

M-Flow: Episodic Memory Architecture for AI Agents

Abstract

This document describes the M-Flow knowledge graph architecture, a system designed to give AI agents persistent, structured memory. Unlike traditional retrieval-augmented generation (RAG) systems that rely on flat vector stores, M-Flow organizes information into a hierarchical episodic memory structure that supports multi-hop reasoning, temporal awareness, and cross-document entity linking. The system processes knowledge through an Extract-Memorize-Load (EML) pipeline that transforms raw heterogeneous inputs into a queryable cognitive memory graph. This paper covers the core architecture, entity recognition and linking mechanisms, knowledge graph structure, multiple retrieval modes, episode routing, procedural memory, performance benchmarks, and integration options.

1. Introduction

Modern large language models are stateless by design. Each conversation starts from scratch, with no memory of prior interactions or accumulated knowledge. This limitation severely constrains their usefulness in enterprise applications where context builds over time across multiple documents, conversations, and data sources. Organizations need AI systems that can accumulate knowledge over weeks and months, connecting insights across thousands of documents and conversations while maintaining accuracy and relevance.

M-Flow addresses this fundamental limitation by providing an Extract-Memorize-Load pipeline that transforms raw text into a queryable cognitive memory. The system processes documents through three distinct phases: extraction of raw content into clean text chunks, memorization through entity recognition and relationship building, and loading through hybrid retrieval that combines vector similarity with graph traversal. Each phase is independently configurable, allowing organizations to tune the system for their specific use cases and performance requirements.

The key innovation of M-Flow is its episodic memory model, inspired by how human memory organizes experiences into episodes, each containing multiple thematic facets and specific factual details. This hierarchical organization enables retrieval at multiple levels of granularity, from broad topic matching to precise factual recall, while maintaining the contextual relationships that make information meaningful.

2. Architecture Overview

The M-Flow architecture centers on a three-layer episodic memory model that mirrors cognitive science models of human episodic memory. At the highest level, Episodes represent coarse-grained units of knowledge, typically corresponding to documents or conversation sessions. Each Episode captures the overall theme, temporal context, and participant information for a body of content.

Each Episode contains multiple Facets, which capture thematic subdivisions of the content. A single document about a company quarterly results might generate separate Facets for financial performance, product updates, market analysis, and strategic outlook. Facets enable the system to isolate distinct topics within a single source document, improving retrieval precision when users ask about specific aspects of a broader subject.

Facets in turn contain FacetPoints, which are the atomic units of retrievable information. FacetPoints represent individual facts, claims, observations, or data points extracted from the source material. Each FacetPoint includes the original text snippet, a semantic embedding for vector search, extracted entities and relationships, and metadata about its provenance and confidence.

This hierarchical structure enables retrieval at multiple levels of granularity. A query about a broad topic might match at the Episode level, returning a comprehensive summary. A more specific question can be answered from individual FacetPoints, providing precise factual responses with clear source attribution. The system automatically determines the appropriate retrieval granularity based on query complexity and the distribution of relevant information across the hierarchy.

3. Entity Recognition and Linking

M-Flow performs named entity recognition during the memorization phase, identifying people, organizations, locations, concepts, dates, quantities, and other domain-specific entities within the text. The NER system uses the configured LLM provider to extract entities with their types, descriptions, and relationships to other entities in the same context.

Critically, entities are linked across Facets and Episodes through a global entity resolution process. When the system encounters MIT in one document and Massachusetts Institute of Technology in another, it recognizes these as the same entity and creates a single node in the knowledge graph with edges connecting to all relevant FacetPoints. This cross-document entity linking creates a rich network of cross-references that enables multi-hop reasoning.

For example, if Document A mentions Dr. Smith works at MIT and Document B mentions MIT researchers published a breakthrough in quantum computing, M-Flow can connect these facts through the shared entity MIT. This allows the system to answer questions like What research is associated with Dr. Smith institution even though no single document contains this complete information. The graph traversal algorithms can follow entity links across multiple hops to synthesize answers that require combining information from diverse sources.

4. Knowledge Graph Structure

The knowledge graph in M-Flow consists of several node types and relationship categories. Entity nodes represent real-world objects and concepts, each with a type classification, description, and set of properties. MemoryTriplet edges capture subject-predicate-object relationships between entities, such as Einstein developed Theory of Relativity or Apple headquartered in Cupertino. These triplets form the backbone of the relational knowledge that enables graph-based reasoning.

Episode, Facet, and FacetPoint nodes form the episodic hierarchy, connected by containment edges. Cross-cutting edges connect entities to FacetPoints where they appear and connect related entities to each other. Additional metadata nodes track provenance, timestamps, and confidence scores. This metadata enables filtered retrieval and quality-aware ranking.

The graph is stored in KuzuDB, a high-performance embedded graph database optimized for analytical queries. KuzuDB provides columnar storage with vectorized execution, making it efficient for both point lookups and graph analytics. Vector embeddings for semantic search are

maintained in LanceDB, providing sub-millisecond approximate nearest neighbor search across millions of embedding vectors. Relational metadata is stored in SQLite for efficient structured queries and fast lookups.

5. Retrieval Modes

M-Flow supports multiple retrieval modes, each optimized for different query types and use cases. The Episodic retrieval mode performs structured recall over the episode hierarchy, using a combination of vector similarity and graph traversal to find relevant FacetPoints. It uses an adaptive weighting system that balances semantic similarity scores with graph-based relevance signals, including entity overlap, topic coherence, and temporal proximity.

The Procedural retrieval mode searches for step-by-step instructions and workflows stored in procedural memory paths. This mode is particularly useful for enterprise knowledge bases that contain standard operating procedures, troubleshooting guides, and process documentation.

The Triplet Completion mode finds and extends knowledge triplets matching the query pattern. Given a partial triplet the system finds all predicates and objects associated with the subject in the knowledge graph. This mode excels at factual recall and relationship exploration.

The Lexical mode provides traditional keyword-based search with BM25 scoring, suitable for queries containing specific technical terms, product names, or identifiers where semantic similarity alone might miss exact matches. A hybrid search option combines lexical and semantic signals using Reciprocal Rank Fusion for robust retrieval across diverse query types.

6. Episode Routing

A key innovation in M-Flow is the Episode Routing mechanism, which determines how new content relates to existing Episodes. When a new document or message arrives, the system analyzes its semantic content and decides whether to extend an existing Episode or create a new one. This routing decision considers topic similarity measured by embedding distance between the new content and existing Episode summaries, temporal proximity, and entity overlap.

Episode Routing ensures that related information is grouped together even when it arrives from different sources at different times. A series of emails about the same project, received over weeks, will be routed to the same Episode, creating a comprehensive knowledge bundle about that project. This temporal and thematic coherence significantly improves retrieval quality for real-world knowledge management scenarios.

7. Procedural Memory

Beyond episodic memory, M-Flow supports procedural memory extraction. The system identifies step-by-step instructions, workflows, processes, and decision trees within the ingested content and organizes them into StepSequence and StepPoint structures. A StepSequence represents a complete procedure, while StepPoints are individual steps within that procedure.

Procedural memory enables the system to answer how-to questions by retrieving relevant procedural knowledge and presenting it as an ordered sequence of steps. This capability is particularly valuable for technical documentation, employee onboarding materials, and operational runbooks where the order and completeness of steps matter for correctness.

8. Performance and Scalability

M-Flow is designed for production deployment with support for Docker containers, Kubernetes orchestration, and cloud-native architectures. The system handles concurrent ingestion and retrieval through asynchronous processing pipelines, with configurable parallelism for both LLM calls and database operations.

Benchmark results demonstrate strong performance across standard evaluation datasets. On HotPotQA with 96 hard instances for multi-hop factual reasoning, M-Flow Episodic Retriever achieves Human-like Correctness scores of 0.88 to 0.92 and DeepEval Correctness scores of 0.76 to 0.79, consistently outperforming baseline graph completion approaches. On a custom Evolving Events dataset with 100 cross-chunk fuzzy and multi-hop reasoning questions over 159 enterprise knowledge chunks, M-Flow demonstrates even stronger advantages with Human-like Correctness of 0.96 to 0.98 at top-k 10.

9. Integration Options

M-Flow provides multiple integration pathways. The Python SDK offers async-first APIs for ingestion, memorization, search, and graph management. The REST API exposes all functionality through HTTP endpoints. The CLI provides command-line access. The Model Context Protocol server enables direct integration with AI assistants. LangChain and LlamaIndex adapters provide compatibility with popular frameworks.

10. Conclusion

The M-Flow episodic memory architecture represents a significant advance in AI agent memory systems. By combining hierarchical episodic structure with cross-document entity linking and multiple retrieval modes, M-Flow enables AI agents to build and query persistent knowledge that grows more valuable over time. The system is open-source under the Apache 2.0 license.

References

- Yang et al. HotPotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. EMNLP 2018.
- Feng et al. KuzuDB: An Embedded Graph Database. CIDR 2023.
- LanceDB Team. LanceDB: Multi-modal Vector Search. 2023.
- Vaswani et al. Attention Is All You Need. NeurIPS 2017.