Mimir: A Hybrid Architecture for a Perpetual, Adaptive Memory System

Your Name

September 13, 2025

Abstract

A perpetually learning agent requires a memory system that can gracefully handle changes in its own conceptual understanding without catastrophic forgetting. This paper details the architecture of the Mimir memory system, designed to solve the stability-plasticity dilemma. We introduce the Entity Management System (EMS), a fast, k-NN based retrieval system for all perceptual and cognitive entities. We then detail the high-priority consolidation protocol for updating these memories after a conceptual upgrade, using sparse-to-sparse mapper networks to maintain backwards-compatibility. Finally, we describe a two-tiered storage architecture that balances efficiency and fidelity by combining a lightweight semantic index with a full-fidelity raw data archive. This hybrid approach enables a memory system that is simultaneously fast, adaptive, and capable of perfect, long-term recall.

1 Introduction

As an intelligent agent learns and adapts, its internal conceptual models of the world necessarily evolve. This presents a fundamental challenge for its memory system: how can long-term memories remain accessible via similarity-based search when the very definition of "similarity" changes? A naive memory system would suffer from catastrophic forgetting, where old memories become inaccessible to new queries.

This paper details the Mimir memory architecture, a system designed to resolve this stability-plasticity dilemma. Our system is built on three core components:

1. A fast and efficient **Entity Management System (EMS)** for k-NN retrieval of all perceptual and cognitive entities.
2. A **high-priority consolidation protocol** that uses mapper networks to gracefully update memories after a conceptual upgrade, guaranteeing backwards compatibility.

3. A **two-tiered storage system** that combines a lightweight semantic index with a raw data archive to balance speed with perfect data fidelity.

Together, these components create a memory system that is not just a passive store, but an active, adaptive part of the cognitive architecture.

2 The Entity Management System (EMS)

The EMS is the system's fast, reflexive k-NN retrieval system. Its role is strictly to act as a fast similarity search index over all persistent entities, both perceptual (e.g., a specific landmark) and cognitive (e.g., a recurring thought pattern or "cognitive schema").

When any new perceptual or cognitive instance is identified by the relevant Proteus module, the EMS is automatically triggered to perform a k-NN search against its index. The retrieved memory (or a null result) is then simply added to the information stream for the next cognitive cycle. The EMS does not perform complex data fusion; its sole responsibility is fast, reliable retrieval.

The EMS operates on the sparse fuzzy membership vectors produced by the various Proteus instances. To integrate these retrieved memories into the dense conscious state vector, the sparse output of the EMS is passed through a shared **Codec Autoencoder**, which is a sparse-to-dense transformation layer detailed in our work on the Proteus Augmented framework. This preserves the search efficiency of a sparse-native index while enabling seamless integration with the dense-vector processing of the cognitive core.

3 Memory Consolidation Protocol

To prevent retrieval failure after a conceptual model is upgraded (e.g., 'Proteus-V1' is replaced by a superior 'Proteus-V2'), the system uses a high-priority consolidation protocol to update its memory index. This protocol has two primary stages.

3.1 Forward-Mapping via Sparse-to-Sparse Translators

Immediately after an upgrade is finalized, a high-priority background process iterates through all memories stored in the old format ('v1'). For each memory, a specialized **sparse-to-sparse mapper network** ('Mapper$_v1_to_v2$') *is used to compute the equivalent represen NN index. The primary goal is to make the vector space of the search index fully consistent as rapidly as possible.*

This mapper is a unique component, distinct from the sparse-to-dense autoencoders used elsewhere. It is typically a multi-layer perceptron with an output layer that uses a 'Sigmoid' activation function, ensuring the resulting mapped vector has the correct dimensionality and its components are in the valid '[0, 1]' range of a fuzzy membership vector.

3.2 Temporary Fallback for Query Mapping

For the brief duration of this consolidation process, memory retrieval is more complex. A query in the new format (‘query$_v$2‘)$isalsomappedbackwardstotheoldformat$(‘$query_v$1‘).$Thek-NNsearchisthenruninparallelagainstboththeconsolidated$(‘$v$2‘)$andunconsolidated$(‘$v$1‘)$partitionsofthei$

4 Two-Tiered Storage Architecture

The memory system uses a hybrid, two-tiered approach to data storage that balances the extreme efficiency of semantic representation with the absolute data fidelity of raw storage.

4.1 Tier 1: The Semantic Memory Index

By default, the system achieves a state of extreme data compression by treating its own perceptual analysis as the canonical record. For a given event (e.g., a video stream), it processes the data, generates a rich set of semantic vectors and instance tags, and then **discards the source pixels**. The memory of the event is not the raw data, but this rich, structured, and highly compressed semantic representation. The vast majority of the system's operations—planning, recall, and internal reasoning—are performed on this lightweight and efficient data.

4.2 Tier 2: The Raw Data Archive (The "Pixel Vault")

The system includes a policy-driven mechanism to enable full-fidelity archival for critical use cases. When this mode is active, the system saves the original, untouched pixel-based data to a separate, high-capacity storage system. The ‘instance$_t$ag‘$sgenerateddduringthesemanticanalysisserveasthecrucialbridge, actingaspointersthatlinkthe$

This two-tiered design is critical for long-term adaptation. It allows the main system to remain fast and responsive, unburdened by the storage cost of raw data, while still providing a "source of truth." This enables powerful functions like **Opportunistic Re-Perception**: when a student model is upgraded, the system can use its semantic index to identify important past events, retrieve the pristine source data from the archive, and re-process it with its new perceptual apparatus to gain deeper insights and upgrade its own memories.

5 Conclusion

We have specified the architecture for a perpetual and adaptive memory system, a critical component for any perpetually learning agent. The system is designed explicitly to resolve the stability-plasticity dilemma. The high-speed Entity Management System provides fast, reflexive recall of memories, while the high-priority consolidation protocol, using sparse-to-sparse mapper networks, ensures that the memory index remains coherent and accessible even as the agent's core conceptual models evolve. Furthermore, the two-tiered storage architecture

provides a powerful combination of efficiency and fidelity, allowing the agent to operate on a lean semantic index while retaining access to a full-fidelity archive for opportunistic re-perception and long-term improvement. This architecture provides a robust and scalable solution for memory in truly adaptive intelligent systems.