primitive that makes adaptation a structural property rather than a periodic operational effort.

The architecture also produces audit-grade inference quality metrics. Every deployed inference function has a verification track record; the track record is itself a credentialed observation that consumers admit through their policy.