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 that a specific behavior will follow 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). The commitment is part of the credentialed observation.

When the window expires, the actual behavior is observed and compared to the prediction. The agreement is itself a credentialed observation — a verification observation against the originating inference. Aggregated verification observations modulate the inference function's parameters: functions that consistently match observed behavior gain weight; functions that consistently miss are demoted; new functions are proposed and tested under sandboxed admission before promotion.

Why Frozen 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.

The frozen-at-training-time pattern is structural: classifiers are trained, deployed, and then operate without architectural mechanism for evolution. Periodic retraining attempts to compensate but lags adversarial adaptation by the retraining cycle. Verification-feedback closes the loop within the operational tempo, making every inference part of the training signal for the next inference.

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: a new version is signed, sandbox-evaluated by consumers, and admitted through composite admissibility. The cycle compresses the adversarial-co-evolution timeline from quarters to days.

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. Inference quality becomes governable rather than opaque. The patent positions the primitive at the layer adversarial-aware classification needs as adversaries' adaptation tempo continues to compress.