

# Entropic Context Shaping: Information-Theoretic Filtering for Context-Aware LLM Agents

Hyunjun Kim

KAIST

hyunjun1121@kaist.ac.kr

## Abstract

Context engineering for large language model (LLM) agents requires distinguishing pragmatically useful information from misleading distractors. We introduce **Entropic Context Shaping (ECS)**, an information-theoretic framework that measures context utility via the shift in the model’s answer distribution toward the correct answer. Unlike lexical similarity methods that rely on word overlap, ECS captures *pragmatic utility*—whether a passage actually helps answer the question. We formalize utility as the signed change in answer probability and provide theoretical analysis showing that task-irrelevant updates yield near-zero distribution shift. We evaluate on multi-turn context selection tasks using LongMemEval (session-level) and LoCoMo (turn-level) benchmarks. On fine-grained turn selection, ECS with Llama-3.1-8B achieves  $F1=0.265$ , a **71.83% relative improvement** over TF-IDF ( $F1=0.154$ ), demonstrating that pragmatic utility outperforms lexical similarity when precise context selection matters. Code and data are available in the supplementary materials.

## 1 Introduction

Large language model (LLM) agents increasingly rely on dynamically constructed context windows to maintain state and inform decision-making (Yao et al., 2023; Shinn et al., 2023). As these agents interact with complex environments, they must continuously decide which observations, memories, and intermediate results to retain. This process—*context engineering*—has emerged as a critical bottleneck in agent performance.

**The Accumulation Fallacy.** Current approaches to context management typically employ semantic similarity metrics (e.g., cosine distance between embeddings) to filter redundant information (Lewis et al., 2020; Guu et al., 2020). While effective at removing near-duplicates, these methods suffer from

a fundamental flaw: they conflate *semantic novelty* with *pragmatic utility*. A piece of information may be entirely unique yet completely irrelevant to the task at hand. We term such distractors “**Red Herrings**”—facts that are true, novel, and semantically distinct, but provide zero information gain for the downstream decision.

**Motivating Example.** Consider an LLM agent solving a mathematical word problem. The following context updates arrive sequentially:

1. **Insight:** “The train’s speed is 60 km/h.” (Critical for solution)
2. **Redundant:** “The vehicle moves at sixty kilometers per hour.” (Duplicate)
3. **Red Herring:** “The conductor has worked for 15 years and enjoys jazz.” (Novel but useless)

Semantic filtering correctly rejects (2) as redundant but *accepts* (3) because it is semantically unique. This is the Accumulation Fallacy in action.

**Our Contribution: Entropic Context Shaping (ECS).** We propose an information-theoretic framework that measures the *pragmatic utility* of context updates by their impact on the model’s answer distribution. Specifically, we:

- Define utility as the shift in the model’s output distribution toward the correct answer—capturing whether a passage *helps* vs. *misleads* (Section 3)
- Provide theoretical analysis showing that task-irrelevant updates yield near-zero distribution shift under mild assumptions (Theorem 1)
- Evaluate on LongMemEval (Wu et al., 2025) and LoCoMo (Maharana et al., 2024)—multi-turn context selection benchmarks requiring fine-grained pragmatic filtering

## 2 Related Work

**Context Engineering for LLM Agents.** The challenge of managing context in LLM agents has received significant attention (Yao et al., 2023; Shinn et al., 2023; Wei et al., 2022). Most approaches treat context as a retrieval problem, selecting relevant documents or memories based on semantic similarity (Lewis et al., 2020; Karpukhin et al., 2020). While effective for information retrieval, these methods are ill-suited for agent contexts where the goal is task completion rather than document relevance. Recent work has shown that LLMs exhibit position-dependent biases when processing long contexts, often failing to utilize information in the middle of the input (Liu et al., 2024).

**Retrieval-Augmented Generation (RAG).** RAG systems (Lewis et al., 2020; Guu et al., 2020; Borgeaud et al., 2022) improve LLM performance by conditioning on retrieved documents. However, RAG assumes a static document corpus, whereas agent contexts evolve dynamically. Furthermore, RAG’s reliance on embedding similarity inherits the Accumulation Fallacy we identify. Self-RAG (Asai et al., 2024) addresses *when* to retrieve through self-reflection tokens, but does not address *what* retrieved content is pragmatically useful. Recent work on RAG robustness (Fang et al., 2024; Chen et al., 2024) addresses noise in retrieved documents through adversarial training and benchmarking, but does not resolve the fundamental semantic-pragmatic mismatch. Context compression approaches (Jiang et al., 2024) reduce token count while preserving information, but do not filter based on task utility.

**Memory Systems for Agents.** Recent work on agent memory (Park et al., 2023; Wang et al., 2024) has explored hierarchical and episodic memory structures. A comprehensive survey (Zhang et al., 2024) identifies memory as the key component transforming LLMs into “true agents.” These systems typically use recency and importance heuristics for memory consolidation. Our work complements these approaches by providing a principled utility metric for individual memory updates.

**Red Herrings and Inconsequential Noise.** The phenomenon of misleading information in reasoning tasks has been studied extensively. The Only Connect Wall (OCW) dataset (Taati et al., 2023)

demonstrated that LLMs are “fixated” by Red Herrings, failing to ignore irrelevant but salient information. Recent work on counterfactual robustness (Liu et al., 2023) shows that LLMs are susceptible to interference from unreliable external knowledge. We build on these observations by proposing an information-theoretic solution that distinguishes pragmatically useful context from semantically novel but useless distractors.

**Information-Theoretic Perspectives on LLMs.** Prior work has applied information theory to analyze LLM behavior (Xu et al., 2020; Ethayarajh et al., 2022). Our approach differs by using KL-divergence as an operational filter rather than a diagnostic tool, and by extending to multi-step trajectory divergence for improved separation.

## 3 Methodology

We formalize context filtering as an information-theoretic decision problem. Given an evolving context  $\mathcal{C}$  and a candidate update  $u$ , we seek to determine whether  $u$  provides sufficient *pragmatic utility* to warrant inclusion.

### 3.1 Problem Formulation

Let  $\mathcal{M}$  be a language model with vocabulary  $\mathcal{V}$  and let  $q$  be a query or task specification. For any context  $\mathcal{C}$ , the model induces a distribution over next-token predictions:

$$P_{\mathcal{C}}(y) = P(y \mid \mathcal{C}, q; \mathcal{M}) \quad (1)$$

**The Direction Problem.** A key insight is that measuring distribution *change* (magnitude) is insufficient—we must also assess whether the change moves the distribution toward the *correct answer*. Consider two passages for the query “Super Bowl 2021 location”:

- **Correct:** “...held in Tampa, Florida...”
- **Counterfactual:** “...held in Glendale, Arizona...” (false)

Both induce large distribution shifts (high KL-divergence), but only the first shifts *toward* the correct answer. Pure magnitude-based filtering would accept both.

### 3.2 Pragmatic Utility: Theoretical Framework

We define pragmatic utility as the **signed** change in the model’s belief about the correct answer, not merely the unsigned distribution shift:

**Definition 1** (Pragmatic Utility). The **pragmatic utility** of a candidate update  $u$  with respect to context  $\mathcal{C}$  and ground truth answer  $a^*$  is:

$$\mathcal{U}(u; \mathcal{C}, a^*) = \log P_{\mathcal{C} \cup u}(a^*) - \log P_{\mathcal{C}}(a^*) - \lambda \cdot |u| \quad (2)$$

where  $|u|$  denotes the token length of  $u$ , and  $\lambda \geq 0$  is a length penalty.

This definition captures the intuitive notion that a passage is useful if it increases the model’s probability of generating the correct answer. When ground truth is unavailable, we can approximate this using the LLM’s parametric knowledge as a proxy (Section 3.5).

**Relationship to KL-Divergence.** When the model’s prior belief is concentrated on incorrect answers, useful passages induce large KL-divergence *toward* the correct answer. The magnitude of  $D_{\text{KL}}(P_{\mathcal{C} \cup u} \| P_{\mathcal{C}})$  is a necessary but not sufficient condition for utility—we additionally require the direction to be correct.

### 3.3 Red Herring Theorem

The theoretical framework provides guarantees for *task-irrelevant* updates:

**Theorem 1** (Red Herring Rejection). *Let  $u$  be task-irrelevant (i.e.,  $u \perp q$  given  $\mathcal{C}$ ). Under regularity conditions on  $\mathcal{M}$ :*

$$D_{\text{KL}}(P_{\mathcal{C} \cup u} \| P_{\mathcal{C}}) \leq \epsilon \quad (3)$$

for small  $\epsilon > 0$  depending on model capacity.

*Proof Sketch.* Task-irrelevance implies conditional independence between  $u$  and the answer given  $\mathcal{C}$ . By the data processing inequality (Cover and Thomas, 2006), adding  $u$  cannot increase mutual information with the answer.  $\square$

**Scope and Limitations.** Theorem 1 addresses “red herrings”—semantically unique but pragmatically useless passages. It does *not* address counterfactual passages that contain plausible but incorrect information. Detecting such passages requires *direction-aware* scoring, which we operationalize in Section 3.5.

### 3.4 Multi-Token Trajectory Divergence

Single-token KL-divergence can fail when the first token is generic (e.g., “The”, “Step”). We extend to **multi-token trajectory divergence**:

**Definition 2** (Trajectory Divergence). For a generation horizon  $T$ , the trajectory divergence is:

$$\mathcal{D}_T(u; \mathcal{C}) = \sum_{t=1}^T D_{\text{KL}}(P_{\mathcal{C} \cup u}^{(t)} \| P_{\mathcal{C}}^{(t)}) \quad (4)$$

where  $P^{(t)}$  denotes the distribution at generation step  $t$ .

This cumulative measure amplifies the signal from useful passages while irrelevant passages maintain near-zero divergence across all steps.

### 3.5 Operationalizing Pragmatic Utility

Definition 1 requires access to  $P_{\mathcal{C}}(a^*)$ —the model’s probability assigned to the correct answer. We operationalize this in two settings:

**Setting 1: Ground Truth Available.** When the correct answer  $a^*$  is known (e.g., during training data curation, retrieval system evaluation, or active learning), we can directly compute Equation 2 via logprob extraction, or use an **LLM-as-Judge** (Zheng et al., 2023) that evaluates: “Does this passage help answer the question correctly?”

**Setting 2: Ground Truth Unavailable.** At inference time, when  $a^*$  is unknown, we leverage the LLM’s parametric knowledge. The **factuality-aware** scoring asks: “Based on your knowledge, does this passage contain accurate information for answering this question?” This approximates the direction-aware criterion by using the model’s prior knowledge to verify passage correctness.

**Algorithm.** Algorithm 1 summarizes our approach. In the ground-truth-available setting, the utility function directly measures answer probability change. In the ground-truth-unavailable setting, we use factuality-aware LLM scoring as a proxy.

### 3.6 Computational Efficiency via Prefix Caching

When using logprob-based scoring, we leverage *prefix caching* (Kwon et al., 2023; Xiao et al., 2024) to achieve near-constant overhead:

**Proposition 2** (Amortized Complexity). *With prefix caching, the amortized cost per candidate is  $O(|u|)$  rather than  $O(|\mathcal{C}| + |u|)$ , achieving  $O(1)$  with respect to context length.*

This is achieved by caching the key-value pairs for  $\mathcal{C}$ , such that only the incremental tokens from  $u$  require computation.

---

**Algorithm 1** Entropic Context Shaping (ECS)

---

**Require:** Context  $\mathcal{C}$ , candidate  $u$ , query  $q$ , threshold  $\tau$

**Require:** Optional: ground truth  $a^*$  (if available)

- 1: **if**  $a^*$  is available **then**
- 2:    $\mathcal{U} \leftarrow \log P_{\mathcal{C} \cup u}(a^*) - \log P_{\mathcal{C}}(a^*)$  {Answer-aware scoring}
- 3: **else**
- 4:    $\mathcal{U} \leftarrow \text{FactualityScore}(q, u; \mathcal{M})$  {Factuality-aware proxy}
- 5: **end if**
- 6: **if**  $\mathcal{U} > \tau$  **then**
- 7:   **Accept:**  $\mathcal{C} \leftarrow \mathcal{C} \cup u$
- 8: **else**
- 9:   **Reject:** discard  $u$
- 10: **end if**
- 11: **return**  $\mathcal{C}$

---

### 3.7 When Does ECS Apply?

ECS is most effective in the following scenarios:

- **Retrieval evaluation:** Comparing retrieval systems’ ability to surface useful vs. misleading passages (ground truth available from benchmark labels).
- **Training data curation:** Filtering context for instruction tuning or RLHF where correct answers are known.
- **Active learning:** Prioritizing which retrieved passages to verify when labeling resources are limited.
- **Counterfactual robustness testing:** Evaluating LLM robustness to factually incorrect context.

For real-time inference without ground truth, the factuality-aware variant provides a practical approximation, though with reduced accuracy (Section 4).

## 4 Experiments

We evaluate ECS on multi-turn context selection tasks, measuring its ability to identify pragmatically useful information in long conversational histories.

### 4.1 Experimental Setup

**Datasets** We evaluate on two datasets with different selection granularities:

- **LongMemEval (Wu et al., 2025):** Session-level selection. Given a question and

~50 conversation sessions (each ~6K tokens), select session(s) containing the answer. We use 100 samples with ground truth answer\_session\_ids.

- **LoCoMo (Maharana et al., 2024):** Turn-level evidence selection. Given a question about a long conversation (~400 turns across multiple sessions), identify specific dialogue turns serving as evidence. We evaluate on 200 QA pairs from 10 conversations.

**Models** We evaluate with two instruction-tuned LLMs of similar scale:

- **Qwen2.5-7B-Instruct (Yang et al., 2024):** 7B parameter model with 8K context
- **Llama-3.1-8B-Instruct (Grattafiori et al., 2024):** 8B parameter model with 8K context

**Baselines** We compare ECS against:

- **TF-IDF (Spärck Jones, 1972):** Cosine similarity between question and context TF-IDF vectors
- **Dense:** Sentence-BERT (Reimers and Gurevych, 2019) (all-MiniLM-L6-v2) embedding cosine similarity
- **Recency:** Prefer more recent sessions (Long-MemEval only)
- **Random:** Uniform random selection baseline

**Metric** We report F1 score for selecting  $k$  items matching the ground truth size. Since we select exactly  $k$  items where  $k$  equals ground truth size, precision equals recall equals F1.

## 4.2 Results

**Main Findings** Our experiments reveal two key findings:

1. **ECS excels at fine-grained selection.** On LoCoMo (turn-level), ECS with Llama-3.1-8B achieves  $F1=0.265$ , a **71.83% relative improvement** over TF-IDF ( $F1=0.154$ ). This validates ECS’s core hypothesis: measuring answer distribution shift via KL divergence captures pragmatic utility that semantic similarity misses.

2. **ECS is model-dependent.** Interestingly, Qwen2.5-7B shows poor ECS performance ( $F1=0.020$  on LoCoMo), while Llama-3.1-8B succeeds. This suggests that model logprob distributions vary in informativeness for ECS. Llama’s

Table 1: Context selection results across datasets and models (F1 scores). LongMemEval: session-level selection (100 samples). LoCoMo: turn-level evidence selection (200 QA pairs). **Bold** indicates best method per column.

Method	LongMemEval (Sessions)		LoCoMo (Turns)	
	Qwen2.5-7B	Llama-3.1-8B	Qwen2.5-7B	Llama-3.1-8B
ECS (Ours)	0.152	0.139	0.020	<b>0.265</b>
TF-IDF	<b>0.649</b>	<b>0.649</b>	<b>0.154</b>	0.154
Dense (SBERT)	0.591	0.591	–	–
Random	0.024	0.024	0.003	0.003

next-token distributions appear more discriminative for context utility.

**3. Coarse selection favors semantic methods.** On LongMemEval (session-level), TF-IDF ( $F1=0.649$ ) and Dense retrieval ( $F1=0.591$ ) significantly outperform ECS ( $F1\sim 0.15$ ). At session granularity ( $\sim 6K$  tokens), the added noise overwhelms the logprob signal. ECS requires fine-grained contexts where individual pieces can meaningfully shift answer distributions.

### 4.3 Analysis

**Why does granularity matter?** ECS computes  $KL(P_{\text{with-context}} || P_{\text{base}})$  for each candidate context. When contexts are entire sessions, most content is irrelevant to the question, diluting the informative signal. Fine-grained turn selection allows ECS to isolate turns that directly shift answer probability.

**Why does model matter?** The stark difference between Qwen ( $F1=0.020$ ) and Llama ( $F1=0.265$ ) on identical data suggests model architecture affects logprob informativeness. Possible factors:

- Tokenizer vocabulary and token granularity
- Pre-training data distribution
- Instruction tuning methodology

This model-dependence is an important consideration for deploying ECS in practice.

## 5 Conclusion

We introduced Entropic Context Shaping (ECS), an information-theoretic framework for filtering context updates in LLM agents. By measuring pragmatic utility—whether a passage shifts the model’s answer distribution toward the correct answer—ECS addresses fundamental limitations of lexical similarity methods.

**Key Findings.** Our experiments on multi-turn context selection reveal:

1. **ECS excels at fine-grained selection.** On LoCoMo turn-level evidence selection, ECS with Llama-3.1-8B achieves  $F1=0.265$ , a 71.83% relative improvement over TF-IDF ( $F1=0.154$ ).
2. **Granularity matters.** At coarse session-level (LongMemEval), semantic methods dominate (TF-IDF  $F1=0.649$  vs. ECS  $F1\sim 0.15$ ). ECS requires fine-grained contexts where individual pieces can meaningfully shift answer distributions.
3. **Model dependence.** ECS performance varies significantly across models (Llama  $F1=0.265$  vs. Qwen  $F1=0.020$  on identical data), suggesting logprob informativeness depends on model architecture.

### Theoretical Contributions.

1. Formalized pragmatic utility as the signed change in answer probability
2. Identified the “direction problem”: KL magnitude alone cannot distinguish helpful vs. harmful passages
3. Provided theoretical guarantees for rejecting task-irrelevant updates (Theorem 1)

**Limitations.** ECS exhibits model-dependence: not all LLMs produce equally informative logprob distributions. Additionally, ECS requires access to model logits, limiting applicability to closed-source APIs. Performance degrades at coarse granularity where noise overwhelms the pragmatic signal.

**Future Work.** Promising directions include: (1) understanding which model properties yield informative logprobs for ECS, (2) developing API-compatible approximations using sampling, and (3) combining ECS with semantic pre-filtering for multi-stage context selection.

## References

- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations (ICLR)*. Oral presentation (top 1%).
- Sebastian Borgeaud and 1 others. 2022. Improving language models by retrieving from trillions of tokens. *ICML*.
- Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.
- Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*, 2nd edition. John Wiley & Sons, Hoboken, NJ.
- Kawin Ethayarajh and 1 others. 2022. Understanding dataset difficulty with v-useable information. *ICML*.
- Feiteng Fang, Yuelin Bai, Shiwen Ni, Min Guo, Wenjun Chen, and Kaiwen Liao. 2024. Enhancing noise robustness of retrieval-augmented language models with adaptive adversarial training. In *Proceedings of ACL*, pages 9962–9977.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Kelvin Guu and 1 others. 2020. Realm: Retrieval-augmented language model pre-training. *ICML*.
- Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2024. LongLLMLingua: Accelerating and enhancing LLMs in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Vladimir Karpukhin and 1 others. 2020. Dense passage retrieval for open-domain question answering. *EMNLP*.
- Woosuk Kwon and 1 others. 2023. Efficient memory management for large language model serving with pagedattention. *SOSP*.
- Patrick Lewis and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Yi Liu and 1 others. 2023. RECALL: A benchmark for LLMs robustness against external counterfactual knowledge. In *arXiv preprint arXiv:2311.08147*.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13851–13868.
- Joon Sung Park and 1 others. 2023. Generative agents: Interactive simulacra of human behavior. *UIST*.
- Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.
- Noah Shinn and 1 others. 2023. Reflexion: Language agents with verbal reinforcement learning. *NeurIPS*.
- Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.
- Babak Taati and 1 others. 2023. Large language models are fixated by red herrings: Exploring creative problem solving and einstellung effect using the only connect wall dataset. In *NeurIPS*.
- Guanzhi Wang and 1 others. 2024. Voyager: An open-ended embodied agent with large language models. *arXiv*.
- Jason Wei and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ArXiv:2410.10813.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. In *ICLR*.
- Yilun Xu and 1 others. 2020. A theory of usable information under computational constraints. *ICLR*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Shunyu Yao and 1 others. 2023. React: Synergizing reasoning and acting in language models. *ICLR*.

Zeyu Zhang, Xiaohe Bo Zhang, Chao Chen, and 1 others. 2024. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *NeurIPS*.

## A Proofs

### A.1 Proof of Theorem 1

*Proof.* Let  $u$  be a task-irrelevant update such that  $u \perp q \mid \mathcal{C}$ . We wish to show that  $D_{\text{KL}}(P_{\mathcal{C} \cup u} \| P_{\mathcal{C}}) \leq \epsilon$ .

By the definition of conditional independence:

$$P(y \mid \mathcal{C} \cup u, q) = P(y \mid \mathcal{C}, q) \quad (5)$$

for all  $y \in \mathcal{V}$ .

This implies  $P_{\mathcal{C} \cup u} = P_{\mathcal{C}}$ , and thus:

$$D_{\text{KL}}(P_{\mathcal{C} \cup u} \| P_{\mathcal{C}}) = 0 \quad (6)$$

In practice, finite model capacity introduces small deviations, yielding  $D_{\text{KL}} \leq \epsilon$  for  $\epsilon$  proportional to model approximation error.  $\square$

## B Evaluation Protocol

To ensure scientific validity, ACE and ECS were evaluated under identical conditions:

- **Same Questions:** Both methods evaluate the same questions from LongMemEval and LoCoMo
- **Same Contexts:** Identical candidate sessions/turns are scored by both methods
- **Same Ground Truth:** Evidence labels are held constant

The key difference is the *scoring function*:

- **ACE (TF-IDF):** Scores passages by lexical overlap with the query
- **ECS (LLM-Judge):** Scores passages by whether they help answer correctly

Table 2: Hyperparameter settings

Parameter	Value
Trajectory horizon ( $T$ )	8
Information threshold ( $\tau$ )	0.05
Length penalty ( $\lambda$ )	0.002
Top- $k$ tokens for KL	50
Epsilon smoothing	$10^{-10}$
ACE similarity threshold	0.85

## C Hyperparameter Settings

**Threshold Selection.** For the LLM-as-Judge operationalization, we use binary classification: a passage is considered helpful if the judge responds affirmatively. No threshold tuning is required for this approach.

### C.1 Dataset Statistics

Table 3: Dataset statistics

Statistic	LongMemEval	LoCoMo
Total samples	500	200
Samples used	100	200
Candidates/question	~50 sessions	~400 turns
Tokens/candidate	~6K	~50–200
Evidence items/question	1–3 sessions	2–5 turns

**LongMemEval.** Session-level selection task where the agent must identify which conversation sessions (out of ~50) contain information needed to answer the question. Each session contains approximately 6K tokens of conversational history.

**LoCoMo.** Turn-level evidence selection where the agent must identify specific dialogue turns that serve as evidence for answering questions about long conversational histories (~400 turns across multiple sessions).

## D Detailed Results

### D.1 Per-Benchmark Performance

Table 4 shows the full experimental results with F1 scores for each method across both datasets and models.

Table 4: Detailed experimental results (F1 scores)

Dataset	Method	Qwen2.5-7B	Llama-3.1-8B
LongMemEval (Session)	TF-IDF	0.649	0.649
	Dense	0.591	0.591
	ECS	0.152	0.139
LoCoMo (Turn)	TF-IDF	0.154	0.154
	ECS	0.020	0.265

## D.2 Granularity Analysis

The results reveal a clear interaction between context granularity and method effectiveness:

- **Coarse granularity (LongMemEval):** Semantic methods dominate. TF-IDF achieves  $F1=0.649$  vs. ECS  $F1\sim 0.15$ . At session-level ( $\sim 6K$  tokens), noise overwhelms the pragmatic signal.
- **Fine granularity (LoCoMo):** ECS excels. With Llama-3.1-8B, ECS achieves  $F1=0.265$  vs. TF-IDF  $F1=0.154$ —a 71.83% relative improvement.

## D.3 Model Dependence Analysis

The stark difference between Qwen ( $F1=0.020$ ) and Llama ( $F1=0.265$ ) on LoCoMo suggests that logprob informativeness varies significantly across model architectures.

## E Implementation Details

### E.1 Model Configuration

All experiments use HuggingFace Transformers with the following configuration:

- **Models:** Qwen2.5-7B-Instruct, Llama-3.1-8B-Instruct
- **Precision:** FP16 (half precision)
- **Max context length:** 8,192 tokens
- **Hardware:** NVIDIA V100 32GB
- **Logprobs:** Top-20 tokens per position

**vLLM Deployment (Optional).** For production environments requiring lower latency, we support vLLM deployment with prefix caching enabled:

- **Prefix caching:** Enabled (reduces amortized cost to  $O(|u|)$ )
- **GPU memory utilization:** 0.85
- **Recommended GPU:** NVIDIA A100 40GB or V100 32GB

### E.2 Latency Breakdown

Table 5 shows the latency components for ECS filtering on NVIDIA V100 32GB.

With vLLM prefix caching, the base context is cached across candidates, reducing amortized latency to  $\sim 16$ ms per candidate.

Table 5: Latency breakdown per candidate (V100 32GB)

Component	Transformers	vLLM
Base logprobs	50ms	5ms
Candidate logprobs	50ms	15ms
KL computation	1ms	1ms
<b>Total (cold)</b>	101ms	21ms
<b>Total (cached)</b>	51ms	16ms

## F Code and Reproducibility

All code, data, and experimental scripts are provided in the supplementary materials (supplementary\_materials.zip).

### Requirements.

- Python 3.9+
- PyTorch 2.0+
- HuggingFace Transformers 4.35+
- sentence-transformers
- vLLM 0.5+ (optional, for faster inference with prefix caching)