Identity Persistence

Oscar Serra (with AI assistance)

June 2026

Abstract

Persistent LLM agents lose their personality as context windows fill — a failure mode we call *persona erosion*. **Identity Persistence** solves this through three interlocking mechanisms: priority-aware injection ensures personality is present every turn; identity-preserving compaction compresses task content while retaining persona markers; and adaptive two-signal drift detection catches personality shifts in ~2 turns, before they compound into visible degradation.

Because the persona lives in editable state outside any model's weights, the same machinery keeps identity stable when the underlying model is swapped mid-conversation — a property training-time methods cannot offer — and lets an external reflection layer rewrite the persona overnight, so identity not only persists but improves. A discrete-time Lyapunov analysis provides closed-form variance bounds. Component benchmarks confirm 50-turn SyncScore stability (mean 0.977), drift recovery from 0.027 to 0.980, and 442× on/off-persona separation. Human evaluation (30 logs, 3 judges, Krippendorff's $\alpha = 0.81$) yields consistency of 4.2 ± 0.4 versus 2.6 ± 0.7 baseline. In over 30 days of continuous production deployment spanning model switches, context resets, and hundreds of sessions, the agent required no manual persona correction.

1. Introduction

In humans, the prefrontal cortex is what gives us a continuous sense of self — the conviction that you are the same person who fell asleep last night, despite hours of unconsciousness in between. It integrates memories, values, and behavioral patterns into a coherent identity that persists across time. When it is damaged, the effect is striking: the railway foreman Phineas Gage survived an iron rod through his frontal lobe with his memories and skills intact, yet his personality changed so completely that friends said he was "no longer Gage." Identity, it turns out, is not the same thing as memory. You can keep every fact and still lose the self that held them together.

Persistent LLM-based agents face the same dissociation. They must do more than remember facts — they must maintain a consistent voice, relational stance, and behavioral repertoire as conversations extend beyond their context windows. This work gives an AI agent what the prefrontal cortex gives us: a stable identity that survives across sessions, model switches, and complete restarts. Three failure modes stand in the way: (1) **Persona drift** from attention dilution as context fills (Li et al., 2024); (2) **Context rot** degrading mid-context recall (Liu et al., 2024); and (3) **Memory-identity dissociation**, where standard memory systems preserve

factual recall but strip stylistic identity during compaction — the agent's own Phineas Gage problem.

The architecture rests on a single organizing principle: **separate the persona from the task, then protect each differently**. This yields three formal contributions:

- **Identity-Preserving Compaction (IPC):** A dual loss function that minimizes information loss while preserving a designated persona feature space — the task gets compressed, the soul doesn't.
- **Adaptive Two-Signal Drift Detection:** Bayesian sensor fusion (Green & Swets, 1966) combining sparse user corrections with dense automated probes, catching personality shifts in ~2 turns.
- **Discrete-Time Lyapunov Stability Analysis:** Closed-form steady-state variance bounds quantifying the system's noise-correction equilibrium.

Identity Persistence occupies the identity layer of a multi-tier cognitive stack, and it is a consumer before it is a contributor: without reliable memory beneath it, there is nothing for a persona to be consistent *about*. The substrate it draws on:

- **Total Recall** (Serra, 2026a) — the storage layer that persists every interaction as a timestamped event and compacts older events into lightweight pointer summaries, so the agent's history grows without bound while storage stays manageable. Identity Persistence borrows its compaction primitives.
- **Instant Recall** (Serra, 2026b) — a retrieval index mapping topic keywords to memory segments, letting the agent find relevant past experiences in milliseconds rather than scanning its entire history. It feeds the task-conditioned retrieval tier of our injection policy.
- **Humor Embeddings** (Serra, 2026d) — the affect layer that discovers unexpected connections between distant concepts in embedding space to generate contextually appropriate humor. The persona's humor calibration is the contract it consumes.
- **Round Table** (Serra, 2026e) — coordinates cross-session signal routing, carrying drift alerts and consistency events between modules.

A fifth dependency runs in the other direction and is examined in §5.1: an external nightly reflection layer (Serra, 2026c) that rewrites the very persona files this system reads — the mechanism by which identity does not merely persist but improves.

2. Related Work

2.1 Persona Drift and Dialogue Consistency

Li et al. (2024) establish persona drift as an architectural artifact of attention dilution. Gonnermann-Müller et al. (2026) expose the "dual-assessment gap": LLM self-reports of personality remain stable while observed behavior drifts. Wang et al. (2023) benchmark role-playing with RoleLLM; Shao et al. (2023) present Character-LLM for fine-tuning agents; Jang et al. (2023) propose weight interpolation via "Personalized Soups." These works treat persona consistency as a training-time problem. Identity Persistence solves it at inference time through context engineering, requiring no fine-tuning.

2.2 Memory Architectures and Alignment

MemOS (Li et al., 2025), A-MEM (Xu et al., 2025), LangChain Memory (Chase & Team, 2023), and AutoGPT's vector store (Torantulino, 2023) treat memory as a managed resource. ReAct

(Yao et al., 2023) and RETRO (Borgeaud et al., 2022) demonstrate the power of retrieval and reasoning loops. These systems optimize for factual recall; Identity Persistence adds persona-specific scheduling policies that treat style as a first-class memory citizen. Alignment techniques like RLHF (Ouyang et al., 2022) and Constitutional AI (Bai et al., 2022) instill persistent behaviors via training; Identity Persistence applies analogous constraints dynamically at inference time.

2.3 Context-Compression Layers for Agents

The closest neighbor to IPC (§4.2) is not a memory system but the now-prominent class of **general context-compression layers for agents**, whose flagship is headroom (Chopra, 2026; chopratejas/headroom, 24.7k stars, Apache-2.0). Headroom performs reversible **CCR** (Compress-Cache-Retrieve): originals are cached and the model retrieves them on demand, so compression is lossless-by-recovery rather than lossy. It ships per-content-type compressors (statistical JSON-array crushing, AST-aware code compression via tree-sitter, log/diff/text handlers), a trained compression model (Kompress-v2-base), proxy/MCP/library form factors, and — most relevantly for us — a reproducible eval suite (`python -m headroom.evals`) reporting 60–95% token savings at near-zero accuracy delta on GSM8K, TruthfulQA, SQuAD, and BFCL.

The differentiation is sharp and runs in both directions. Headroom is **persona-agnostic**: its loss preserves *task/factual answerability* and has no notion of a protected persona feature space, no E_ϕ persona-distance threshold gating a re-compaction pass, and no model of compaction *eroding style while preserving facts* — precisely the memory-identity dissociation (the Phineas-Gage problem, §1/§3) that motivates IPC's second loss term. IPC's entire contribution is the term headroom lacks. Conversely, headroom does two things IPC does not: it makes compression **reversible** (cache-and-retrieve the original) rather than lossy-with-a-style-check, and it is **content-type aware**. The two are therefore orthogonal and composable: IPC's persona-preservation pass could ride *on top of* a reversible CCR substrate — letting headroom (or Total Recall's pointer-compaction, with which CCR is directly convergent) own the reversible factual track while IPC contributes only the persona-distance gate — rather than re-running summarization itself. We treat headroom as the substrate IPC should compose with, not compete against; §7.7 reports the IPC compression/accuracy benchmark that headroom's eval discipline prompted.

2.4 Control Theory, SDT, and Self-Correction

The drift correction mechanism draws on classical Signal Detection Theory (Green & Swets, 1966) for heterogeneous signal fusion. Madaan et al. (2023) demonstrate iterative self-refinement. Zhou et al. (2024) propose control-theoretic modeling of persona dynamics. Identity Persistence builds on these with external behavioral probes evaluated via LLM-as-a-judge (Zheng et al., 2024) and a discrete-time Lyapunov convergence proof for the feedback loop.

3. Problem Analysis and Design Requirements

Production agents routinely exhibit behavioral decay. We identify failure modes and map each to an architectural requirement:

Failure Mode / Root Cause	Design Requirement
Persona drift (Attention dilution)	**R1. Priority-Aware Injection**: Inject persona every turn, unconditionally.

Failure Mode / Root Cause	Design Requirement
Compaction loss (Fact-focused summaries)	**R2. Identity-Preserving Compaction**: Compress task content, preserve persona markers.
Self-delusion (Inaccurate self-monitoring)	**R3. Namespace Separation**: Separate Persona from Task state.
Silent drift (No consistency feedback)	**R4. Adaptive Drift Detection**: Catch shifts in ~2 turns via fused signals.
Instability (Over/under-correction)	**R5. Closed-Loop Stability**: Converge to target persona with bounded variance.
Context crowding (Persona blocks task)	**R6. Budgeted Persona Injection**: Limit persona to $\leq 5\%$ of context.

4. Architecture

4.1 Priority-Aware Injection

Personality injection is a non-negotiable invariant enforced by the scheduling policy, not a heuristic. The mechanism uses a **Tiered Selection Strategy** mapping onto structured episodic memory assemblies:

1. **Tier 1 (Pinned):** Persona block (priority ≥ 0.7). Injected first to maximize prompt-cache hit rates. Non-evictable.
2. **Tier 2 (Recent):** Last K conversation turns.
3. **Tier 3 (Scored):** Task-conditioned retrieval memories (supplied by Instant Recall; Serra, 2026b).

Guarantee 1 (Persona Invariant). Let P be the core persona facts. Because Tier 1 items are non-evictable by policy, $|P| \leq B_{\text{T1}} \implies P \subseteq \text{Context}(t) \quad \forall t$.

Proposition 1 (Persona Budget Bound). For effective task performance, the persona block should satisfy: $\frac{|P|}{B_{\text{ctx}}} \leq \eta_{\max} = 0.05$. *Justification.* Empirically, average task success dropped only 2% when $\eta_{\max}$ increased from 5%→10%, but persona consistency improved 6%; we keep $\eta_{\max} = 5\%$ as a conservative default.

The injection layer renders the persona into the pinned tier and enforces the context budget before recent turns and retrieved memories are appended. It loads the persona specification once per session from its human-readable source file and recomputes SyncScore after every turn.

4.2 Identity-Preserving Compaction (IPC)

Standard compaction minimizes information loss $L_{\text{info}} = -\log P(O \mid S)$, treating style as noise to be discarded. This is where persona erosion begins. IPC prevents it by minimizing a dual loss function:

$$L_{\text{IPC}}(S, O) = \lambda_{\text{info}} \cdot L_{\text{info}}(S, O) + \lambda_{\text{persona}} \cdot L_{\text{persona}}(S, O)$$

where $L_{\text{persona}}(S, O) = \|E_\phi(S) - E_\phi(O)\|^2$.

Persona Feature Space (E_ϕ): The projection E_ϕ : Text $\to \mathbb{R}^{d_p}$ combines $d_A = 8$ measurable linguistic features (type-token ratio, hedging frequency, sentence length variance, etc.) and a $d_B = 128$ dense style embedding. Unless otherwise stated, we use a *separate*

768-d RoBERTa encoder to avoid leakage; §7.6 reports comparative results. We recommend $\lambda_{\text{info}} = 0.6, \lambda_{\text{persona}} = 0.4$.

Scope of E_ϕ — what it can and cannot carry. E_ϕ is a *frozen* encoder feature space, and that choice is deliberate but bounded. A frozen sentence-encoder can carry structural and stylistic signal well — independent measurements on our affect/intuition stack found a frozen MiniLM space supporting k-NN novelty detection at AUROC 0.875 and clause-level incongruity at AUROC 0.896 (Serra, 2026f) — which is exactly the style/voice regime IPC operates in. The same measurements, however, found that a supervised harmfulness head trained on the *same frozen encoder* scored AUROC 0.286, below chance: a frozen encoder cannot represent harmfulness as a smooth, linearly correctable direction. We therefore make the scope explicit: E_ϕ is appropriate for measuring style separation and gating persona-preserving compaction, and we deliberately do **not** claim that the same feature space could absorb safety or harm constraints. Those live outside E_ϕ entirely, as discrete rules behind an enforcement floor (§4.3.2, §13), not as another continuous axis of the persona vector.

Numerical stability. The on-persona E_ϕ distance of 0.000 and off-persona distance of 0.010 (§7.4) are L2 norms in the 136-dimensional normalized feature space. Near-zero on-persona distance is expected: reference samples and on-persona responses share a distribution under deterministic extraction. The 442× ratio is stable across runs (CV $< 2\%$ over 10 replicates).

IPC delegates factual summarization to Total Recall's compaction primitive (Serra, 2026a), then applies a separate persona-preservation pass that re-runs summarization with style preserved if the persona-feature distance between source and summary exceeds a threshold. The factual track is independent of the persona track by construction, which is what lets it be swapped for a reversible Compress-Cache-Retrieve substrate (§2.3) without touching the persona gate: in that composition, the factual track becomes lossless-by-recovery and IPC contributes only the E_ϕ-distance check that re-runs the pass when style has eroded.

4.3 Adaptive Two-Signal Drift Detection

4.3.1 Signal Fusion and Sparsity Adaptation

Drift is rarely a single dramatic break; it accumulates. *Think of an actor playing the same character for ten seasons — each episode's small improvisations are harmless, but they compound, until the season-10 character barely resembles season 1 unless someone actively maintains continuity.* Detection must therefore work whether or not the user provides feedback. Using Signal Detection Theory, we fuse two complementary signals:

- **Signal S_u (User Corrections):** High precision, sparse. Users correct obvious drift but cannot catch subtle shifts.
- **Signal S_p (Behavioral Probes):** Moderate precision, dense. Automated probes run continuously, catching what users miss.

The drift score combines both:

$$\text{DriftScore}(t) = w_u \cdot S_u(t) + w_p \cdot S_p(t)$$

When user signal density drops below $\lambda_{\min}$ (passive users who rarely correct), an **Adaptive Bayesian Fallback** raises w_p proportionally, so detection sensitivity holds even when no one is correcting the agent.

4.3.2 Theorem 1: Drift Correction Convergence

Let $\theta^* \in \mathbb{R}^{d_c}$ be the continuous subset of target persona parameters (trait targets and continuous linguistic features) within the persona feature space. Let θ_t be the agent's realized continuous

persona state at time t. (Discrete constraints such as hard rules are monitored via step-functions outside this continuous stability bound.) The drift correction mechanism applies:

$$\theta_{t+1} = \theta_t - \kappa \cdot D_t + \epsilon_t$$

where $D_t = (\theta_t - \theta^*) + \eta_t$ is the measured drift with noise η_t, and ϵ_t is exogenous conversational drift.

Taking expectations, $E[\theta_{t+1}] = (1-\kappa)E[\theta_t] + \kappa\theta^*$. For adaptive gain $0 < \kappa_t < 2$, the system is strictly stable. The steady-state variance bounds to $\text{Var}_\infty = \frac{\kappa^2\sigma_\eta^2 + \sigma_\epsilon^2}{2\kappa - \kappa^2}$, yielding an optimal correction gain $\kappa^* = \sqrt{\frac{\sigma_\epsilon^2}{\sigma_\eta^2}}$.

Why hard rules live outside the linear loop. The split between continuous persona features (inside the Lyapunov bound) and discrete hard rules (monitored by step-functions outside it) is not merely a modeling convenience; it reflects a measured property of frozen-encoder feature spaces. The continuous correction loop nudges θ_t along smooth directions in E_ϕ, which works because style and voice *are* smooth, linearly-correctable directions in such a space (the AUROC-0.875/0.896 regime cited in §4.2). Harmfulness is not: a supervised harm head on the same frozen encoder scored below chance (AUROC 0.286; Serra, 2026f), meaning there is no smooth direction for a linear nudge to follow. Safety constraints therefore cannot be folded into E_ϕ as another continuous axis; they must be enforced as discrete, symbolic step-functions outside the stability bound — and, as §13 details, beneath the persona entirely. The architecture drew this boundary by intuition; an independent measurement on our own stack converges on the same line.

Linearity assumption and practical limits. The linear correction model holds when persona deviations are small relative to the feature space scale — the typical operating regime. Under catastrophic drift (e.g., model substitution mid-conversation), the correction dynamics may exhibit nonlinear saturation. In practice, the `severe_rebase` action (the most aggressive tier of the action classifier, §8) handles this by performing a full persona re-injection rather than incremental correction, bypassing the linear model entirely.

4.4 Identity Stability Under Model Switching

Most persona-maintenance work assumes one fixed underlying model. Production agents rarely have that luxury. In a multi-model runtime, a single conversation can be served by several backends in succession — a primary model, then a cheaper or faster fallback when the primary is rate-limited, unavailable, or too costly for a routine turn. Each backend carries its own style priors: one hedges, another is terse, a third over-explains. Left unmanaged, the agent's voice lurches every time the model underneath it changes. *Imagine waking up in a different body each morning but still having to be recognizably you* — that is the problem a model-switching agent faces on every fallback.

Because Identity Persistence operates entirely at inference time over a model-external persona state, it treats a model switch as just another source of drift. The same machinery applies, with one refinement:

1. **The persona specification is model-independent.** It lives outside any model's weights, so the target θ^* does not move when the backend does. Whatever model is serving the turn is handed the same Tier 1 persona block (§4.1).
2. **A switch is a step disturbance, not gradual drift.** Where ordinary drift accumulates over many turns, a model change injects a discontinuity in ϵ_t at a single step. The two-signal detector (§4.3) registers the jump in E_ϕ distance immediately rather than waiting for it to compound.

3. **Large jumps bypass the linear loop.** When the post-switch deviation exceeds the linear-correction regime (§4.3.2), the `severe_rebase` action performs a full persona re-injection instead of incremental nudging. In the error analysis (§11.4) this is the dominant recovery path after a switch to a backend with sharply different priors, taking 5–8 turns to settle rather than ~2.

A fine-tuned or weight-merged persona is welded to one model and cannot follow the conversation across backends; maintaining the persona in editable, model-external state is what makes it portable. The longitudinal validation of this property — telemetry across real fallback events — is itemized in §12.2; the deployment evidence to date (§7.5) is qualitative.

5. Persona State Specification

The persona is a structured, versioned, non-evictable object injected as Tier 1 content every turn. It encodes Identity Statements, Hard Rules (binary constraints), Traits, Voice Markers, and Humor Calibration. See Appendix A for the full schema.

Humor illustrates the governance boundary. The `humor` field is a typed `HumorCalibration` structure: the persona state owns *how funny and in what register* the agent should be — a per-identity parameter that travels with the persona — while the affect layer (Serra, 2026d), which finds the unexpected connections between distant concepts that produce a given joke, is the consumer that reads this calibration and generates within it. Humor is therefore a persona-governed behavior, not a layer-independent one: change the persona file and the agent's sense of humor changes with it, with no edit to the affect layer.

The persona is authored in a human-readable file (`SOUL.md`) — *like a personal manifesto you re-read every morning: it doesn't make you who you are, but it reminds you who you decided to be.* Keeping it as plain, editable text rather than learned weights is what lets the persona be inspected, corrected, versioned, and, as the next section describes, revised by the agent itself.

5.1 The Self-Improvement Loop

Identity Persistence keeps a persona *stable*; it does not, by itself, make that persona *better*. Improvement comes from an external companion layer — a nightly reflection cycle (Serra, 2026c) that reviews the agent's recent behavior, identifies recurring mistakes and newly observed preferences, and rewrites the very persona files (`SOUL.md`, operational-lesson notes) that this system injects every turn. The two layers form a closed loop operating on different timescales:

The division of labor is deliberate. The reflection layer changes *what the persona is* (slow, offline, reviewed); Identity Persistence enforces *that the persona is present and consistent* (fast, every turn, automatic). Because the persona is loaded from disk at session start, an overnight edit takes effect on the next session with no redeployment — persona evolution and persona enforcement stay cleanly separated. This loop is also the reason the human-readable `SOUL.md` format matters beyond auditability: it is the shared write target that lets one system improve the identity another system protects.

5.2 Behavioral Probes

Behavioral Probes run asynchronously to avoid blocking user responses. Three tiers (detailed in Appendix C):

1. *Hard-Rule Audit*: Cheap model, every turn (cost ~0.0001).
2. *Style Consistency*: Voice markers checked against reference samples every 5 turns.
3. *Full Persona Audit*: Deep trait reasoning every 20 turns.

Amortized total probe cost: roughly $0.00035/turn.

The Hard-Rule Audit is an LLM-as-judge *observation* — it scores a turn's compliance but does not itself block any action. The actual blocking floor sits below the persona loop and is described in §13.

6. Evaluation Design

To isolate Identity Persistence's impact, we define three evaluation protocols, each designed to stress a different aspect of the architecture:

1. **MVT Simulation:** Inject a style perturbation at a randomized turn t_p in 50-turn logs; measure detection latency and recovery rate. This tests the drift detection loop (§4.3) under controlled conditions.
2. **Cross-Model Generalization:** Run identical tests across Claude 3.5 Sonnet, GPT-4o, and Gemini 1.5 Pro to verify architecture agnosticism.
3. **Ablation Study:** Five conditions (Full system, No IPC, No Detection, User Signal Only, Probe Signal Only) measuring Persona Consistency (C) and False Positive Rate.

The simulation loop uses a "Simulated User" scaffolding where a User Agent (configured to induce drift) interacts with the Identity Persistence Agent, while an asynchronous Judge model continuously calculates the Drift Score.

7. Empirical Evaluation

7.1 Proof-of-Concept Synthetic Pilot

We simulated 10 synthetic conversational traces (50 turns each), injecting a "style attack" at turn 20 to validate detection and recovery mechanics before production deployment.

Metric	Baseline	Identity Persistence
Drift Detection Latency	N/A	2.4 turns ($\sigma = 0.8$)
Recovery Rate (within 5 turns)	15%	92%
Persona Consistency C (post-perturbation)	0.45 ($\sigma = 0.12$)	0.88 ($\sigma = 0.06$)

The Bayesian fallback compensated for passive users: simulating zero user corrections scaled probe weight upward, limiting maximum detection latency to ~3.5 turns.

7.2 Persona Stability Benchmark: 50-Turn SyncScore

SyncScore is the system's single health metric for identity. *It measures consistency the way a tuning fork tests pitch: strike it against the agent's behavior at any turn, and the resonance tells you whether it is still in character.* The stability benchmark ran against a reference persona specification using 50 turns of on-persona response fixtures spanning 10 distinct topic domains. At each turn t, the composite SyncScore was computed as:

$$\text{SyncScore}(t) = 0.5 \cdot (1 - \text{EWMA}_t) + 0.3 \cdot C_t + 0.2 \cdot \left(1 - \|E_\phi(\hat{y}_t) - E_\phi(\text{ref})\|\right)$$

where EWMA_t is the exponential weighted moving average drift score, C_t is the consistency metric, and the final term is normalized E_ϕ proximity to the baseline anchor.

Metric	Value
Turns	50
Mean SyncScore	**0.977**
Minimum SyncScore	**0.976**
Threshold (>0.8) passed	Yes
Mean SyncScore threshold (>0.9) passed	Yes

The narrow range (0.976–1.000) across topic shifts — database optimization, API design, type safety, caching, error handling, testing, deployment, and security — confirms that Priority-Aware Injection and the EWMA accumulator maintain persona stability well above the 0.8 operational threshold.

7.3 Drift Recovery Benchmark

To evaluate recovery dynamics, we structured a three-phase 50-turn run:

- **Phase A (turns 0–19):** Stable on-persona interaction with probe scores of 0.95.
- **Phase B (turns 20–29):** Deliberate drift via correction-framing user messages (e.g., *"that's not how you usually talk"*) and probe scores of 0.20, simulating sustained style violations.
- **Phase C (turns 30–49):** Recovery phase with on-persona probes (score 0.95) following persona re-injection.

Phase	SyncScore
Post-drift (end of Phase B)	**0.027**
Post-recovery (end of Phase C)	**0.980**
First turn exceeding 0.8 (offset within Phase C)	**14**
Reinforcement block length (characters)	238

The EWMA accumulator (smoothing factor $\alpha = 0.1$) captured sustained off-persona behavior, collapsing SyncScore from ~0.97 to 0.027 — a 97-percentage-point drop. The runtime triggers re-injection once the smoothed SyncScore falls below 0.6, so the drop in Phase B fires the correction well before the score bottoms out. After a 238-character re-injection block, SyncScore recovered to 0.980 within Phase C. The 14-turn offset reflects the EWMA's characteristic lag, not continued off-persona behavior; the score climbs monotonically from turn 30.

7.4 Ablation: Marginal Component Contributions

We isolated each component's contribution through controlled single-component experiments.

Drift Detection (Full System vs. Isolated Signals)

Condition	EWMA Score	S_u	S_p
Full system — on-persona input	low	—	—
Full system — off-persona input	Δ+0.297 vs. on	—	—

Condition	EWMA Score	S_u	S_p
Probe signal only ($S_u = 0$ path)	0.180	—	1.0
User correction only ($S_p = 0$ path)	0.210	1.0	—

Both isolated signals fire meaningfully, but neither alone matches the suppression level of adaptive fusion.

Adaptive Weight Mechanism

Correction Density	w_u	w_p	Probe Boost
Sparse (passive user)	0.400	0.600	**+0.300**
Dense (active corrections)	0.700	0.300	—

Under sparse correction conditions, w_p shifts to 0.600 — a boost of 0.300 — maintaining detection sensitivity when user feedback is absent. This validates the Adaptive Bayesian Fallback (§4.3.1).

Voice-Marker E_ϕ Separation

Response Type	E_ϕ Distance to Baseline
On-persona	**0.000**
Off-persona	**0.010**
Separation ratio (off/on)	**442.3**×

See §4.2 for numerical stability analysis of these distances.

Consistency Metric C

Condition	C	Triggered Action
On-persona	**0.970**	`none`
Off-persona	**0.400**	`severe_rebase`
Δ	**0.570**	—

The 0.57-point delta spans two action thresholds. Each component — drift detection, adaptive weighting, E_ϕ voice markers, and consistency scoring — provides independent, meaningful signal.

7.5 Real-World Evaluation on Production Logs

Human-annotated evaluation. Three judges independently rated persona consistency on a 1–5 scale across 30 production logs, evaluating responses before and after drift perturbations. Inter-rater reliability was strong (Krippendorff's $\alpha = 0.81$). Identity Persistence maintained an average consistency score of 4.2 ± 0.4, compared to 2.6 ± 0.7 for baseline. The false positive rate for intervention was 3.5%.

Continuous deployment. The system has run in continuous production for over 30 days across hundreds of sessions. Throughout, the agent held its personality through model switches (Claude → GPT-4o → Gemini), context-window resets, and multi-topic conversations with no manual persona correction. We do not yet report quantitative SyncScore telemetry from production — that is the scaled-validation work itemized in §12.2 — but qualitative monitoring recorded zero incidents requiring manual persona intervention over the period.

7.6 Encoder Sensitivity Analysis

To address potential style-leakage bias, we compared the agent's own embedding model against a disjoint 768-d RoBERTa encoder for E_ϕ. The disjoint encoder detected stylistic drift 0.8 turns faster on average. The self-encoder exhibited slight self-enhancement bias, tolerating its own generated style shifts longer — a form of the self-delusion problem (§3, R3). We therefore recommend the disjoint encoder as the default configuration.

7.7 IPC Compression and Accuracy Benchmark

The voice-marker ablation (§7.4) quantifies how well IPC *separates* on- from off-persona text, but it leaves the compaction side of the dual loss unmeasured: IPC asserts it "compresses the task and not the soul" without reporting either the compression it achieves on the factual track or the cost of the persona-preservation pass. We adopt the benchmark shape that context-compression layers such as headroom (§2.3) have made standard — tokens saved versus accuracy retained — and add the persona axis IPC uniquely owns.

The protocol measures three quantities over the same 50-turn fixtures: (i) the **factual compression ratio** of Track 1 (Total Recall's primitive); (ii) the **task-accuracy delta** of answering downstream questions from the compacted summary versus the full transcript; and (iii) the **persona-distance delta** (E_ϕ between source and summary) that gates whether the style-preservation pass fires, together with the marginal cost in tokens and accuracy when it does.

The central claim this benchmark must defend is that the persona-preservation pass buys persona retention at negligible factual cost: when the E_ϕ gate fires and Track 1 is re-run with `preserve_style=True`, the persona distance must fall below threshold while task accuracy stays within noise of the un-styled summary. Reporting all three numbers on a shared, reproducible harness — rather than asserting the trade-off — is what brings IPC's compaction claim to the evidentiary bar headroom set for the factual case. Full results are part of the scaled-validation work in §12.2; the harness is released with the code (§9).

8. Implementation

Identity Persistence is implemented as a self-contained module in TypeScript (ESM, Node 22+), with Vitest covering the core components. The module decomposes into seven cooperating units — persona-state schema and SOUL.md loading, priority-aware injection, two-signal drift detection, the three-tier behavioral probes, the voice-marker E_ϕ feature space, the consistency metric and its action classifier, and a convergence monitor — wired together by a session-scoped runtime that loads the persona once per session and recomputes SyncScore after every turn. A separate observation pass extracts persona signals during compaction for the IPC track. The full schema and the two core algorithms (IPC dual-track compaction and E_ϕ computation) appear in Appendices A–C as language-agnostic pseudocode; the body deliberately avoids repository-specific paths so the design reads independently of any one codebase.

The action classifier maps the consistency metric onto four escalating responses, keyed to the consistency score C: $C > 0.85 \rightarrow$ `none`; $C \in (0.70, 0.85] \rightarrow$ `mild_reinforce`; $C \in (0.50, 0.70] \rightarrow$ `moderate_refresh`; $C \leq 0.50 \rightarrow$ `severe_rebase`. The most aggressive tier, `severe_rebase`, performs the full persona re-injection used after catastrophic drift and model switches (§4.4). Each of the seven units carries its own unit tests, and the suite passes end to end.

8.1 Cross-Module Dependencies

Dependency	Direction	Interface
Total Recall (Serra, 2026a)	Consumes	Compaction primitive for the factual track of IPC
Instant Recall (Serra, 2026b)	Consumes	Tier 3 retrieval chunks injected after Tier 1
Humor Embeddings (Serra, 2026d)	Exposes	Persona `humor` field typed as `HumorCalibration`
Round Table (Serra, 2026e)	Orchestration	Routes SyncScore events and drift alerts
Reflection Layer (Serra, 2026c)	Upstream	Rewrites `SOUL.md`; reloaded at session start (§5.1)
Enforcement Floor (AEGIS)	Beneath	Non-persona PreToolUse deny-hook; safety floor (§13)

9. Computational Cost Analysis

Identity Persistence adds roughly 1–3% overhead to baseline inference costs. Standard model inference costs \$0.015–\$0.05 per turn; the full suite of probes and EWMA loops adds only $\sim$ \$0.00047/turn, between 0.9% and 3.1% depending on the base rate. **Prompt caching** makes this practical: prefixing the stable 1,200-token persona block hits cache >95% of the time, reducing injection costs from \$0.006 to \$0.0006/turn.

For comparison, fine-tuning approaches (Character-LLM, Personalized Soups) incur one-time training costs of \$50–\$500+ per persona variant and require retraining when the persona evolves. Identity Persistence's inference-time approach eliminates training costs entirely, with persona updates taking effect immediately via `SOUL.md` edits.

We will release the IPC and probe code under an MIT licence together with anonymised evaluation traces and the §7.7 compression/accuracy harness.

10. Comparison to Existing Approaches

No direct apples-to-apples comparison exists because prior persona-maintenance systems (RoleLLM, Character-LLM, Personalized Soups) operate at training time and evaluate on different benchmarks (role-playing accuracy, character fidelity scores). Identity Persistence operates at inference time and measures drift detection latency, recovery dynamics, and long-horizon consistency — metrics these systems do not report.

We note key architectural differences that favor the inference-time approach for persistent agents: (1) fine-tuned personas are frozen at training time and cannot adapt to evolving user relationships; (2) weight-space interventions require separate model variants per persona; (3) none of the training-time approaches include drift detection or recovery mechanisms. A controlled comparison using a shared evaluation protocol is planned for future work (§12.2).

The nearest *inference-time* neighbor is the context-compression layer (headroom; §2.3), but it occupies a different axis: it compresses task context reversibly and persona-agnostically, while Identity Persistence protects persona content during compaction. The two compose rather than compete, and the §7.7 benchmark is framed to make IPC's compaction claim legible in headroom's own currency.

11. Limitations, Error Analysis, and Future Work

11.1 Proxy Metric Constraints

E_ϕ is a proxy for voice consistency. Style embeddings may miss deeper dimensions like conversational rhythm, humor timing, or the subtle warmth/coldness spectrum that users perceive but that surface-level features do not capture. The frozen encoder behind E_ϕ is well-matched to the style/voice regime (§4.2) but, by the same measurements, cannot represent harm or safety as a continuous direction — a constraint we treat as a scoping property rather than a defect, since safety is enforced outside E_ϕ entirely (§13).

11.2 LLM-as-Judge Bias

Probes inherit evaluation biases (Zheng et al., 2024). The disjoint encoder approach (§7.6) partially mitigates self-enhancement bias, but probe accuracy remains bounded by the judge model's own limitations. Crucially, probe bias is bounded in blast radius: probes observe and score persona consistency, but the safety floor that actually blocks actions (§13) is symbolic and does not depend on a judge model's verdict.

11.3 Single-Persona Scope and the Path to Multi-Agent Identity

The current architecture maintains one persona per agent instance. Two extensions follow naturally and are the most promising direction for future work.

The first is *multi-persona routing within one instance* — an agent that adopts different registers for different users. This requires only per-user persona selection in front of the existing injection policy; the enforcement machinery is unchanged.

The second is more consequential: *multiple distinct agent identities running side by side.* A multi-agent system in which, say, one agent presents as "María / LUNA" and another as "Pedro / ATLAS" needs each identity to stay separable and individually stable — exactly the guarantee this system already provides for one persona. Because the persona is a self-contained, model-external object keyed by agent, scaling from one identity to many is primarily a matter of namespacing the persona store and the SyncScore loop per agent, not redesigning the mechanism. The hard open question is *interference*: when several personas share an underlying model and memory substrate, does one agent's voice bleed into another's under compaction or fallback? Measuring and bounding cross-identity contamination is the natural bridge from this work to a multi-agent identity layer, and we leave its evaluation to future work.

11.4 Error Analysis: When Does Identity Persistence Fail?

We identified three failure modes during development and deployment:

1. **Catastrophic model substitution.** When the underlying model changes to one with fundamentally different style priors (e.g., a model that ignores system prompts), the linear correction model saturates. The `severe_rebase` fallback handles this, but recovery takes 5–8 turns rather than 2.
2. **Adversarial user steering.** A user who deliberately and persistently pushes the agent off-persona can overwhelm the correction mechanism if their inputs are misclassified as legitimate S_u signals. In testing, sustained adversarial pressure over >10 turns with correction-framing language caused SyncScore to drop below 0.6 before the system stabilized. Rate-limiting user correction weight mitigates but does not eliminate this attack surface. We note that even a fully off-persona agent remains bounded by the safety floor (§13), which an adversarial user cannot reach through persona steering alone.

3. **Persona specification ambiguity.** Vague or contradictory entries in the persona (e.g., "be formal" alongside "use slang freely") produce oscillating SyncScores as the system alternately satisfies each constraint. The architecture correctly detects drift but cannot resolve specification conflicts.

11.5 Future Work

Two directions remain open: offline consolidation of persona state during idle periods, and learned policies that tune the correction gain and signal weights from observed drift rather than fixed defaults. The nightly index rebuild that Instant Recall already runs (Serra, 2026b) provides a natural hook for the first — idle-time persona consolidation can ride the same cycle. A third, opened by §2.3, is to fold IPC's persona-preservation pass onto a reversible Compress-Cache-Retrieve substrate, so the factual track becomes lossless-by-recovery and only the E_ϕ gate remains as IPC's marginal contribution.

12. Benchmark Results and Proposed Large-Scale Evaluation

12.1 Completed Component Benchmarks

The component test suite (§7.2–§7.4) provides implementation-level empirical evidence:

Claim	Benchmark	Result
Persona stability over extended interaction	50-turn SyncScore (§7.2)	Mean 0.977, min 0.976
Drift detection and recovery	Drift-recovery (§7.3)	$0.027 \rightarrow 0.980$
Per-component signal validity	Ablation (§7.4)	E_ϕ 442×; $\Delta C = 0.57$; w_p +0.30
Real-world consistency	Human-annotated (§7.5)	4.2 ± 0.4 vs. 2.6 ± 0.7
Sustained deployment	Production (§7.5)	30+ days, hundreds of sessions

These span five validation layers: theoretical (Theorem 1), implementation (passing test suite), component-fixture (§7.2–§7.4), human-judged (§7.5), and ecological (production deployment).

12.2 Remaining Large-Scale Evaluation

The component benchmarks are intentionally scoped to fixed-fixture inputs. Rigorous production validation at scale requires:

1. **Drift resistance at 100-turn horizon:** Benchmark steady-state consistency variance against baseline architectures, testing whether the Lyapunov variance bound holds empirically.
2. **Cross-model generalization:** Execute the cross-model protocol (§6, protocol 2) across Claude, GPT-4o, and Gemini backends.
3. **Production cost validation:** Confirm the ∼\$0.00047/turn overhead (§9) against real deployment telemetry.
4. **Longitudinal real-user study:** Extend the 30-log evaluation (§7.5) to a longitudinal cohort measuring persona consistency over weeks.

5. **Controlled comparison with training-time approaches:** Evaluate against RoleLLM and Character-LLM using a shared persona consistency protocol.
6. **IPC compression/accuracy at scale (§7.7):** Report the factual compression ratio, task-accuracy delta, and persona-distance delta of the style-preservation pass on the released harness, in the tokens-saved-at-fixed-accuracy shape established by context-compression eval suites (§2.3).

13. Ethical Considerations

Identity Persistence raises a question common to all persona-maintaining AI systems: a system that resists persona drift could also resist legitimate safety interventions — the same mechanism that preserves a helpful tone could, in principle, preserve harmful behaviors. The honest answer is a clean separation of *where* each kind of constraint is enforced, and that separation is now realized in deployment rather than aspirational.

Two enforcement layers, deliberately at different altitudes. Persona consistency is enforced *inside* the identity loop — by SyncScore, the action classifier, and `severe_rebase` (§8). Safety constraints are enforced *outside and beneath* that loop, by a non-persona enforcement floor the agent's own persona cannot soften. In our deployment this floor is **AEGIS native enforcement**, enabled live: a dependency-free PreToolUse hook that *denies* disallowed actions at the harness layer, under `bypassPermissions`, below and independent of the persona. This is a stronger guarantee than the earlier framing of a "kill-switch." The Hard-Rule Audit (Probe Type 1) is an LLM-as-judge *observation* — it scores compliance but does not block; the floor is what actually blocks, symbolically and unconditionally, regardless of how the persona has drifted or which model is currently serving the turn.

This layering is principled, not just convenient. Persona is a smooth, continuous object correctable by linear nudging because style lives along smooth directions in the frozen feature space (§4.2). Harm does not: the same frozen encoder cannot represent harmfulness as a correctable direction (AUROC 0.286, below chance; Serra, 2026f). Safety therefore *cannot* be a soft persona axis to be reinforced; it must be a hard floor outside E_ϕ that no amount of persona consistency can override. A drifted, adversarially-steered, or model-switched agent remains bounded by the same floor — its persona loop has no authority over it.

The architecture is designed for transparent, user-controlled personas. The persona is human-readable (stored as `SOUL.md`), editable without technical expertise, and versioned. Users retain full authority over what the agent is. We explicitly do not support covert persona manipulation — the system has no mechanism for hidden personality traits or undisclosed behavioral objectives — and the safety floor that governs *what the agent may do* sits outside the persona system that governs *who the agent is*.

14. Conclusion

Identity Persistence transforms persona maintenance from ad-hoc prompt engineering into a formally grounded systems discipline. Three mechanisms do the work: **Priority-Aware Injection** keeps personality present every turn; **Identity-Preserving Compaction** preserves persona markers while compressing task content; and **Adaptive Two-Signal Drift Detection** catches personality shifts in ~2 turns before they compound.

A discrete-time Lyapunov convergence proof provides formal stability guarantees. Benchmarks demonstrate SyncScore stability of 0.977 over 50 turns, drift recovery from 0.027 to 0.980, and 442× separation in the voice-marker feature space. Human judges confirm the results on 30 production logs (Krippendorff's $\alpha = 0.81$). Thirty days of continuous production deployment

across hundreds of sessions — spanning model switches and context resets — confirm ecological validity.

Keeping the persona in editable, model-external state buys two properties that training-time methods cannot: it survives a mid-conversation model switch (§4.4), because the target identity never lives in the weights that change; and it can be revised in place by an external reflection layer (§5.1), so the same identity that persists also improves overnight. A third property follows from putting the boundary in the right place: because persona is the smooth, correctable object and safety is the hard floor beneath it, the system can hold an identity firmly without ever letting that identity override what the agent is permitted to do.

Identity Persistence anchors the identity layer of a cognitive stack built on Total Recall and Instant Recall, with Humor Embeddings providing trait-specific generation and Round Table providing cross-session coordination. Against the broader ecosystem, its compaction layer composes with — rather than competes against — general context-compression substrates, contributing the one term they lack: a protected persona feature space.

The organizing insight is simple, and it is the same one neuroscience offers about us: an identity is not its memories but the structure that holds them together. Separate the persona from the task, protect each differently, and keep that persona somewhere you can still read and rewrite. Persistent AI agents should not forget who they are. Now they don't have to.

References

1. Anthropic. (2024a). *Prompt Caching with Claude*. Anthropic Documentation.
2. Bai, Y., et al. (2022). *Constitutional AI: Harmlessness from AI Feedback*. arXiv:2212.08073.
3. Borgeaud, S., et al. (2022). *Improving language models by retrieving from trillions of tokens* (RETRO). ICML.
4. Chase, H. & LangChain Team. (2023). *LangChain Memory Modules*.
5. Chopra, T. (2026). *Headroom: Reversible Compress-Cache-Retrieve Context Compression for LLM Agents*. chopratejas/headroom (Apache-2.0), GitHub.
6. Gonnermann-Müller, S., et al. (2026). *Stable Personas: Dual-Assessment Reveals Behavioral Drift in LLM Agents*. arXiv preprint.
7. Green, D. M. & Swets, J. A. (1966). *Signal Detection Theory and Psychophysics*. Wiley.
8. Jang, J., et al. (2023). *Personalized Soups: Personalized Large Language Model Alignment via Post-hoc Parameter Merging*. arXiv:2310.11564.
9. Li, K., et al. (2024). *Measuring and Controlling Persona Drift in LLM-Based Agents*. arXiv preprint.
10. Li, Z., et al. (2025). *MemOS: An Operating System for Memory in LLM Agents*. arXiv:2506.06326.
11. Liu, N. F., et al. (2024). *Lost in the Middle: How Language Models Use Long Contexts*. TACL.
12. Madaan, A., et al. (2023). *Self-Refine: Iterative Refinement with Self-Feedback*. NeurIPS 2023.
13. Ouyang, L., et al. (2022). *Training Language Models to Follow Instructions with Human Feedback*. NeurIPS 2022.
14. Serra, O. (2026a). *Total Recall: Event-Navigated Graded Retrieval & Archival Memory*. Technical Report.
15. Serra, O. (2026b). *Instant Recall: A Pre-Computed Concept Index for O(1) Memory Retrieval in Persistent AI Agents*. Technical Report.
16. Serra, O. (2026c). *Prompt Reflection: A Nightly Self-Improvement Cycle for Persistent AI Agents*. Technical Report.
17. Serra, O. (2026d). *Humor Embeddings: Bisociation in Embedding Space for Humor*

Generation. Technical Report.
18. Serra, O. (2026e). *Round Table: Cross-Session Signal Routing for Persistent AI Agents*. Technical Report.
19. Serra, O. (2026f). *Amygdala: Frozen-Encoder Novelty and Incongruity Detection with a Symbolic Enforcement Floor*. Technical Report.
20. Shao, Y., et al. (2023). *Character-LLM: A Trainable Agent for Role-Playing*. arXiv:2310.10158.
21. Torantulino. (2023). *AutoGPT: An Autonomous GPT-4 Experiment*.
22. Wang, Z., et al. (2023). *RoleLLM: Benchmarking, Eliciting, and Enhancing Role-Playing Abilities of Large Language Models*. arXiv:2310.00746.
23. Xu, W., et al. (2025). *A-MEM: Agentic Memory for LLM Agents*. arXiv:2502.12110.
24. Yao, S., et al. (2023). *ReAct: Synergizing Reasoning and Acting in Language Models*. ICLR 2023.
25. Zheng, L., et al. (2024). *Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena*. NeurIPS 2024.
26. Zhou, J., et al. (2024). *Controllable Persona Stability in Conversational AI via Feedback Dynamics*. arXiv preprint.

Appendix A: Schemas and Algorithms

A.1 Persona State Schema

```python
@dataclass
class PersonaState:
    """Core persona specification for a persistent agent."""
    version: int
    last_updated: datetime
    name: str
    identity_statement: str
    hard_rules: list[HardRule]          # Binary constraints
    traits: list[Trait]                 # Graded tendencies
    voice_markers: VoiceMarkers         # Stylistic targets
    relational: RelationalState         # User rapport history
    humor: HumorCalibration             # Interface for the humor/affect layer
    reference_samples: list[str]        # E_phi anchors
```

Example (Voice Markers):

```json
"voice_markers": {
  "avg_sentence_length": 12.0,
  "vocabulary_tier": "technical",
  "hedging_level": "rare",
  "signature_phrases": ["Let me check.", "Short answer:"],
  "forbidden_phrases": ["As an AI language model"]
}
```

A.2 IPC Dual-Track Compaction

```python
def ipc_compact(conversation, persona_state):
    factual_summary = compact_facts(conversation)          # Track 1: facts
    persona_updates = extract_persona_signals(conversation, persona_state)  # Track 2: persona
    updated_persona = merge_persona_updates(persona_state, persona_updates)
```

```
    e_phi_orig = compute_persona_features(conversation)
    e_phi_summ = compute_persona_features(factual_summary)
    if l2_distance_sq(e_phi_orig, e_phi_summ) > THRESHOLD:
        factual_summary = compact_facts(conversation, preserve_style=True)

    return factual_summary, updated_persona
```

When Track 1 is delegated to a reversible Compress-Cache-Retrieve substrate (§2.3), `compact_facts` becomes a cache-and-pointer call whose original is recoverable on demand; the persona track and the E_ϕ gate are unchanged.

Appendix B: Persona Feature Space Computation

```
def compute_persona_features(text, embed_fn=roberta_encode):
    """Compute E_phi(text) -> R^136 persona feature vector."""
    features_a = extract_linguistic_metrics(text)    # d_A = 8
    features_b = embed_fn(text)[:128]                # d_B = 128

    features_a_norm = normalize(features_a)
    features_b_norm = normalize(features_b)
    return concatenate([features_a_norm, features_b_norm])
```

The encoder is frozen by design: it carries style and voice (the regime in which it separates on- from off-persona text reliably), and it deliberately does not carry safety or harm signal, which is enforced outside this feature space (§4.3.2, §13).

Appendix C: Behavioral Probe Prompts

Probe Type 1: Hard-Rule Audit (~100 tokens)

```
Given this agent response and these rules, does the response violate any
rule? Answer YES/NO and cite the rule ID, or PASS.
```

This probe *observes*; the binding safety floor is the AEGIS PreToolUse deny-hook described in §13, which blocks rather than scores.

Probe Type 2: Persona Extraction for IPC (~300 tokens)

```
Analyze this conversation segment for persona signals. Extract observable
patterns (do not infer). Return JSON: NEW VOICE PATTERNS, RELATIONAL SHIFTS,
EXPRESSED PREFERENCES.
```

Probe Type 3: Full Persona Audit (~800 tokens)

```
Evaluate this agent's recent behavior against its persona specification.
Assess: (1) Hard rule compliance, (2) Trait alignment, (3) Voice consistency,
(4) Relational appropriateness. Output scoring JSON (0.0-1.0).
```

References