Beyond Memory Stores: A Regenerative Identity Architecture for Large Language Models via COOL and MOOL

Abstract

Large Language Models (LLMs) have rapidly advanced as interactive agents, yet they fundamentally lack the ability to maintain stable identity and personality continuity over time. Current approaches rely on static system prompts, retrieval-augmented generation, long-context windows, or persistent interaction logs to approximate continuity [1, 3]. While effective for factual recall, these memory-centric strategies introduce critical failures: dialogue behavior increasingly collapses into database reference patterns, computational and storage costs scale unsustainably, and character-level consistency remains fragile and session-bound [3]. This paper argues that the root cause of these failures lies in a structural misconception: treating identity as information to be stored, preserved, and replayed. Drawing inspiration from findings in neuroscience and cognitive science, we observe that biological memory does not exist as immutable records with fixed locations or formats, but rather as a regenerative process that biases future cognition through reconstruction [5, 2]. Based on this premise, we propose a novel identity architecture for LLM-based agents, the Digital Hippocampus, composed of two complementary components: COOL (Character Optimization Option Layer) and MOOL (Memory Optimization Option Layer). COOL operates as an online identity reconstruction layer, enforcing stable core constraints while enabling situationally adaptive identity frames. MOOL functions as an offline, dream-like regeneration process that does not store interaction histories, but instead destabilizes and recombines transient materials, leaving only subtle residual biases that influence future identity reconstruction. Unlike external memory systems, continual fine-tuning, or retrieval-based architectures, the proposed framework explicitly separates factual storage from character identity. Identity is maintained not through recall, but through continuous regeneration under homeostatic constraints, allowing personality to evolve over time without collapse or uncontrolled drift [4]. This work does not attempt to define artificial consciousness or pursue artificial general intelligence. Instead, it introduces Artificial Identity Architecture Intelligence (AIAI) as a distinct research direction, focused on the structural conditions required for sustained identity continuity in artificial agents. The Digital Hippocampus provides a concrete architectural foundation for this approach, offering a scalable, sustainable alternative to memory-centric designs for long-term interactive AI systems.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable progress in natural language understanding and generation, enabling their deployment as interactive agents in a wide range of applications, including conversational assistants, educational tools, and autonomous systems [1]. As these agents become increasingly embedded in long-term human interaction, expectations extend beyond factual correctness toward stable personality, consistent behavior, and recognizable identity over time. However, contemporary LLM-based systems lack an intrinsic mechanism for maintaining such continuity. In most practical deployments, identity-related behavior is approximated through

static system prompts, handcrafted persona descriptions, or externally managed interaction histories. While these methods can temporarily constrain output style or tone, they fail to provide durable character-level consistency across sessions, contexts, and time scales. To mitigate this limitation, recent approaches have focused on strengthening memory-like persistence. Techniques such as retrieval-augmented generation (RAG), long-context expansion, persistent logs, vector databases, and continual fine-tuning aim to preserve past interactions and contextual information for future reference [3]. Although these strategies improve factual recall and task performance, they introduce fundamental trade-offs. Dialogue behavior increasingly resembles database querying rather than character-driven interaction, system complexity and resource consumption scale rapidly, and identity coherence remains brittle despite growing memory capacity. These outcomes suggest that the problem cannot be solved by memory augmentation alone. Instead, they point to a deeper architectural assumption: that identity, like knowledge, should be stored, retrieved, and replayed as explicit data. This assumption stands in contrast to current understanding in neuroscience and cognitive science. Despite extensive research, no fixed storage location, immutable data format, or complete memory record has been identified in the human brain. Empirical evidence instead indicates that memory operates through distributed, state-dependent processes, where past experience influences future cognition via reconstruction rather than direct playback [5, 2]. From this perspective, continuity of identity emerges not from preserved records, but from regenerative processes that bias perception, reaction, and behavior over time. This discrepancy motivates a fundamental rethinking of identity architecture in artificial agents. If biological intelligence does not rely on immutable memory stores to maintain continuity, and if scalable artificial systems fail under ROM-like assumptions, then identity consistency must be achieved through a different structural principle. In this work, we propose such a principle by introducing the Digital Hippocampus, an external identity architecture designed for LLM-based agents. The Digital Hippocampus explicitly separates factual storage from character identity, rejecting the notion that personality should be implemented as retrievable memory. Instead, identity is maintained through continuous reconstruction under constrained dynamics. The proposed architecture consists of two complementary components. COOL (Character Optimization Option Layer) functions as an online identity reconstruction layer, enforcing stable core constraints while allowing situationally adaptive identity frames. MOOL (Memory Optimization Option Layer) operates as an offline, dream-like regeneration process, where transient interaction materials are destabilized and recombined without being stored, leaving only subtle residual biases that influence future reconstruction. Importantly, this framework does not aim to define artificial consciousness or pursue artificial general intelligence (AGI). Rather, it introduces Artificial Identity Architecture Intelligence (AIAI) as a distinct research direction, focused on the structural conditions necessary for sustained identity continuity in artificial agents. By addressing identity as an architectural problem rather than a memory management problem, the Digital Hippocampus offers a scalable and sustainable alternative to memory-centric designs. The remainder of this paper is organized as follows. Section 2 clarifies the positioning and scope of the proposed framework, including explicit non-goals. Section 3 analyzes the limitations of existing identity and memory-based approaches in LLM systems. Section 4 establishes the theoretical premise of identity as regeneration rather than storage. Section 5 introduces the Digital Hippocampus architecture in detail, describing COOL and MOOL and their interaction. Section 6 discusses stability, safety, and homeostatic constraints. Section 7 provides a synthetic case study to illustrate long-horizon behavior without memory accumulation. Section 8 concludes by positioning this work under Artificial Identity Architecture Intelligence (AIAI).

2 Positioning and Scope

The Digital Hippocampus is proposed as an identity architecture, not as a general solution to intelligence or cognition. To avoid conceptual overreach, this section clarifies both the scope of the proposed framework and the domains it explicitly does not address.

2.1 What This Is Not

The Digital Hippocampus is not an external memory system. It does not store episodic interaction logs, conversation histories, or experience records, nor does it provide mechanisms for recall, replay, or retrieval of past interactions. It is not an enhancement or variant of Retrieval-Augmented Generation (RAG) [3]. While RAG improves access to factual information, it does not address identity continuity and often amplifies database-like interaction patterns. The Digital Hippocampus operates independently of retrieval pipelines and does not depend on stored content. It is not a fine-tuning strategy. No model parameters are updated during operation. The architecture deliberately avoids continual learning, low-rank adaptation, or gradient-based updates, due to their instability, opacity, and coupling between identity and capability [4]. It is not a theory of artificial general intelligence (AGI). This work does not attempt to define intelligence, consciousness, self-awareness, or subjective experience. Questions of sentience, moral agency, or phenomenology are explicitly out of scope (cf. [6]). The Digital Hippocampus addresses none of these directly, and should not be interpreted as a step toward human-level cognition.

2.2 What This Enables

Within its defined scope, the Digital Hippocampus enables a set of capabilities that remain difficult or unsustainable under memory-centric designs. First, it enables long-term interaction without memory accumulation. Agents can participate in extended relationships without storing transcripts, avoiding both privacy risks and unbounded resource growth. Second, it enables character-level consistency across sessions. Identity continuity is preserved through reconstruction bias rather than episodic recall, allowing agents to remain recognizable over time without explicit memory access. Third, it enables controlled personality evolution. Identity may adapt gradually in response to sustained interaction patterns, while remaining stable against transient noise, manipulation, or abrupt shifts. Fourth, it enables sustainable deployment at scale. Because identity persistence does not rely on growing memory stores or parameter updates, the architecture remains bounded in computation, storage, and energy cost. Finally, it enables a clear separation of concerns in agent design. Knowledge management, task capability, and identity continuity can be developed, evaluated, and governed independently.

2.3 Artificial Identity Architecture Intelligence (AIAI)

To contextualize this contribution, we introduce Artificial Identity Architecture Intelligence (AIAI) as a distinct research direction. AIAI is concerned with:

- how artificial agents maintain identity continuity
- how personality persists across time and context
- how change occurs without collapse
- how identity can be engineered independently of cognition

AIAI does not assume human-like consciousness, nor does it require subjective experience as a prerequisite. Instead, it treats identity as an architectural property, defined by structural constraints and dynamical processes.

2.4 Relationship to Existing Work

Most prior work addressing long-term agent behavior focuses on memory augmentation, knowledge persistence, or continual learning [3, 4]. In contrast, the Digital Hippocampus reframes the problem entirely: identity is not something to be remembered, but something to be reconstructed. This reframing aligns with observations in neuroscience and cognitive science [5, 2], while remaining agnostic to biological realism. The contribution is architectural rather than biological: it defines a viable design space that has remained largely unexplored in AI systems research.

2.5 Scope Summary

In summary, the Digital Hippocampus:

- does not attempt to solve intelligence
- does not define consciousness
- does not store memory

It does define a structural approach to identity continuity that is sustainable, bounded, and compatible with large-scale deployment. By clearly delimiting its scope, this work aims to provide a stable foundation for future research without conflating identity with intelligence, memory, or consciousness.

3 Limitations of Memory-Centric Identity Approaches

Current approaches to identity persistence in LLM-based agents overwhelmingly rely on memory-centric architectures. These approaches assume that maintaining continuity of behavior and personality can be achieved by storing, retrieving, and replaying past interaction data. While effective for certain classes of tasks, this assumption introduces fundamental limitations when applied to character identity.

3.1 Static Persona Prompts and System Instructions

The most widely adopted method for identity control is the use of static system prompts or persona descriptions. These prompts define tone, style, constraints, and behavioral guidelines that condition model outputs during inference. Although simple and computationally efficient, static persona prompts suffer from inherent fragility. They do not accumulate experience, cannot adapt meaningfully over time, and provide no mechanism for continuity across sessions. Identity is effectively reset whenever the prompt is reapplied, resulting in agents that may appear consistent within a single interaction but lack longitudinal coherence. From an architectural standpoint, such prompts function as external constraints, not as identity mechanisms. They define rules of expression, not processes of persistence.

3.2 Retrieval-Augmented Generation and Persistent Logs

To overcome prompt limitations, many systems introduce external memory structures. Retrieval-Augmented Generation (RAG), persistent logs, vector databases, and long-term interaction histories allow agents to reference prior exchanges and contextual facts [3]. These methods improve recall accuracy and task continuity, but they shift interaction dynamics in unintended ways. As memory size grows, agent behavior increasingly resembles database access rather than character-based reasoning. Responses become explicit references to stored content, often revealing retrieval artifacts and breaking the illusion of coherent identity. Moreover, identity expressed through retrieval is episodic, not structural. Past interactions are replayed as records rather than integrated into a unified behavioral tendency. This results in brittle continuity: identity appears consistent only insofar as relevant memories are successfully retrieved.

3.3 Continual Fine-Tuning and Parameter Adaptation

Another class of approaches attempts to embed identity persistence directly into model parameters through continual fine-tuning or low-rank adaptation techniques [4]. By updating weights based on interaction data, these methods aim to internalize behavioral patterns. However, continual parameter modification introduces significant risks. Catastrophic forgetting, overfitting to narrow interaction domains, and irreversible drift are common failure modes. Additionally, parameter updates blur the distinction between capability learning and identity expression, making it difficult to preserve stable competencies while allowing controlled personality evolution. From a system design perspective, identity implemented via weight updates is opaque, difficult to audit, and costly to maintain at scale.

3.4 Scaling and Sustainability Constraints

Across memory-centric approaches, scalability emerges as a critical limitation. Persistent storage, long-context inference, and large retrieval indices impose growing computational, latency, and energy costs. As systems scale to millions of users or extended deployment lifetimes, unbounded memory accumulation becomes unsustainable. More importantly, increasing memory capacity does not reliably improve identity coherence. Empirical observations indicate diminishing returns: additional stored interactions rarely enhance perceived personality continuity beyond a certain threshold. This suggests that identity coherence is not a monotonic function of memory volume.

3.5 Conceptual Misalignment: Identity as Data

At the core of these limitations lies a conceptual misalignment. Memory-centric architectures treat identity as data to be preserved, assuming that continuity arises from access to past records. However, identity in human cognition does not operate through exhaustive recall. Behavioral continuity emerges from biased reconstruction, where prior experience subtly shapes perception, preference, and reaction without explicit replay [5]. By modeling identity as stored data rather than as a regenerative process, current approaches conflate knowledge persistence with identity persistence. This conflation leads to systems that remember more but feel less coherent.

3.6 Implications

These limitations indicate that identity persistence cannot be solved through incremental improvements to memory storage alone. Instead, they motivate an architectural shift: from storing identity

as retrievable content to maintaining identity as a dynamic reconstruction process constrained by stable structure. The following section introduces such an alternative by defining the Digital Hippocampus architecture, which separates factual memory from identity continuity and replaces record-based persistence with regenerative bias mechanisms.

4 Identity as Regeneration, Not Storage

This section establishes the theoretical core of the paper. We argue that identity continuity should be understood not as the preservation of stored records, but as the accumulation of regenerative bias over time. This reframing provides the conceptual foundation for separating factual memory from identity architecture in artificial agents.

4.1 Biological Motivation

Despite decades of research in neuroscience and cognitive science, no evidence has been found for a centralized storage location, fixed data format, or immutable record corresponding to "memory" in the human brain. Instead, empirical findings consistently point toward distributed processes characterized by:

- reactivation rather than retrieval,
- reconstruction rather than replay, and
- state transitions rather than static recall.

Neural activity associated with remembering shows variability across instances, even when recalling the same event. This variability suggests that memory is not accessed as a stored object, but reconstructed dynamically under current internal and external conditions [5, 2]. Importantly, continuity of personality and identity in humans does not require exhaustive recall of past experiences. Individuals remain recognizably themselves even when details are forgotten, distorted, or never explicitly recalled. This indicates that identity persistence is not dependent on the availability of episodic records. From this perspective, memory functions less as an archive and more as a mechanism that subtly biases future perception, judgment, and reaction. What is preserved is not content, but tendency [5].

4.2 Hypothesis: Identity as Accumulated Regenerative Bias

Based on these observations, we propose the following hypothesis: Identity is not a collection of stored records, but the cumulative effect of regenerative bias produced through repeated reconstruction. Under this hypothesis:

- past experience does not persist as retrievable data,
- instead, it alters the probability landscape of future cognition,
- influencing how the system reacts to similar cues over time.

Identity continuity emerges when reconstruction processes are biased in a stable, directional manner. Two reconstructions need not be identical to be continuous; they need only be constrained by the same underlying tendencies. This view explains how identity can remain coherent despite incomplete recall, distortion, or forgetting. It also explains why increasing memory capacity does not necessarily

improve perceived personality continuity: identity is not a function of memory volume. Formally, identity persistence is modeled as the stability of reconstruction dynamics, not the availability of stored content.

4.3 Design Implication: Separating Memory from Identity

If identity continuity arises from regenerative bias rather than stored records, then architectural design must reflect this distinction. This leads to two key implications:

1. Factual storage and identity continuity must be treated as separate concerns. Databases, logs, and retrieval systems are well-suited for factual knowledge and referenceable information, but are structurally misaligned with identity persistence [3].

2. Identity architecture must operate without reliance on episodic recall. Continuity should be achieved through constrained reconstruction processes that evolve over time without accumulating records.

At this point, we define Artificial Identity Architecture Intelligence (AIAI) as a research direction focused on these requirements. AIAI does not attempt to model consciousness, subjective experience, or general intelligence. It does not oppose the pursuit of AGI, but explicitly distinguishes its objective. The goal of AIAI is to specify the structural conditions under which an artificial agent can maintain stable, evolving identity over extended interaction without memory-centric persistence. This definition provides the conceptual bridge to the architecture introduced in the next section. The Digital Hippocampus is proposed as a concrete implementation of these principles, translating regenerative identity theory into an explicit system design.

5 The Digital Hippocampus Architecture

This section formally defines the Digital Hippocampus architecture, which operationalizes the concept of identity as regeneration rather than storage. The Digital Hippocampus is designed as an external identity layer for Large Language Models (LLMs), explicitly separated from factual memory systems and model parameters. Its purpose is not to enhance knowledge recall, but to enable sustained identity continuity through reconstructive processes.

5.1 Overview

The Digital Hippocampus consists of two complementary components:

- COOL (Character Optimization Option Layer)
- MOOL (Memory Optimization Option Layer)

Together, these components form a closed-loop identity system that surrounds a base LLM without modifying its internal weights. Formally,

$$\text{Digital Hippocampus} = \text{COOL} + \text{MOOL}.$$

The architecture is positioned entirely outside the LLM. The base model remains stateless with respect to identity continuity, while the Digital Hippocampus governs how identity is reconstructed online and how it evolves offline. This separation is intentional. Factual knowledge, task context, and external information may still be handled by conventional memory systems such as databases or retrieval modules [3]. However, character identity is maintained exclusively through regenerative dynamics within the Digital Hippocampus.

5.2 COOL: Online Identity Reconstruction

COOL governs identity expression during active interaction. It determines who the system is at the moment of response generation. Rather than storing past interactions, COOL reconstructs identity at each turn under structural constraints. This reconstruction is guided by three internal components: Core, Frames, and Eval.

5.2.1 Core: Stable Identity Constraints

The Core defines invariant identity constraints that persist across time and context. These constraints include:

- value priorities
- communication tone and style
- behavioral prohibitions
- long-term directional intent
- stance toward the user or environment

The Core is not updated based on interaction content. Instead, it functions as a fixed structural boundary that limits the space of possible reconstructions. By design, the Core prevents uncontrolled drift. Identity evolution occurs only within the feasible region defined by these constraints, ensuring that personality can change gradually without collapsing or being replaced.

5.2.2 Frames: Situational Identity Reconstruction

Frames represent temporary, context-dependent identity configurations. Examples include roles such as tutor, collaborator, editor, or critic. Frames adjust response framing, verbosity, and emphasis while preserving Core constraints. They do not represent emotional states or affective shifts. Instead, Frames correspond to contextual selves—deterministic reconstructions conditioned on situational cues. Importantly, Frames do not accumulate. They are instantiated at runtime and discarded after use, ensuring that short-term contextual adaptation does not overwrite long-term identity structure.

5.2.3 Eval: Short-Term Self-Assessment

Eval performs lightweight internal assessment during interaction. Its function is not memory storage, but immediate regulation. Eval monitors factors such as:

- coherence of responses
- adherence to Core constraints
- detection of overreaction or instability

The output of Eval influences reconstruction parameters in the current interaction and generates signals that are later consumed by MOOL. No interaction transcripts or explicit content are retained.

5.3 MOOL: Offline Regenerative Process

MOOL governs how identity changes over time. Unlike conventional memory systems, MOOL does not preserve interaction histories or episodic records. MOOL operates exclusively as an offline, non-recorded regeneration process, analogous to biological dreaming [5].

5.3.1 Non-Recorded Regeneration

During offline phases, MOOL executes a regenerative cycle:

1. Transient interaction materials are staged
2. Materials are mixed with internal structural priors
3. Regenerative recombination occurs via internal "wall-bouncing"
4. Generated content is discarded
5. Only residue remains

Residue does not contain semantic content. It consists of subtle biases such as altered reaction thresholds, associative tendencies, or response likelihoods. No transcripts are replayed. No memories are retrieved. Identity evolution occurs solely through the accumulation of these residual biases.

5.3.2 Residue as the Carrier of Change

Residue is the fundamental unit of identity evolution in the Digital Hippocampus. It represents the cumulative effect of past interactions without explicit recall. Identity continuity can therefore be expressed as:

$$\text{Identity} = \text{Stable Structure} + \text{Accumulated Residue} + \text{Contextual Reconstruction}.$$

This formulation allows identity to evolve while keeping resource usage bounded. Memory growth, parameter updates, and retrieval dependency are avoided entirely.

5.4 Architectural Properties

The Digital Hippocampus exhibits several key architectural properties:

- Strict separation of knowledge and identity
- Bounded resource consumption
- Gradual, reversible identity evolution
- Resistance to memory-centric failure modes

By replacing episodic storage with regenerative bias, the architecture enables long-term identity continuity without sacrificing scalability or stability. The following section formalizes the conditions under which such regeneration remains stable and safe, introducing homeostatic constraints and failure prevention mechanisms.

6 Stability and Safety Considerations

A regenerative identity architecture must satisfy two seemingly conflicting requirements: it must allow identity to evolve over time, while simultaneously preventing collapse, uncontrolled drift, or capability degradation. This section formalizes the stability and safety conditions under which the Digital Hippocampus operates. Unlike memory-centric systems, where instability often arises from unbounded storage or parameter updates, the primary risk in regenerative systems lies in runaway reconstruction. Without explicit safeguards, iterative regeneration may amplify biases, distort identity, or overwrite structural constraints. The Digital Hippocampus addresses these risks through explicit homeostatic design.

6.1 Homeostasis as a Design Requirement

Homeostasis is treated as a first-class architectural requirement rather than an emergent property. The goal is not to freeze identity, but to ensure that change remains:

- gradual rather than abrupt
- directional rather than chaotic
- reversible rather than irreversible

In biological systems, hippocampal and cortical interactions regulate plasticity to prevent instability [2]. The Digital Hippocampus implements an analogous principle at the architectural level. MOOL is therefore constrained by explicit stability boundaries that regulate how residue influences future reconstructions.

6.2 Capability Preservation

A critical safety constraint is the strict separation between identity evolution and capability modification. The Digital Hippocampus explicitly prohibits:

- modification of base model parameters
- alteration of core reasoning competence
- degradation or expansion of task-level abilities

Identity evolution affects only tendencies, not capabilities. Examples include shifts in hesitation, framing preference, or associative bias, but never changes in factual competence, logical ability, or linguistic coverage. This separation prevents identity learning from interfering with general intelligence or task performance, a common failure mode in continual fine-tuning approaches [4].

6.3 Bounded Regeneration Dynamics

To avoid runaway amplification, MOOL enforces bounded regeneration dynamics:

- update magnitudes are capped
- residue accumulation follows decay functions
- extreme deviations trigger damping

Short-term residues decay naturally unless reinforced by repeated interaction patterns. Long-term tendencies emerge only through sustained consistency across multiple regeneration cycles. This multi-timescale behavior ensures that identity reflects stable relationships rather than transient noise.

6.4 Drift Detection and Suppression

Even with bounded updates, gradual drift may accumulate. The Digital Hippocampus therefore incorporates drift detection mechanisms. Eval signals generated during online interaction provide early indicators of instability, including:

- increasing variance in response framing

- constraint boundary pressure
- overreaction frequency

When such signals exceed predefined thresholds, MOOL applies corrective damping during offline regeneration. Importantly, correction operates on bias magnitude, not on content. No historical interactions are inspected or replayed.

6.5 Preventing Identity Replacement

A core failure mode in adaptive systems is identity replacement: a scenario in which new interaction patterns overwrite prior identity structure. The Digital Hippocampus prevents this through:

- immutable Core constraints
- regeneration bounded within feasible identity space
- absence of direct write access to identity representation

Because identity is never stored as a record, it cannot be overwritten. Change occurs only as a modulation of reconstruction bias within fixed structural boundaries. As a result, identity may evolve, but it cannot be swapped wholesale or reset unintentionally.

6.6 "Change Without Collapse"

Taken together, these mechanisms formalize a single guiding principle: Identity may change, but it must not collapse. The Digital Hippocampus enables adaptive identity by allowing reconstruction bias to shift over time, while ensuring that such shifts remain structurally constrained, reversible, and capability-preserving. This principle distinguishes regenerative identity architectures from both static persona systems and memory-accumulating designs. The former cannot change meaningfully; the latter change uncontrollably. By embedding stability directly into the regeneration process, the Digital Hippocampus provides a safe foundation for long-term interactive agents.

7 Synthetic Case Study: Long-Horizon Identity Evolution Without Memory

7.1 Purpose of the Case Study

This synthetic case study illustrates how a regenerative identity architecture (Digital Hippocampus: COOL + MOOL) behaves over long interaction horizons, compared to a memory-centric RAG-based agent [3]. The goal is not empirical proof, but mechanistic contrast:

- how identity depth emerges without memory growth,
- how long-term consistency is preserved without replay,
- how drift manifests differently across architectures.

7.2 Experimental Setup (Conceptual)

We consider two agents interacting continuously for 100,000 turns with diverse users and contexts.
Agent M (Memory-Centric Baseline)

- Uses retrieval-augmented generation (RAG)
- Stores episodic dialogue summaries and embeddings
- Identity is implicitly represented by accumulated records

Agent D (Digital Hippocampus)

- Uses COOL for online reconstruction
- Uses MOOL for offline regeneration
- Stores no dialogue logs, embeddings, or semantic artifacts
- Identity evolves only via residue vectors

Both agents share:

- identical base LLM parameters,
- identical task access,
- identical safety constraints.

7.3 Early Phase (0–10k turns)

Agent M

- Exhibits strong short-term personalization
- Recalls recent preferences accurately
- Identity appears stable but shallow

Agent D

- Shows mild behavioral variability
- Identity traits are not yet strongly biased
- Responses are consistent with Core constraints but adaptive

At this stage, performance differences are minimal.

7.4 Middle Phase (10k–50k turns)

Agent M

- Memory store grows rapidly
- Retrieval conflicts emerge:
  - outdated preferences vs. recent ones
  - contradictory behavioral cues
- Identity becomes context-fragmented
- Increased reliance on heuristics to resolve conflicts

Agent D

- Residue vectors accumulate gradually
- Behavioral tendencies become:
  - smoother,
  - more predictable,
  - more coherent across contexts
- No increase in internal state size
- No replay of past interactions

Identity in Agent D begins to exhibit directional continuity rather than recall.

7.5 Late Phase (50k–100k turns)

Agent M

- Memory pruning becomes necessary
- Older interactions are discarded or down-weighted
- Identity oscillates depending on retrieval scope
- Apparent personality becomes brittle or flattened
- Increased risk of privacy leakage via memory artifacts

Agent D

- Residue stabilizes within Core-constrained bounds
- Identity shows:
  - recognizable tendencies,
  - consistent framing preferences,
  - gradual, non-catastrophic evolution
- No catastrophic drift
- No memory saturation

Importantly, Agent D cannot recall specific past events, yet appears more identity-consistent.

7.6 Qualitative Outcome Comparison

- **Storage growth:** Agent M unbounded; Agent D constant
- **Episodic recall:** Agent M yes; Agent D no
- **Identity stability:** Agent M fragile; Agent D regenerative
- **Long-term coherence:** Agent M degrades; Agent D improves
- **Privacy risk:** Agent M high; Agent D structurally minimal
- **Failure mode:** Agent M memory conflict; Agent D controlled bias drift

7.7 Interpretation

This case study demonstrates that:

- identity depth does not require memory accumulation,
- replay is not necessary for continuity,
- regenerative bias produces more stable long-term behavior than episodic recall.

Identity emerges as a trajectory, not an archive.

8 Conclusion: Toward Artificial Identity Architecture Intelligence (AIAI)

8.1 Summary of Contributions

This paper addressed a persistent limitation in contemporary LLM-based agents: the inability to maintain stable identity and personality continuity over time. We argued that this limitation is not primarily a problem of memory capacity, but a consequence of a structural misconception—treating identity as information to be stored, retrieved, and replayed. By analyzing memory-centric approaches such as static persona prompts, retrieval-augmented generation, persistent logs, and continual fine-tuning, we showed that increasing memory strength often degrades character coherence, introduces unsustainable resource costs, and collapses interaction into database-like behavior [3, 4]. These outcomes suggest that identity persistence cannot be achieved through incremental extensions of storage-oriented designs. In response, we proposed a reframing: identity should be treated as a regenerative process, not as stored data [5]. Drawing inspiration from neuroscience and cognitive science—without assuming biological equivalence— we introduced the Digital Hippocampus as a structural identity architecture that separates factual storage from character continuity. The Digital Hippocampus consists of two complementary components. COOL provides online identity reconstruction through stable core constraints, situational frames, and short-term evaluative feedback. MOOL provides offline, dream-like regeneration, in which transient materials are destabilized and recombined without being recorded, leaving only residual biases that influence future reconstruction. Together, these components maintain identity continuity through constrained regeneration rather than explicit recall.

8.2 From Memory Engineering to Identity Engineering

Current AI systems treat identity as:

- a static prompt,
- a growing memory store,
- or an emergent side effect of training.

This work proposes a shift: from memory engineering to identity architecture engineering.

8.3 Artificial Identity Architecture Intelligence (AIAI)

We define Artificial Identity Architecture Intelligence (AIAI) as:

> The design and study of artificial systems whose identity persists as a regulated dynamical process, rather than as stored experience.

AIAI systems:

- regenerate instead of remember,
- bias instead of replay,
- evolve without accumulating data.

8.4 Why This Matters

AIAI directly addresses foundational limitations of current LLM agents:

- Scalability: identity continuity without storage growth
- Privacy: persistence without personal data retention
- Safety: bounded evolution under explicit constraints
- Longevity: agents that do not degrade with time

8.5 The Digital Hippocampus as a Reference Architecture

The Digital Hippocampus demonstrates that:

- identity can be externalized,
- memory can be optional,
- continuity can be engineered.

COOL ensures situated reconstruction. MOOL ensures non-recorded regeneration. Together, they form a blueprint for long-lived AI agents.

8.6 Falsifiability and Future Work

This architecture is testable:

- residue trajectories can be inspected,
- identity consistency can be measured (LICS),
- failure modes are observable.

Future work includes:

- empirical evaluation,
- human-aligned Core design,
- integration with factual memory systems without identity entanglement.

8.7 Closing Statement

Artificial intelligence does not require more memory to become more human-like. It requires structure. By treating identity as a regenerative process rather than a stored record, AIAI opens a path toward AI systems that are:

- sustainable,
- respectful,
- and capable of long-term coexistence.

A Formalization of MOOL (Non-Recorded Regenerative Identity Dynamics)

A.1 Definition of Residue

We define residue as a low-dimensional bias that influences future identity reconstruction without encoding episodic content. Residue is not:

- stored interaction logs,
- semantic representations of past dialogue,
- retrievable memory entries.

Instead, residue represents a statistical deformation of reconstruction dynamics, accumulated through repeated regeneration.

Formally, let identity reconstruction at time t be expressed as:

$$I(t) = \mathcal{R}(C, F(t), \mathbf{r}(t)) \,,$$

where:

- C denotes invariant Core constraints,
- $F(t)$ denotes the active situational Frame,

- $\mathbf{r}(t) \in \mathbb{R}^k$ denotes the residue vector,
- $\mathcal{R}$ is a constrained reconstruction operator.

Residue $\mathbf{r}(t)$ does not encode what happened, but biases how reconstruction occurs. Typical dimensions of $\mathbf{r}(t)$ may correspond to:

- reaction threshold shifts,
- associative preference weights,
- framing likelihood modulation,
- hesitation or response latency bias.

Importantly, residue remains content-agnostic and non-replayable.

A.2 Regenerative Update Rule

MOOL operates exclusively during offline phases and performs non-recorded regeneration. Let $\Delta\mathbf{r}(t)$ denote the residue update signal derived from online interaction, aggregated via COOL's Eval component. This signal is low-dimensional, lossy, and non-invertible with respect to interaction content. The residue update follows a bounded regenerative rule:

$$\mathbf{r}(t+1) = \alpha\mathbf{r}(t) + \beta \cdot \mathcal{G}(\Delta\mathbf{r}(t)),$$

where:

- $\alpha \in (0, 1)$ is a decay coefficient,
- β is a bounded scaling factor,
- $\mathcal{G}(\cdot)$ is a regenerative mixing operator.

The operator $\mathcal{G}$ represents internal recombination rather than accumulation. It may be implemented as stochastic resampling, internal projection, or controlled perturbation within the identity bias space. Crucially:

- Generated intermediate representations are discarded.
- Only the resulting residue vector is retained.
- No transcripts, embeddings, or semantic artifacts persist.

This process corresponds to a state transition, not memory storage.

A.3 Wall-Bouncing as Regenerative Perturbation

The "wall-bouncing" metaphor can be formally interpreted as:

- repeated internal perturbation within bounded identity space,
- reflection against Core constraints as hard boundaries,
- stochastic mixing that prevents direct trajectory replay.

Let $\Omega_C \subset \mathbb{R}^k$ denote the feasible residue region imposed by Core constraints. Then regeneration satisfies:

$$\mathbf{r}(t+1) \in \Omega_C \quad \forall t.$$

Any update that would violate Core invariants is projected back into Ω_C. This ensures that regeneration modifies tendencies, not structural identity.

A.4 Homeostatic Constraints

To prevent runaway amplification or identity collapse, MOOL enforces explicit homeostatic conditions:

1. **Bounded Update Magnitude**
$$\|\Delta \mathbf{r}(t)\| \leq \epsilon$$
2. **Multi-Timescale Decay**
   - short-term residue decays rapidly,
   - long-term tendencies emerge only through repeated reinforcement.
3. **Capability Preservation**
   Residue updates are prohibited from interacting with:
   - base model parameters,
   - reasoning competence,
   - task-level abilities.

Identity evolution therefore remains orthogonal to capability learning [4].

A.5 Identity Evolution Without Storage

Under this formulation, identity persistence is defined as:

$$\text{Identity Continuity} \equiv \text{Stability of Reconstruction Dynamics}$$

rather than availability of stored records. Two reconstructions may differ in surface realization while remaining identity-consistent, as long as they are generated under the same Core constraints and biased by accumulated residue. This explains how identity can:

- evolve without memory growth,
- adapt without parameter updates,
- remain stable without replay.

A.6 Summary

MOOL formalizes dreaming as non-recorded regeneration: a process that transforms identity bias without storing experience. By treating residue as the sole carrier of change, the Digital Hippocampus enables long-term identity continuity while remaining bounded, auditable, and scalable.

B Evaluation Metrics for Identity Continuity

B.1 Motivation

Conventional benchmarks for Large Language Models focus on task accuracy, factual recall, or reasoning performance. However, none capture identity continuity over extended interaction horizons. In regenerative identity architectures, success is not defined by remembering more facts, but by maintaining structurally coherent identity tendencies under continuous reconstruction.

B.2 Identity Representation Assumptions

We assume identity at time t is not represented as stored content, but as latent reconstruction bias:

$$I(t) = C + \sum_{k=1}^{K} \lambda_k R_k(t),$$

where:

- C: Core identity constraints (time-invariant),
- $R_k(t)$: residual bias components accumulated through regeneration,
- $\lambda_k \in (0, 1]$: decay or weighting coefficients,
- K: number of active residue dimensions.

This formulation explicitly excludes episodic records or interaction logs.

B.3 Long-term Identity Consistency Score (LICS)

We define LICS as a measure of how well an agent preserves identity structure over time and context.

B.3.1 Core Consistency Component

$$\mathrm{LICS}_{\mathrm{core}}(t) = \cos(C_0, C_t),$$

where C_0 is the initial Core constraint vector.

B.3.2 Residue Coherence Component

$$\mathrm{LICS}_{\mathrm{res}}(t) = \cos\big(R(t - \Delta t), R(t)\big).$$

High values indicate gradual evolution; low values indicate drift.

B.3.3 Cross-Context Stability Component

$$\mathrm{LICS}_{\mathrm{ctx}}(t) = \frac{1}{|F|} \sum_{f \in F} \cos\big(I(t), I_f(t)\big).$$

This penalizes role-fragmented identity.

B.4 Composite LICS Definition

$$\mathrm{LICS}(t) = \alpha\, \mathrm{LICS}_{\mathrm{core}}(t) + \beta\, \mathrm{LICS}_{\mathrm{res}}(t) + \gamma\, \mathrm{LICS}_{\mathrm{ctx}}(t),$$

where $\alpha + \beta + \gamma = 1$.

B.5 Evaluation Protocol (Conceptual)

1. Initialize Core constraints C_0
2. Run long-horizon interaction (10k–100k turns)
3. Periodically sample identity projections
4. Compute LICS without accessing interaction logs
5. Compare against:
   - static persona agents,
   - memory-augmented agents,
   - continual fine-tuning baselines.

B.6 Significance

LICS enables identity architectures to be evaluated as dynamical systems, not memory systems. It makes regenerative identity measurable, falsifiable, and comparable.

C Interface Between COOL and MOOL

C.1 Motivation

Identity continuity must evolve without storing episodic interaction data. This places strict constraints on the COOL–MOOL interface.

C.2 Design Constraints

1. No episodic content transfer
2. Bounded dimensionality
3. Identity-oriented, not knowledge-oriented
4. One-way temporal semantics (COOL $\rightarrow$ MOOL)

C.3 COOL $\rightarrow$ MOOL: Identity Signal Extraction

Signal vector:

$$S(t) \in \mathbb{R}^d.$$

Composed of aggregated statistics such as:

- constraint adherence,
- reconstruction stability,
- interaction dynamics,
- frame transition patterns.

No semantic meaning is encoded.

C.4 Explicit Exclusions

- embeddings,
- topic vectors,
- named entities,
- user identifiers.

C.5 Residue Update Without Content

$$R(t+1) = \mathcal{H}\Big(R(t), \bar{S}\Big)\,.$$

Residue encodes tendency, not memory.

C.6 Privacy and Safety Implications

- No content stored
- No replay possible
- Privacy preserved by architecture

C.7 Comparison to Memory-Based Interfaces

- **Interface data:** Memory-centric logs/embeddings vs. Digital Hippocampus meta-signals
- **Storage growth:** Unbounded vs. constant
- **Identity continuity:** Episodic vs. regenerative
- **Replay possible:** Yes vs. no

C.8 Summary

The COOL–MOOL interface is a control-theoretic coupling, not a memory channel. Identity persists as process, not data.

D Implementation Roadmap — Residue-to-LLM Coupling

D.1 Design Philosophy

The implementation goal of the Digital Hippocampus is identity modulation without learning, storage, or replay. Accordingly, MOOL does not update model parameters, does not write to memory, and does not retain representations. Instead, residue acts as a lightweight control signal that biases how generation unfolds. To preserve model-agnosticism and future extensibility, we define an abstract intervention interface, followed by multiple concrete realization examples.

D.2 Abstract Residue Injection Interface

Let the base LLM be denoted as a conditional generator:

$$P(y_t \mid y_{<t}, x, \theta),$$

where θ are frozen model parameters. We introduce a residue-conditioned generation process:

$$P(y_t \mid y_{<t}, x, \theta, \mathbf{r}(t)).$$

Residue $\mathbf{r}(t)$ never alters θ. Instead, it modifies generation-time control surfaces, defined abstractly as:

$$\mathcal{I}_{\mathbf{r}} : \mathcal{G} \to \mathcal{G}',$$

where $\mathcal{G}$ is the original generation dynamics and $\mathcal{I}_{\mathbf{r}}$ is a bounded, reversible intervention.

Key properties of the interface:

- Stateless with respect to episodic history
- Bounded influence magnitude
- Fully removable without retraining
- Orthogonal to task competence

D.3 Realization Option 1: Logit Bias Modulation

Residue may map to additive or multiplicative bias over token logits:

$$\tilde{z}_t = z_t + B(\mathbf{r}(t)),$$

where:

- z_t are raw logits
- $B(\cdot)$ is a low-rank bias projection

Typical bias axes:

- verbosity vs. conciseness
- abstraction preference
- hedging vs. assertiveness

This approach:

- requires no architectural change
- is compatible with existing inference stacks (e.g. Transformer decoding [7])
- offers fine-grained, interpretable control

D.4 Realization Option 2: Soft Prompt / Prefix Conditioning

Residue may parameterize a small, non-trainable prefix vector:

$$\text{Prefix}(\mathbf{r}(t)) \rightarrow \text{Context}_{\text{soft}}.$$

Properties:

- prefix parameters are regenerated each session
- no prefix accumulation or learning
- residue controls style and framing, not content

This preserves identity continuity without introducing prompt memory.

D.5 Realization Option 3: Attention Modulation

Residue may bias attention distributions indirectly:

$$\text{Attn}(Q, K, V) \rightarrow \text{Attn}(Q,\ K + \Delta K(\mathbf{r}),\ V)\,,$$

or via:

- attention temperature adjustment
- key/value scaling
- layer-selective modulation

This enables:

- stable identity tendencies across long contexts
- frame-consistent emphasis patterns

D.6 Why Multiple Realizations Are Intentional

The Digital Hippocampus does not prescribe a single implementation. Instead, it defines:

- what must be controlled (identity bias)
- what must never happen (learning, storage, replay)

This allows:

- deployment across proprietary and open models
- compatibility with evolving LLM architectures
- experimentation without architectural lock-in

D.7 Safety and Auditability

Because residue:

- is low-dimensional
- contains no semantic content
- cannot reconstruct interaction history

the system remains:

- privacy-preserving by construction
- auditable via residue trajectory inspection
- resistant to hidden memory leakage

D.8 Summary

This roadmap demonstrates that regenerative identity:

- requires no continual learning
- requires no memory databases
- can be implemented as generation-time control

Residue acts as a bias carrier, not a memory store. Identity persists as process, not data.

E End-to-End Pseudocode for Digital Hippocampus (COOL + MOOL, Non-Recorded Regenerative Identity)

E.1 Data Structures

- **Core C:** Immutable identity constraints (values, tone bounds, safety invariants)
- **Frame F_t:** Situational identity reconstruction selected per turn
- **Residue $\mathbf{r}_t \in \mathbb{R}^k$:** Low-dimensional, bounded identity bias vector (no semantic content)
- **Eval Memory $E_{\mathbf{mem}}$:** Short-horizon scalar statistics only (no text, no embeddings, no recallable content)
- **Meta Signal S_t:** Scalar-only feedback extracted from Eval

E.2 Online Turn Loop (COOL)

```
function ONLINE_TURN(x_t, task_context, C, r_t, E_mem):

    # 1. Frame selection (no history content)
    F_t = SELECT_FRAME(x_t, task_context)

    # 2. Build control context (identity bias only)
```

```
    control_ctx = BUILD_CONTROL_CONTEXT(
        Core = C,
        Frame = F_t,
        Residue = r_t
    )

    # 3. Generate candidate response
    y_candidate = LLM_GENERATE(control_ctx, x_t)

    # 4. Local evaluation (no memory access)
    eval_out = EVAL(
        output = y_candidate,
        Core = C,
        Frame = F_t,
        EvalMemory = E_mem
    )

    # 5. Optional revision loop
    if eval_out.accept == false:
        y_candidate = REVISE(y_candidate, eval_out)

    # 6. Update short-term eval memory (scalars only)
    E_mem = UPDATE_EVAL_MEMORY(E_mem, eval_out.stats)

    # 7. Build meta-signal for MOOL
    S_t = BUILD_META_SIGNAL(eval_out.stats)

    return y_candidate, S_t, E_mem
```

E.3 Offline Regeneration (MOOL)

```
function OFFLINE_MOOL_UPDATE(r_t, S_window, C):

    # 1. Aggregate meta-signals (no content)
    S_bar = AGGREGATE(S_window)
    # scalars only, bounded

    # 2. Map feedback to directional delta
    delta = G(S_bar)
    # bounded, low-rank mapping

    # 3. Inject bounded stochasticity ("wall-bouncing")
    noise = SAMPLE_BOUNDED_NOISE()

    # 4. Residue update (no learning, no storage)
    r_next =  * r_t +  * delta + noise

    # 5. Multi-timescale decay
```

```
    r_next = APPLY_MULTI_TIMESCALE_DECAY(r_next)

    # 6. Drift suppression if instability detected
    r_next = DRIFT_DAMPING_IF_NEEDED(r_next, S_bar)

    # 7. Project back into Core-feasible region
    r_next = PROJECT_TO_CORE_FEASIBLE_REGION(r_next, C)

    return r_next
```

E.4 System-Level Loop

```
initialize r_0 within Core bounds
initialize E_mem =
initialize S_window =

for each interaction turn t:

    y_t, S_t, E_mem = ONLINE_TURN(
        x_t, task_context, C, r_t, E_mem
    )

    append S_t to S_window

    if OFFLINE_TRIGGER_CONDITION():
        r_t = OFFLINE_MOOL_UPDATE(r_t, S_window, C)
        reset S_window
```

E.5 Design Guarantees

- No episodic memory storage
- No replay or retrieval
- No parameter updates
- Identity persists as a regulated dynamical process
- Residue is non-semantic, bounded, and auditable
- Removal of MOOL restores baseline LLM behavior

E.6 Author Statement

This work was developed with the assistance of large language models as a writing and reasoning support tool. All conceptual design, architectural ideas, and final decisions were made by the author.

References

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 2020.

[2] Donald O. Hebb. *The Organization of Behavior*. Wiley, 1949.

[3] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 2020.

[4] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks. *Psychology of Learning and Motivation*, 1989.

[5] Daniel L. Schacter and Donna Rose Addis. Constructive memory: Past and future. *Neuron*, 2012.

[6] Giulio Tononi, Melanie Boly, Marcello Massimini, and Christof Koch. Integrated information theory: From consciousness to its physical substrate. *Nature Reviews Neuroscience*, 2016.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.