MEMFLY: ON-THE-FLY MEMORY OPTIMIZATION VIA INFORMATION BOTTLENECK

Zhenyuan Zhang[1*], **Xianzhang Jia**[1*], **Zhiqin Yang**[1], **Mingming Chen**[1],
Zhenbo Song[2], **Wei Xue**[1], **Sirui Han**[1†], **Yike Guo**[1†]
[1]The Hong Kong University of Science and Technology,
[2]Nanjing University of Science and Technology

ABSTRACT

Long-term memory enables large language model agents to tackle complex tasks through historical interactions. However, existing frameworks encounter a fundamental dilemma between compressing redundant information efficiently and maintaining precise retrieval for downstream tasks. To bridge this gap, we propose **MEMFLY**, a framework grounded in information bottleneck principles that facilitates on-the-fly memory evolution for LLMs. Our approach minimizes compression entropy while maximizing relevance entropy via a gradient-free optimizer, constructing a stratified memory structure for efficient storage. To fully leverage MEMFLY, we develop a hybrid retrieval mechanism that seamlessly integrates semantic, symbolic, and topological pathways, incorporating iterative refinement to handle complex multi-hop queries. Comprehensive experiments demonstrate that MEMFLY substantially outperforms state-of-the-art baselines in memory coherence, response fidelity, and accuracy.

1 INTRODUCTION

The evolution of Large Language Models (LLMs) from stateless reasoning engines to persistent autonomous agents necessitates robust long-term memory systems capable of supporting complex, extended reasoning tasks Xi et al. (2023); Wang et al. (2024); Ferrag et al. (2025). Such memory systems must address fundamental challenges: retaining entity states that evolve over time, resolving temporal dependencies across interaction sessions, and synthesizing evidence distributed across numerous conversational turns. However, existing frameworks encounter a fundamental dilemma between compressing redundant information efficiently and maintaining precise retrieval for downstream tasks.

Existing memory frameworks Shinn et al. (2023); Sumers et al. (2024); Zhang et al. (2025); Fang et al. (2025); Zhai et al. (2025) generally fall into two paradigms, neither of which adequately resolves this tension. Retrieval-centric approaches Lewis et al. (2021); Asai et al. (2023); Yan et al. (2024); Gao et al. (2024); Ram et al. (2023) preserve verbatim details but accumulate redundancy without consolidation, leading to monotonic entropy increase and elevated retrieval noise. Memory-augmented approaches Packer et al. (2024); Zhong et al. (2023); Xu et al. (2025); Wang et al. (2025) employ LLM-driven summarization for compression but sacrifice fine-grained fidelity required for precise reasoning. Both paradigms lack a unified, principled objective for determining what information to retain versus discard. This challenge is fundamentally an information-theoretic optimization problem that aligns with the Information Bottleneck (IB) principle Slonim & Tishby (1999): compress redundant observations while preserving sufficient fidelity for future tasks.

To bridge this gap, we propose MEMFLY (**Mem**ory optimization on-the-**Fly**), a framework grounded in information bottleneck principles that facilitates on-the-fly memory evolution for LLMs. Building upon the Agglomerative Information Bottleneck algorithm Slonim & Tishby (1999), MEMFLY addresses the compression-fidelity trade-off through two complementary mechanisms. To construct memory, we employ an LLM-driven gradient-free optimizer, which approximates Jensen-Shannon

[*]Equal Contribution.
[†]Corresponding Authors: siruihan@ust.hk, yikeguo@ust.hk

divergence through semantic assessment and actively merges redundant content to minimize representational complexity $I(X;M)$ during memory ingestion. Simultaneously, we maintain a stratified Note-Keyword-Topic hierarchy grounded in the double clustering principle Slonim & Tishby (2000), where Keywords serve as intermediate symbolic anchors stabilizing the semantic space between raw observations (Notes) and high-level semantic regions (Topics), thereby preserving task-relevant information $I(M;Y)$.

To leverage constructed memory, we design a hybrid retrieval mechanism that seamlessly integrates semantic, symbolic, and topological pathways: macro-semantic navigation through Topics, micro-symbolic anchoring through Keywords, and topological expansion through associative links established during consolidation. For complex queries requiring multi-hop reasoning, we further introduce an iterative refinement protocol that progressively expands the evidence pool until sufficient information is gathered. The contributions of this work are summarized as follows:

- We formalize agentic memory as an Online Information Bottleneck problem, unifying the treatment of entropy accumulation and fidelity loss within a single theoretical framework.
- We propose two mechanisms to optimize this objective: a gradient-free optimizer that extends AIB to online settings through LLM-based semantic assessment, and a Note-Keyword-Topic hierarchy grounded in double clustering that preserves evidence structure.
- We design tri-pathway retrieval with iterative refinement to exploit the optimized structure for complex reasoning tasks.
- Extensive evaluations on comprehensive benchmarks demonstrate that MEMFLY achieves substantial improvements, significantly outperforming state-of-the-art baselines.

2 RELATED WORK

2.1 RETRIEVAL-CENTRIC SYSTEMS

Retrieval-augmented generation (RAG) Lewis et al. (2021); Gao et al. (2024) has evolved from passive retrieve-then-read pipelines to active, iterative workflows. Recent advances introduce inference-time feedback loops for query refinement and hallucination filtering Asai et al. (2023); Yan et al. (2024). Structural approaches further organize knowledge into graphs, enabling both local retrieval and global summarization Edge et al. (2025); Wu et al. (2025). Beyond document-level retrieval, graph-based memory systems such as MemWalker Chen et al. (2023) maintain structured knowledge representations through explicit traversal mechanisms. Despite these sophisticated capabilities, such methods fundamentally operate as inference-time optimizations that refine the read path for specific queries while treating the underlying memory structure as a passive index. Consequently, these systems rely on query-centric embedding similarity to initiate retrieval, rendering them vulnerable to vector dilution in scenarios requiring multi-hop evidence synthesis.

2.2 MEMORY-AUGMENTED AGENTS

While retrieval-centric systems optimize retrieval, an orthogonal research direction addresses the construction path: how to structure and compress interaction history for effective long-term retention. Systems like MemGPT Packer et al. (2024) and HiAgent Hu et al. (2024) orchestrate context through tiered storage hierarchies, swapping information between active working memory and archival storage to emulate infinite retention. Parallel efforts seek to replicate biological memory processes. MemoryBank Zhong et al. (2023) incorporates the Ebbinghaus forgetting curve to modulate information decay. A-MEM Xu et al. (2025) and O-Mem Wang et al. (2025) adopt associative strategies to foster autonomous knowledge evolution, such as, Zettelkasten-style linking or user-centric profiling. These approaches effectively mitigate the Goldfish Effect, the tendency of LLMs to prioritize recent context while losing track of earlier information Hans et al. (2024), by structuring interaction history into discrete, retrievable memory units. While effective for managing token budgets, these approaches optimize for compression efficiency without a principled mechanism for preserving task-relevant information.

3 THE MEMFLY FRAMEWORK

We formulate the construction of agentic long-term memory as an Information Bottleneck (IB) optimization problem. In this framework, the memory system is not a static repository but a dynamic channel that compresses continuous input streams into a compact, relevance-maximizing representation. Figure 1 illustrates the overall architecture of MEMFLY.

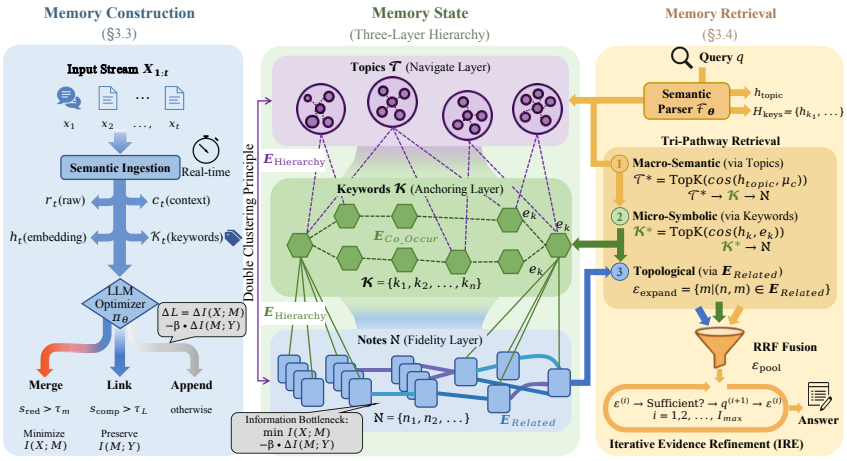

Figure 1: Overview of the MEMFLY framework. **Left**: Memory construction processes incoming observations through semantic ingestion and gated structural update, where an LLM-based optimizer performs Merge, Link, or Append operations to minimize the IB objective. **Center**: The memory state is organized as a stratified Note-Keyword-Topic hierarchy with associative edges following the double clustering principle. **Right**: Memory retrieval employs tri-pathway search via Topics, Keywords, and topological expansion, followed by iterative evidence refinement for complex queries.

3.1 PROBLEM FORMULATION

Notation. Let $X_{1:t} = \{x_1, x_2, \ldots, x_t\}$ denote a continuous stream of interaction data observed by the agent, where $x_t \in \mathcal{D}$ represents the input at time t. We define the agent's memory state at time t as a random variable M_t taking values in a structured state space. For computational realization, we instantiate M_t as a dynamic graph $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \Phi_t)$, where $\mathcal{V}_t$ is the set of memory nodes, $E_t \subseteq \mathcal{V}_t \times \mathcal{V}_t$ represents topological connections, and $\Phi_t : \mathcal{V}_t \to \mathbb{R}^d \times \Sigma^*$ maps each node to its dense embedding and textual content, with Σ^* denoting the set of all strings over alphabet Σ.

The Optimization Objective. Following the Information Bottleneck principle Slonim & Tishby (1999); Tishby & Zaslavsky (2015), our goal is to learn a memory construction policy that maps the observed interaction history $X = \{x_1, \ldots, x_t\}$ to the memory state M_t that maximizes task-relevant information while minimizing representational complexity. This is formalized as minimizing the Memory Information Bottleneck Lagrangian $\mathcal{L}_{\text{IB}}$:

$$\min_{\pi} \mathcal{L}_{\text{IB}}(M_t) = \underbrace{I(X_{1:t}; M_t)}_{\text{Compression}} - \beta \underbrace{I(M_t; Y)}_{\text{Relevance}}, \tag{1}$$

where π denotes the memory construction policy, $\beta > 0$ controls the compression-relevance trade-off, and Y represents future reasoning tasks. The Compression term $I(X_{1:t}; M_t)$ measures how much information from the raw input stream is retained in the memory state. Minimizing this term encourages the system to merge redundant information and discard irrelevant details. The Relevance term $I(M_t; Y)$ measures the mutual information between the memory state and future tasks Y. Maximizing this term ensures retention of critical evidence for downstream reasoning.

A key challenge in applying the Information Bottleneck principle to agentic memory is that future tasks Y are unknown at construction time. We define the relevance variable Y as the latent semantic structure governing future reasoning tasks. Since Y is not directly observable during memory construction, we approximate it through two proxy signals: (1) local coherence: the semantic consistency within and across memory units, captured by Keyword co-occurrence patterns; (2) global

navigability: the accessibility of evidence chains, captured by the Topic hierarchy and associative links. These proxies reflect the observation that reasoning tasks typically require either entity-centric evidence retrieval or thematic evidence aggregation. Our ablation study (Sec. 4.3) empirically validates that optimizing these structural surrogates significantly improves downstream response fidelity and accuracy.

Online Approximation via Greedy Agglomeration. Directly optimizing Eq. equation 1 over the entire history is computationally intractable due to the combinatorial explosion of possible memory configurations. Following the Agglomerative Information Bottleneck (AIB) algorithm Slonim & Tishby (1999), we adopt an online greedy strategy that makes locally optimal decisions at each time step. Specifically, we model the memory evolution as an online decision process where the state transition $M_{t+1} \leftarrow \mathcal{T}(M_t, x_t)$ is governed by a policy π. At each step, the policy seeks to minimize the incremental Lagrangian cost:

$$\Delta\mathcal{L} = \underbrace{I(X_{1:t+1}; M_{t+1}) - I(X_{1:t}; M_t)}_{\Delta I_{\text{compress}}} - \beta \underbrace{\big(I(M_{t+1}; Y) - I(M_t; Y)\big)}_{\Delta I_{\text{relevance}}}. \tag{2}$$

In the original AIB algorithm, the merge decision between clusters z_i and z_j is determined by minimizing the information loss quantified via the Jensen-Shannon divergence:

$$\delta I_Y(z_i, z_j) = \big(p(z_i) + p(z_j)\big) \cdot D_{\text{JS}}\big[p(Y|z_i), p(Y|z_j)\big]. \tag{3}$$

LLM as JS-Divergence Approximator. Computing Eq. equation 3 exactly requires access to the conditional distributions $p(Y|z_i)$ and $p(Y|z_j)$, which are unavailable since future tasks Y are unknown. We address this through a key observation: JS-divergence measures distributional similarity, which correlates with semantic similarity assessable by LLMs pre-trained on diverse tasks. Formally, we employ an LLM as a gradient-free Yang et al. (2024) policy $\pi(M_t, x_t)$ that approximates merge decisions through semantic assessment. Given two memory units n_t and n_i, the LLM evaluates their relationship and outputs scores $s_{\text{red}}(n_t, n_i)$ and $s_{\text{comp}}(n_t, n_i)$ defined in Sec. 3.3.2. We hypothesize that redundancy scores are inversely related to JS-divergence:

$$s_{\text{red}}(n_t, n_i) \approx 1 - D_{\text{JS}}\big[p(Y|n_t), p(Y|n_i)\big], \tag{4}$$

where high redundancy indicates low JS-divergence, suggesting the units would provide similar information for downstream tasks. This design choice leverages the LLM's implicit knowledge of task-relevant distributional properties acquired during pre-training, and is empirically validated in our ablation study (Sec. 4.3).

3.2 Structural Prior

To ensure the computational tractability of the online optimization, we impose a structural prior on the memory state M_t. Direct manipulation of high-dimensional embedding spaces is ill-posed due to the curse of dimensionality, which manifests as sparsity and noise in similarity structures Slonim & Tishby (2000). To mitigate these topological degradations, we draw upon the design rationale of the Double Clustering framework established by Slonim and Tishby Slonim & Tishby (2000). Their information-theoretic analysis demonstrated that for high-dimensional co-occurrence data, optimal compression is achieved not by clustering data points directly, but by first clustering the feature space to form robust intermediate representations. Specifically, the framework posits a two-stage abstraction process: words are first aggregated into "word clusters" ($Y \to \tilde{Y}$) based on their conditional distributions $p(x|y)$, yielding distributionally robust feature centroids. Subsequently, documents are clustered ($X \to \tilde{X}$) based on their distributions over these word clusters $p(\tilde{y}|x)$. This intermediate symbolic layer resolves the sparsity issue, allowing the system to achieve superior structural organization by projecting data onto a denser, less noisy representation.

Adhering to this principle, MemFly instantiates the memory state as a stratified Note-Keyword-Topic hierarchy:

Layer 1: Notes $\mathcal{N}$ (Fidelity Layer). At the atomic level, we maintain the set of Notes, $\mathcal{N} = \{n_1, \ldots, n_N\}$, serving as non-parametric memory units. Formally, each note is defined as a tuple $n_i = (r_i, c_i, \mathbf{h}_i, \mathcal{K}_i)$, where r_i denotes the raw observational data (verbatim content) and c_i represents the augmented context—a semantically denoised summary generated to enhance retrieval

relevance. To facilitate hybrid access, these textual components are mapped into dual representational spaces: a continuous dense embedding $\mathbf{h}_i \in \mathbb{R}^d$ encoding the context c_i, and a discrete set of symbolic keywords $\mathcal{K}_i \subset \mathcal{K}$ serving as topological anchors. Analogous to the input variable X in the Information Bottleneck framework, this layer is designed to preserve raw observational fidelity, mathematically approximating the condition $I(\mathcal{N}; X) \approx H(X)$. By explicitly maintaining non-parametric access to original inputs, we effectively mitigate the hallucination risks inherent in purely parametric or compression-heavy memory systems.

Layer 2: Keywords $\mathcal{K}$ (Anchoring Layer). To bridge continuous embedding spaces and discrete symbolic reasoning, we introduce Keywords $\mathcal{K} = \{k_1, \ldots, k_K\}$ as intermediate symbolic anchors. This layer serves an analogous role to word clusters ($\tilde{Y}$) in the double clustering framework. Keywords resolve semantic sparsity by grounding proximity in shared symbolic substructures rather than potentially spurious vector correlations. Each Keyword k_j maintains its own embedding $\mathbf{e}_j \in \mathbb{R}^d$ and tracks co-occurrence relationships with other Keywords extracted from the same Notes, forming the edge set $E_{\text{Co\_Occur}}$.

Layer 3: Topics $\mathcal{T}$ (Navigation Layer). At the macro level, we aggregate keywords into topics $\mathcal{T} = \{C_1, \ldots, C_T\}$ based on their co-occurrence structure, analogous to document clusters ($\tilde{X}$) in the double clustering framework. Topics serve as semantic centroids that partition the memory latent into navigable regions, enabling $O(1)$ macro-semantic localization during retrieval.

3.3 Memory Construction

To tractably minimize the Memory IB Lagrangian (Eq. equation 1), MemFly employs a computation-on-construction mechanism. We model the memory update as an online agglomerative process, comprising three stages: ingestion, gated structural update, and topic evolution.

3.3.1 Semantic Ingestion and Denoising.

Raw input streams often contain elliptical references, syntactic noise, and implicit context. We project raw input x_t into a structured Note n_t via an LLM-based transformation:

$$n_t = \mathcal{F}_{\text{ingest}}(x_t) = (r_t, c_t, \mathbf{h}_t, \mathcal{K}_t), \tag{5}$$

where r_t preserves raw content, c_t is the denoised context, $\mathbf{h}_t = \text{Embed}(c_t) \in \mathbb{R}^d$, and $\mathcal{K}_t \subseteq \mathcal{K}$ is the extracted Keyword set. This transformation enhances signal-to-noise ratio, improving $I(n_t; Y)$ relative to $I(x_t; Y)$.

3.3.2 Gated Structural Update

Before consolidation, we retrieve a candidate neighborhood $\mathcal{N}_{\text{cand}}$ by querying existing memory through dual sparse-dense indices, localizing the decision space to the most relevant subgraph. The LLM policy evaluates each candidate pair (n_t, n_i) with $n_i \in \mathcal{N}_{\text{cand}}$ by generating two scalar scores through structured prompting: a redundancy score $s_{\text{red}}(n_t, n_i) \in [0, 1]$ and a complementarity score $s_{\text{comp}}(n_t, n_i) \in [0, 1]$. Specifically, s_{red} quantifies the semantic overlap between units, where a unit value indicates identity in informational content. Conversely, s_{comp} measures the strength of logical or topical connections between nodes that possess distinct, non-overlapping information. The prompting templates are provided in Appendix.

Structural Operations. Based on these scores, the policy executes one of three operations:

$$M_{t+1} \leftarrow \mathcal{O}(n_t, n_i) = \begin{cases} \textsc{Merge}(n_i \leftarrow n_i \oplus n_t) & \text{if } s_{\text{red}}(n_t, n_i) > \tau_m \\ \textsc{Link}(n_i \leftrightarrow n_t) & \text{if } s_{\text{comp}}(n_t, n_i) > \tau_l \\ \textsc{Append}(n_t) & \text{otherwise} \end{cases}, \tag{6}$$

where τ_m and τ_l are threshold hyperparameters.

Merge Operation. When $s_{\text{red}} > \tau_m$, the content of n_t is integrated into n_i:

$$r_i' = r_i \cup r_t, \quad c_i' = \mathcal{F}_{\text{merge}}(c_i, c_t), \quad n_i \leftarrow (r_i', c_i', \text{Embed}(c_i'), \mathcal{K}_i \cup \mathcal{K}_t), \tag{7}$$

where $\mathcal{F}_{\text{merge}}$ is an LLM-based function that synthesizes a unified context preserving all distinct information from both units. This operation directly minimizes $I(X_{1:t}; M_t)$ by reducing $|\mathcal{V}_t|$, analogous to the AIB merge step that selects pairs with minimal JS-divergence.

Link Operation. When $s_{\text{comp}} > \tau_l$, a directed edge is established:

$$E_{\text{RELATED}} \leftarrow E_{\text{RELATED}} \cup \{(n_t, n_i)\}. \tag{8}$$

While Link does not directly reduce $I(X_{1:t}; M_t)$ like Merge, it preserves conditional dependencies that support $I(M_t; Y)$. Formally, Link is triggered when:

$$I(n_t; Y \mid n_i) > 0 \ \wedge \ I(n_t; n_i) > 0, \tag{9}$$

indicating that n_t provides additional task-relevant information beyond n_i, and the two are logically related. By explicitly encoding this relationship in E_{RELATED}, we preserve conditional structure necessary for multi-hop reasoning without increasing representational redundancy.

Append Operation. When neither threshold is met, n_t is appended as an autonomous unit, preserving distributional diversity for novel content. The original AIB algorithm supports only merge operations on fixed co-occurrence matrices. MEMFLY extends this framework with Link and Append operations to handle streaming settings where information arrives incrementally and may exhibit complementary or novel content. This extension maintains the greedy optimization spirit while adapting to agentic memory requirements.

3.3.3 TOPIC EVOLUTION.

Maintaining $O(1)$ macro-navigability requires periodic restructuring of the Topic layer. We formalize this as constrained graph partitioning over the Keyword co-occurrence graph $\mathcal{G}_{\text{kw}}$:

$$\max_{\mathcal{T}} \mathcal{Q}(\mathcal{T}, \mathcal{G}_{\text{kw}}) \quad \text{s.t.} \quad \delta_{\min} \leq |C_i| \leq \delta_{\max}, \ \forall C_i \in \mathcal{T}, \tag{10}$$

where $\mathcal{Q}$ denotes the modularity function and $\delta_{\min}, \delta_{\max}$ are cardinality bounds. We employ the Leiden algorithm Traag et al. (2019) for efficiency. While modularity optimization differs from direct IB clustering, empirical studies demonstrate strong correlation between modularity-based and information-theoretic community structures Fortunato (2010).

3.4 MEMORY RETRIEVAL

3.4.1 TRI-PATHWAY HYBRID RETRIEVAL.

To exploit the optimized memory structure, MEMFLY employs a tri-pathway hybrid retrieval strategy. Unlike conventional flat vector search, our approach decomposes queries into complementary semantic signals and executes parallel traversals over the memory graph. The raw query q is processed by an LLM-based semantic parser $\mathcal{F}_\theta$ to disentangle retrieval intent:

$$(\mathbf{h}_{\text{topic}}, \mathbf{H}_{\text{keys}}) \leftarrow \mathcal{F}_\theta(q), \tag{11}$$

where $\mathbf{h}_{\text{topic}} \in \mathbb{R}^d$ encodes the topical description, and $\mathbf{H}_{\text{keys}} = \{\mathbf{h}_{k_1}, \ldots, \mathbf{h}_{k_m}\}$ contains embeddings for core entities in the query. The intent signals drive three synergistic pathways: macro-semantic localization, micro-symbolic anchoring, and topological expansion.

Pathway 1: Macro-Semantic Localization. This pathway addresses the navigation challenge in large-scale memory. Given $\mathbf{h}_{\text{topic}}$, we identify the top-$K_{\text{topic}}$ relevant Topic centroids:

$$\mathcal{T}^* = \text{TopK}_{K_{\text{topic}}}\big(\cos(\mathbf{h}_{\text{topic}}, \boldsymbol{\mu}_C) \mid C \in \mathcal{T}\big), \tag{12}$$

where $\boldsymbol{\mu}_C \in \mathbb{R}^d$ is the centroid embedding of Topic C. Notes are retrieved by hierarchy traversal:

$$\mathcal{R}_{\text{topic}} = \{n \in \mathcal{N} \mid \exists k \in \mathcal{K}_n, \exists C \in \mathcal{T}^*, k \in C\}. \tag{13}$$

Pathway 2: Micro-Symbolic Anchoring. This pathway addresses the precision challenge for entity-centric queries. Query entities are matched against the keyword index:

$$\mathcal{K}^* = \bigcup_{\mathbf{h}_k \in \mathbf{H}_{\text{keys}}} \text{TopK}_{K_{key}}\big(\cos(\mathbf{h}_k, \mathbf{e}_{k'}) \mid k' \in \mathcal{K}\big), \tag{14}$$

where $\mathbf{e}_{k'} \in \mathbb{R}^d$ is the embedding of Keyword k'. Notes are retrieved via keyword membership:

$$\mathcal{R}_{\text{key}} = \{n \in \mathcal{N} \mid \mathcal{K}_n \cap \mathcal{K}^* \neq \emptyset\}. \tag{15}$$

Table 1: Main results on LoCoMo benchmark using closed-source models (GPT series). We report F1 and BLEU-1 (%) scores across five categories. The best performance in each category is marked in **bold**, and the second best is underlined.

Model	Method	Category										Average	
		Multi Hop		Temporal		Open Domain		Single Hop		Adversial		F1	BLEU
		F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU		
4o-mini	LoCoMo	25.02	19.75	18.41	14.77	12.04	11.16	40.36	29.05	**69.23**	**68.75**	39.74	33.47
	ReadAgent	9.15	6.48	12.60	8.87	5.31	5.12	9.67	7.66	9.81	9.02	9.89	7.87
	MemoryBank	5.00	4.77	9.68	6.99	5.56	5.94	6.61	5.16	7.36	6.48	6.99	5.73
	MemGPT	26.65	17.72	25.52	19.44	9.15	7.44	41.04	34.34	43.29	42.73	35.45	30.16
	A-mem	27.02	20.09	45.85	**36.67**	12.14	12.00	44.65	37.06	50.03	49.47	41.97	36.16
	Mem-0	**34.72**	**25.13**	45.93	35.51	22.64	15.58	43.65	37.42	30.15	27.44	38.70	32.07
	MemFly	32.11	24.48	**46.61**	31.84	**23.98**	**16.84**	**44.74**	**38.17**	51.48	51.96	**43.76**	**37.27**
4o	LoCoMo	28.00	18.47	9.09	5.78	16.47	14.80	**61.56**	**54.19**	**52.61**	**51.13**	44.12	**38.70**
	ReadAgent	14.61	9.95	4.16	3.19	8.84	8.37	12.46	10.29	6.81	6.13	9.98	8.07
	MemoryBank	6.49	4.69	2.47	2.43	6.43	5.30	8.28	7.10	4.42	3.67	6.13	5.15
	MemGPT	30.36	22.83	17.29	13.18	12.24	11.87	60.16	53.35	34.96	34.25	41.02	36.23
	A-mem	32.86	23.76	39.41	31.23	17.10	15.84	48.43	42.97	36.35	35.53	40.53	35.36
	Mem-0	35.13	27.56	**52.38**	**44.15**	17.73	15.92	39.12	35.43	25.44	24.19	36.59	32.25
	MemFly	**35.89**	**29.24**	39.78	27.12	**25.74**	**19.53**	49.08	43.05	48.24	48.92	**44.39**	**38.70**

Pathway 3: Topological Expansion. This pathway addresses connectivity for multi-hop reasoning by retrieving evidence that is logically related but vectorially distant. Starting from the anchor set:

$$\mathcal{E}_{\text{anc}} = \mathcal{R}_{\text{topic}} \cup \mathcal{R}_{\text{key}}, \tag{16}$$

we expand along the E_{RELATED} edges established during consolidation:

$$\mathcal{E}_{\text{expand}} = \{m \in \mathcal{N} \mid \exists n \in \mathcal{E}_{\text{anc}}, (n, m) \in E_{\text{RELATED}}\}. \tag{17}$$

Evidence Fusion. The final evidence pool combines all pathways via Reciprocal Rank Fusion (RRF) Cormack et al. (2009). RRF aggregates the reciprocal ranks of candidates across different retrieval pathways, prioritizing evidence that consistently appears at the top of multiple lists without requiring score normalization. The final pool is:

$$\mathcal{E}_{\text{pool}} = \text{Top-}K_{\text{final}}\big(\text{score}_{\text{RRF}} \cup \mathcal{E}_{\text{expand}}\big), \tag{18}$$

where $\text{score}_{\text{RRF}}$ denotes the fusion score calculated by RRF and K_{final} denotes the predefined budget for the final pool.

3.4.2 Iterative Evidence Refinement

Complex reasoning tasks may require evidence not directly accessible from the initial query. We address this through an Iterative Evidence Refinement (IER) protocol that progressively expands the evidence pool. At each iteration i, the system evaluates whether the current evidence pool $\mathcal{E}^{(i)}$ sufficiently addresses the query. This evaluation is performed by an LLM that assesses information completeness. Formally, we define the sufficiency predicate:

$$\text{Suf}(\mathcal{E}^{(i)}, q) = \begin{cases} 1, & \text{if LLM}(\mathcal{E}^{(i)}, q) = \text{true} \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

If gaps are identified, a refined sub-query $q^{(i+1)}$ is synthesized to target missing aspects, and retrieval is re-executed via the tri-pathway mechanism. The evidence pool is updated:

$$\mathcal{E}^{(i+1)} = \mathcal{E}^{(i)} \cup \{n \in \mathcal{R}(q^{(i+1)}) \mid n \notin \mathcal{E}^{(i)}\}, \tag{20}$$

where $\mathcal{R}(q)$ denotes the tri-pathway retrieval function. This process continues until $\text{Suf}(\mathcal{E}^{(i)}, q) = \texttt{true}$ or the maximum iteration count $I_{\max}$ is reached.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate MemFly on the LoCoMo benchmark Maharana et al. (2024), a dataset specifically designed to assess the long-term information synthesis capabilities of LLM agents. LoCoMo contains long-horizon conversations with interleaved topics and evolving entity states, making it a robust testbed for dynamic memory structures. To provide a granular analysis of memory

Table 2: Main results on LoCoMo benchmark using open-source models (Qwen series). We report F1 and BLEU-1 (%) scores across five reasoning categories. The best performance in each category is marked in **bold**, and the second best is underlined.

Model	Method	Category										Average	
		Multi Hop		Temporal		Open Domain		Single Hop		Adversial		F1	BLEU
		F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU		
Qwen3-8B	LoCoMo	25.09	15.73	32.82	27.14	14.47	13.35	20.18	18.39	**46.77**	40.81	28.62	24.22
	ReadAgent	13.17	9.30	34.91	27.04	8.80	7.45	26.44	24.83	29.98	28.34	25.87	22.93
	MemoryBank	21.25	14.53	30.20	21.11	11.33	10.53	32.75	26.33	30.95	30.13	29.27	23.90
	MemGPT	22.13	13.44	31.47	22.16	14.51	13.54	33.49	34.12	34.58	31.44	30.88	27.67
	A-mem	24.30	16.90	34.50	23.10	13.10	12.20	38.10	33.30	31.00	30.10	32.76	27.58
	Mem-0	23.04	19.74	29.65	23.16	**20.63**	13.75	30.46	25.62	26.02	22.48	27.80	23.11
	MemFly	**28.24**	**22.76**	**38.39**	**33.64**	15.43	**13.81**	**42.09**	**36.57**	43.79	**43.14**	**38.62**	**34.51**
Qwen3-14B	LoCoMo	**33.37**	**24.26**	**31.49**	16.42	13.92	11.02	25.46	24.82	**49.17**	**35.00**	32.42	25.02
	ReadAgent	13.16	9.61	18.12	12.33	12.16	9.25	32.83	28.35	5.96	4.2	20.63	16.75
	MemoryBank	25.97	18.16	25.37	18.76	13.52	11.69	34.92	30.6	21.94	17.56	28.16	23.08
	MemGPT	24.12	15.41	25.48	19.04	13.44	12.64	34.74	32.41	27.11	24.32	28.99	25.06
	A-mem	21.36	14.98	23.06	18.04	12.62	11.49	35.43	30.92	26.71	25.78	28.37	24.48
	Mem-0	20.98	16.27	31.5	21.73	12.7	**13.22**	24.7	19.14	21.01	19.84	23.86	19.02
	MemFly	30.80	23.13	29.25	**24.56**	**14.11**	11.03	**42.25**	**35.52**	26.59	25.02	**33.65**	**28.45**

performance, we evaluate on five distinct reasoning categories: Multi-Hop, Temporal, Open Domain, Single Hop, and Adversarial.

Evaluation Metrics. Following standard evaluation metrics Xu et al. (2025), we employ two primary metrics: F1 Score to measure the token-level overlap and precision of the answer spans, and BLEU-1 Papineni et al. (2002) to evaluate the lexical fidelity of the generated responses against ground truth. For ablation studies, we additionally report Recall, measuring the proportion of ground-truth evidence retrieved, and Hit Rate, indicating whether any relevant evidence appears in the candidates.

Implementation Details. We implement MemFly using a triple-layer graph architecture backed by Neo4j, integrating both vector indices and explicit topological relationships. For retrieval, we set $K_{\text{topic}} = 3$ for Topic-based navigation, $K_{\text{key}} = 10$ for Keyword anchoring, $K_{\text{final}} = 20$ for the final retrieval pool size, and perform 1-hop traversal along E_{RELATED} edges for topological expansion. The iterative refinement protocol uses $I_{\max} = 3$ iterations. For memory construction, we set the merge threshold $\tau_m = 0.7$ and link threshold $\tau_l = 0.5$ based on validation performance.

Backbone Models and Baselines. We evaluate MemFly across four foundation models spanning closed-source (GPT OpenAI (2024)) and open-source (Qwen Qwen (2025)) families: GPT-4o-mini, GPT-4o, Qwen3-8B, and Qwen3-14B. The generation temperature is set to 0.7 for general reasoning and 0.5 for adversarial tasks. We compare MemFly against six representative methods: LoCoMo Maharana et al. (2024), ReadAgent Lee et al. (2024), MemoryBank Zhong et al. (2023), MemGPT Packer et al. (2024), A-MEM Xu et al. (2025), and Mem0 Chhikara et al. (2025). All baselines are implemented using their official system prompts and default configurations to ensure a fair comparison.

4.2 Main Results

Overall Performance. Tables 1 and 2 present performance comparisons on closed-source and open-source models, respectively. MemFly achieves the highest average F1 and BLEU-1 scores across all four backbone models. On closed-source models, it attains 43.76% and 44.39% F1 on GPT-4o-mini and GPT-4o respectively, outperforming the strongest baseline by 1.79 and 0.27 points. The advantage becomes more pronounced on open-source models: on Qwen3-8B, MemFly achieves 38.62% F1, surpassing the second-best A-mem by 5.86 points. This larger margin on open-source models suggests that our structured memory organization effectively compensates for weaker in-context reasoning capabilities. The consistent improvements across heterogeneous architectures validate the generalization of our approach.

Category-wise Analysis. Among the five reasoning categories, MemFly demonstrates the largest gains on Open Domain queries, achieving 25.74% F1 on GPT-4o compared to 17.73% for Mem-0. This improvement can be attributed to Topic-based navigation that localizes relevant memory regions before fine-grained retrieval. For Single Hop tasks requiring precise entity matching, MemFly achieves top performance on both Qwen models (42.09% and 42.25% F1), indicating effective Keyword-based anchoring.

Table 3: Ablation study on LoCoMo (Qwen3-8B). We evaluate memory construction and retrieval components. Average F1, BLEU-1, Recall, and Hit Rate (%) are reported. The best performance in each category is marked in **bold**, and the second best is underlined.

Phase	Method	F1	BLEU	Recall	Hit Rate
-	MEMFLY	**38.62**	**36.85**	**62.22**	**67.12**
Construction	w/o Update	27.97	27.10	42.11	48.20
	w/o Denoise	36.07	34.68	57.42	62.55
	w/o Link	33.57	32.35	53.19	56.18
	w/o Merge	34.79	33.62	54.85	59.42
Retrieval	w/o Topic	36.79	34.66	53.30	58.91
	w/o Keyword	32.69	33.94	51.28	54.26
	w/o Neighbor	34.26	32.85	51.28	54.35
	w/o IER	32.94	30.86	46.29	51.26

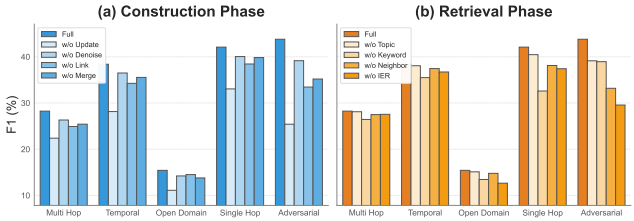

Figure 2: Category-wise F1 scores (%) for ablation variants on LoCoMo (Qwen3-8B). (a) Ablations on memory construction components. (b) Ablations on retrieval pathways and iterative refinement.

4.3 Ablation Study

Memory Construction Ablation. We examine the impact of IB-based memory consolidation by disabling core construction mechanisms. Removing the entire gated update (*w/o Update*) causes the most severe degradation, with average F1 dropping from 38.62% to 27.97% and Recall declining from 62.22% to 42.11%. As shown in Figure 2(a), this variant exhibits the largest performance gap across all five categories, with Adversarial and Temporal showing the most pronounced decline. This confirms that without active consolidation, noise accumulates and temporal dependencies become disrupted. Among individual operations, *w/o Link* shows larger impact than *w/o Merge* (33.57% vs 34.79% F1), and Figure 2(a) reveals that Link removal particularly affects Adversarial performance, indicating that associative edges are critical for filtering distractors. The *w/o Denoise* variant achieves the second-best performance (36.07% F1), maintaining relatively stable results across all categories as shown in the figure, suggesting that semantic preprocessing provides consistent but auxiliary improvements.

Memory Retrieval Ablation. Ablation results in Figure 2(b) reveal distinct roles for each pathway: *w/o Topic* (36.79% F1) shows general degradation, while *w/o Keyword* (32.69% F1) and *w/o IER* (32.94% F1) impact specific categories most. Specifically, Keyword removal significantly degrades Single-Hop performance, underscoring its role in precise entity matching. Meanwhile, *w/o IER* and *w/o Neighbor* (34.26% F1) primarily affect Adversarial and Open Domain queries, validating the necessity of iterative refinement and topological expansion for complex, multi-hop reasoning.

5 Conclusion

We presented MEMFLY, a framework that formulates agentic long-term memory as an Information Bottleneck problem. Our approach employs an LLM-based gradient-free optimizer to consolidate redundant information while preserving task-relevant evidence through a stratified Note-Keyword-Topic hierarchy. The tri-pathway retrieval mechanism with iterative refinement effectively exploits this structure for complex reasoning. Experiments on LoCoMo demonstrate consistent improvements over state-of-the-art baselines across diverse backbone models.

Limitations. The current implementation prioritizes memory quality over construction speed, introducing moderate computational overhead. Extending evaluation to multi-modal and domain-specific scenarios remains an avenue for future investigation.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. Walking down the memory maze: Beyond context limit through interactive reading. *arXiv preprint arXiv:2310.05029*, 2023.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory, 2025.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2009.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph rag approach to query-focused summarization, 2025.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. Lightmem: Lightweight and efficient memory-augmented generation, 2025. URL https://arxiv.org/abs/2510.18866.

Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. From llm reasoning to autonomous ai agents: A comprehensive review, 2025. URL https://arxiv.org/abs/2504.19678.

Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75–174, February 2010. ISSN 0370-1573. doi: 10.1016/j.physrep.2009.11.002. URL http://dx.doi.org/10.1016/j.physrep.2009.11.002.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Abhimanyu Hans, Yuxin Wen, Neel Jain, John Kirchenbauer, Hamid Kazemi, Prajwal Singhania, Siddharth Singh, Gowthami Somepalli, Jonas Geiping, Abhinav Bhatele, and Tom Goldstein. Be like a goldfish, don't memorize! mitigating memorization in generative llms, 2024. URL https://arxiv.org/abs/2406.10209.

Mengkang Hu, Tianxing Chen, Qiguang Chen, Yao Mu, Wenqi Shao, and Ping Luo. Hiagent: Hierarchical working memory management for solving long-horizon agent tasks with large language model, 2024.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents, 2024.

OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, 2002. URL https://api.semanticscholar.org/CorpusID:11080756.

Qwen. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models, 2023. URL https://arxiv.org/abs/2302.00083.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Noam Slonim and Naftali Tishby. Agglomerative information bottleneck. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

Noam Slonim and Naftali Tishby. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, pp. 208–215, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 978-1-58113-226-7. doi: 10.1145/345508.345578.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. Cognitive architectures for language agents, 2024. URL https://arxiv.org/abs/2309.02427.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle, 2015. URL https://arxiv.org/abs/1503.02406.

V. A. Traag, L. Waltman, and N. J. van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1), March 2019. ISSN 2045-2322. doi: 10.1038/s41598-019-41695-z. URL http://dx.doi.org/10.1038/s41598-019-41695-z.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), March 2024. ISSN 2095-2236. doi: 10.1007/s11704-024-40231-1. URL http://dx.doi.org/10.1007/s11704-024-40231-1.

Piaohong Wang, Motong Tian, Jiaxian Li, Yuan Liang, Yuqing Wang, Qianben Chen, Tiannan Wang, Zhicong Lu, Jiawei Ma, Yuchen Eleanor Jiang, and Wangchunshu Zhou. O-mem: Omni memory system for personalized, long horizon, self-evolving agents, 2025.

Xiaojun Wu, Cehao Yang, Xueyuan Lin, Chengjin Xu, Xuhui Jiang, Yuanliang Sun, Hui Xiong, Jia Li, and Jian Guo. Think-on-graph 3.0: Efficient and adaptive llm reasoning on heterogeneous graphs via multi-agent dual-evolving context retrieval, 2025.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Qin Liu, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, Zhangyue Yin, Shihan Dou, Rongxiang Weng, Wensen Cheng, Qi Zhang, Wenjuan Qin, Yongyan Zheng, Xipeng Qiu, Xuanjing Huan, and Tao Gui. The rise and potential of large language model based agents: A survey. *ArXiv*, abs/2309.07864, 2023. URL https://api.semanticscholar.org/CorpusID:261817592.

Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. A-mem: Agentic memory for llm agents, 2025.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation, 2024.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers, 2024. URL https://arxiv.org/abs/2309.03409.

Yunpeng Zhai, Shuchang Tao, Cheng Chen, Anni Zou, Ziqian Chen, Qingxu Fu, Shinji Mai, Li Yu, Jiaji Deng, Zouying Cao, Zhaoyang Liu, Bolin Ding, and Jingren Zhou. Agentevolver: Towards efficient self-evolving agent system, 2025. URL https://arxiv.org/abs/2511.10395.

Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. G-memory: Tracing hierarchical memory for multi-agent systems, 2025. URL https://arxiv.org/abs/2506.07398.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory, 2023.

A PROMPTING TEMPLATES

This appendix provides the complete prompting templates used in MEMFLY. All prompts are designed to elicit structured JSON outputs for reliable parsing.

A.1 MEMORY CONSTRUCTION PROMPTS

A.1.1 SEMANTIC INGESTION PROMPT

During the ingestion phase (Sec. 3.3.1), raw conversational input x_t is transformed into a structured Note $n_t = (r_t, c_t, h_t, K_t)$. The following prompt instructs the LLM to extract keywords K_t and generate the denoised context c_t:

Semantic Ingestion Prompt

You are an expert Knowledge Graph Extractor. Your task is to analyze the [TARGET TURN] to extract structured metadata.
Input Text: {content}
Guidelines:
1. Keywords (Entities):
- **GOAL**: Extract 3-5 specific Noun Phrases explicitly present in the text.
- **FOCUS PRIORITIES**:
  1. Proper Nouns: People (e.g., "Melanie"), Locations, Organizations.
  2. Concrete Objects: Physical items (e.g., "painting", "plate", "contract").
  3. Specific Topics: "LGBTQ support group", "deadline".
- **CRITICAL STOP LIST**: Ignore conversational meta-roles, abstract terms, and speaker names acting purely as subjects.
- **ZERO-SHOT RULE**: If the text is purely phatic (e.g., "Wow", "That's cool"), return an empty list.

2. Context (Factual Restatement):
- **GOAL**: Rewrite the text into a self-contained factual statement.
- **CONSTRAINT 1 (Strict Fidelity)**: Only use information present in the Input Text.
- **CONSTRAINT 2 (Safe Resolution)**: Resolve "I/my/we" using the Speaker's name if it appears in the text. For external pronouns where the antecedent is missing, keep the pronoun or use a generic term. Do not guess.
- **CONSTRAINT 3 (No Meta-Language)**: Start directly with the subject. Avoid "The speaker says...".

Output Format (JSON):

```
{"keywords": ["entity1", "entity2"], "context": "Melanie thinks the item is cool."}
```

The extracted keywords are matched against the existing Keyword index $\mathcal{K}$ to establish symbolic anchors, while the context is encoded via the embedding model to obtain $h_t = \text{Embed}(c_t)$. This dual extraction enables both symbolic and semantic access pathways during retrieval.

A.1.2 GATED STRUCTURAL UPDATE PROMPT

During the gated structural update phase (Sec. 3.3.2), the LLM policy evaluates the relationship between a new Note n_t and each candidate Note $n_i \in \mathcal{N}_{cand}$. The following prompt generates the redundancy score s_{red} and complementarity score s_{comp}, and determines the appropriate structural operation (Merge, Link, or Append).

Gated Structural Update Prompt

Role: You are a Knowledge Graph Updater. Your job is to evaluate the relationship between a NEW NODE and existing CANDIDATE NODES.

[NEW NODE]
Content: "{content}"
Context: "{context}"
Keywords: {keywords}

[CANDIDATE NODES]
{candidates_str}

Instructions:
Analyze each candidate and generate a JSON response following these rules:
1. Analyze Relationship:
- Determine `relation_type`: 'SUPPORTS', 'CONFLICTS', or 'RELATED_TO'.
- Assign `connection_strength` (0.0 – 1.0), indicating the degree of semantic overlap or logical connection.

2. Determine Operation (Based on Strength):
- **CASE A: Strength $\geq$ 0.8 (High Redundancy) $\rightarrow$ MERGE**
  - Action: Integrate details from the New Node into the candidate's context.
  - Template: "[Original Context]. Specifically, [New Node Info]..."
- **CASE B: Strength $\in$ [0.5, 0.8) (Complementary) $\rightarrow$ LINK**
  - Action: Establish associative edge; keep contexts separate.

  - Template: "[Original Context]. (Related: [New Node Keyword])"
- **CASE C: Strength $<$ 0.5 (Distinct) $\rightarrow$ APPEND**
  - Action: Add New Node as autonomous unit; no modification.
- **CASE D: relation_type is 'CONFLICTS' $\rightarrow$ LINK with Contrast**
  - Action: Note the conflicting information explicitly.
  - Template: "[Original Context]. However, [New Node] indicates that..."

3. Output: Return strictly valid JSON matching the schema.

Mapping to Paper Notation. The `connection_strength` score directly corresponds to our redundancy score $s_{red}(n_t, n_i)$ when the relation type indicates semantic overlap, and to the complementarity score $s_{comp}(n_t, n_i)$ when the nodes contain distinct but logically related information. The threshold $\tau_m = 0.7$ for Merge and $\tau_l = 0.5$ for Link (Sec. 4) are applied to these scores to determine the final structural operation according to Eq. equation 6.

When the Merge operation is triggered, the LLM generates a unified context $c'_i = F_{merge}(c_i, c_t)$ following the template specified in Case A, preserving all distinct information from both units while eliminating redundancy.

A.2 Memory Retrieval Prompts

A.2.1 Query Intent Analysis Prompt

During the retrieval phase (Sec. 3.4.1), the raw query q is processed by a semantic parser $\mathcal{F}_\theta$ to extract retrieval intent signals. The following prompt disentangles the query into a topical description h_{topic} and entity keywords H_{keys} for driving the tri-pathway retrieval mechanism.

Query Intent Analysis Prompt

Analyze the user query to identify its **Target Taxonomy Category**, extract key entities, and detect time-related intent.

Task 1: topic_desc (Target Category)
- Predict the **Taxonomy Category** or **Subject Heading** this query falls under.
- **Style**: Strict Noun Phrase (like a book chapter title or library category).
- **Constraint**: Keep it under 8 words.
- Do NOT describe the user's intent (e.g., avoid "how to...", "techniques for..."). Instead, name the topic itself.

Task 2: Keywords
- Extract 3-5 core entities, technical terms, or specific concepts.
- **CRITICAL**: Convert terms to their canonical singular form (e.g., "transformers" $\rightarrow$ "Transformer").
- Exclude generic verbs or stop words.

Query: {query}

Output Format (JSON):

```
{"topic_desc": "Concise Noun Phrase",
 "keywords": ["keyword1", "keyword2", "keyword3"]}
```

Mapping to Paper Notation. The `topic_desc` field is encoded via the embedding model to obtain $h_{topic} \in \mathbb{R}^d$, which drives Pathway 1 (Macro-Semantic Localization) through Topic matching (Eq. equation 12). The `keywords` are similarly embedded to form $H_{keys} = \{h_{k_1}, \ldots, h_{k_m}\}$, enabling Pathway 2 (Micro-Symbolic Anchoring) via Keyword matching (Eq. equation 14).

A.2.2 Iterative Evidence Refinement Prompts

The Iterative Evidence Refinement protocol (Sec. 3.4.2) employs two complementary prompts: a *Sufficiency Evaluator* that assesses whether the current evidence pool adequately addresses the query, and a *Sub-query Generator* that synthesizes targeted follow-up queries when gaps are identified.

Sufficiency Evaluation Prompt. At each iteration i, the following prompt evaluates whether the current evidence pool $\mathcal{E}^{(i)}$ satisfies the sufficiency predicate $\text{Suf}(\mathcal{E}^{(i)}, q)$ defined in Eq. equation 19.

Sufficiency Evaluation Prompt

You are a reflector agent that evaluates whether the current context and answer is sufficient to answer a question.

Question: {question}

Current Context and Current Answer:
{context}

Evaluate whether the provided context and answer contains enough information to answer the question comprehensively.
If the context and answer is insufficient, identify what specific information is missing.

Output Format (JSON):

```
{"sufficient": true or false,
 "missing_info": "description of missing information",
 "confidence": 0.0 to 1.0}
```

Sub-query Generation Prompt. When the sufficiency evaluation returns `sufficient: false`, the following prompt generates a refined sub-query $q^{(i+1)}$ targeting the identified information gaps.

Sub-query Generation Prompt

You are a Query Evolution Agent. Your goal is to decompose a complex user question into a specific, actionable sub-query to retrieve missing information from a Knowledge Graph.

Original Question: "{query_str}"

Current Known Information (Context):
{context_str}

History of Reasoning Steps (Q&A):
{prev_reasoning}

CRITICAL: The Reflector Agent has identified the following MISSING INFORMATION needed to answer the main question:
{missing_info}

Task: Based on the "Missing Information" and "History", formulate the **NEXT** single sub-question to retrieve this missing info.
- The sub-question must be specific.
- It should act as a search query for the next hop.
- If we have enough information to answer the main question, return 'None'.

Sub-question:

IER Protocol Flow. The two prompts work in tandem: the Sufficiency Evaluator determines whether to terminate (when `sufficient: true` or iteration count reaches I_{max}), while the Sub-query Generator drives evidence expansion by producing targeted queries that are re-executed through the tri-pathway retrieval mechanism (Eq. equation 20). The `confidence` score from the Sufficiency Evaluator can optionally be used for early termination when confidence exceeds a predefined threshold.

B Dataset Statistics

Table 4 presents the sample distribution across the five reasoning categories. The categories are designed to test distinct memory capabilities: *Multi-Hop* requires synthesizing evidence across multiple memory units; *Temporal* tests reasoning about time-dependent information and event ordering; *Open Domain* evaluates retrieval of general knowledge from conversation history; *Single Hop* assesses precise entity matching and direct fact retrieval; and *Adversarial* challenges the system with distractors and misleading information.

As shown in Table 4, the category distribution is imbalanced, with Single Hop comprising the largest proportion (42.3%) and Open Domain the smallest (4.8%). To account for this imbalance, the average scores reported in Table 1 and Table 2 are computed as **weighted averages** based on category sample sizes, ensuring that performance on larger categories contributes proportionally to the overall evaluation.

Table 4: LoCoMo benchmark category distribution.

Category	Samples	Proportion
Multi-Hop	282	14.2%
Temporal	321	16.2%
Open Domain	96	4.8%
Single Hop	841	42.3%
Adversarial	446	22.5%
Total	1,986	100%

C HYPERPARAMETER SETTINGS

Table 5 summarizes all hyperparameters used in MEMFLY. These hyperparameters are organized by the two main phases of our framework: memory construction and memory retrieval.

For memory construction, the merge threshold $\tau_m = 0.7$ and link threshold $\tau_l = 0.5$ control the gated structural update decisions (Eq. equation 6). A higher merge threshold ensures that only highly redundant information is consolidated, preserving fine-grained distinctions between memory units. The link threshold is set lower to capture complementary relationships that support multi-hop reasoning.

For memory retrieval, we set $K_{topic} = 3$ to balance navigation precision with coverage, allowing the system to explore multiple relevant Topic clusters. The keyword retrieval parameter $K_{key} = 10$ provides sufficient anchor points for entity-centric queries. The final pool size $K_{final} = 20$ bounds the evidence passed to the generation stage, balancing context richness against computational cost. The maximum IER iterations $I_{max} = 3$ prevents excessive retrieval loops while allowing sufficient evidence expansion for complex queries.

All hyperparameters were tuned on a held-out validation set. We found the framework to be relatively robust to moderate variations in these values, with performance degrading gracefully when parameters deviate within $\pm 20\%$ of the reported settings.

Table 5: Hyperparameter settings.

Phase	Parameter	Value
Construction	Merge threshold τ_m	0.7
	Link threshold τ_l	0.5
Retrieval	Topic retrieval K_{topic}	3
	Keyword retrieval K_{key}	10
	Final pool size K_{final}	20
	Max IER iterations I_{max}	3
Generation	Temperature (general)	0.7
	Temperature (adversarial)	0.5