

Learning from Supervision with Semantic and Episodic Memory: A Reflective Approach to Agent Adaptation

Jackson Hassell, Dan Zhang, Hannah Kim, Tom Mitchell, Estevam Hruschka

Megagon Labs
{jackson, dan_z, hannah, tom, estevam}@megagon.ai

Abstract

We investigate how agents built on pretrained large language models (LLMs) can learn target classification functions from labeled examples without parameter updates. While conventional approaches like fine-tuning are often costly, inflexible, and opaque, we propose a memory-augmented framework that leverages both labeled data and LLM-generated critiques. Our framework uses episodic memory to store instance-level critiques—capturing specific past experiences—and semantic memory to distill these into reusable, task-level guidance. Across a diverse set of tasks, incorporating critiques yields up to a 24.8% accuracy improvement over retrieval-based (RAG-style) baselines that rely only on labels. Through extensive empirical evaluation, we uncover distinct behavioral differences between OpenAI and open-source models, particularly in how they handle fact-oriented versus preference-based data. To interpret how models respond to different representations of supervision encoded in memory, we introduce a novel metric, suggestibility. This helps explain observed behaviors and illuminates how model characteristics and memory strategies jointly shape learning dynamics. Our findings highlight the promise of memory-driven, reflective learning for building more adaptive and interpretable LLM agents.

1 Introduction

Large language models (LLMs) have demonstrated impressive generalization capabilities across a wide range of tasks. These AI agents rely on intelligence embedded in their pretrained parameters, and increasingly, on learning from task-specific signals, whether explicit (e.g., labeled supervision) or implicit (e.g., user interactions, feedback). A key challenge is enabling agents to continuously improve their performance and generalize to unseen domains or tasks by distilling knowledge from such signals and storing them in a reusable and interpretable form.

Traditional approaches to learning from new signals often involve updating model parameters through fine-tuning (Radford et al. 2018; Howard and Ruder 2018) or adaptation mechanisms such as parameter-efficient methods (e.g., LoRA adapters) (Houlsby et al. 2019; Hu et al. 2022). While effective, these approaches incur computational cost, require retraining for every new signal or task, and often lack interpretability or controllability. Furthermore, they provide limited support for never-ending learning, where an agent

must continuously adapt without retraining from scratch or storing large sets of models.

An alternative paradigm is memory-augmented learning (Weston, Chopra, and Bordes 2015; Zhong et al. 2024), where the underlying model remains frozen, and adaptation occurs through interaction with an external memory. This memory stores relevant task knowledge, examples, demonstrations, or explanations, that can be retrieved at inference time to inform the model’s decisions. Among such approaches, in-context learning (ICL) (Dong et al. 2024a) has emerged as a simple yet powerful mechanism, where the model is conditioned on a prompt consisting of a small number of examples (few-shot learning). However, directly incorporating supervised signals in the LLM context often relies on only few-shot input-output examples and tends to result in shallow pattern mimicking, due to a lack of deeper abstraction or conceptual understanding.

Recent work (Madaan et al. 2023; Yao et al. 2023; Shinn et al. 2023) has highlighted the capacity of LLM agents to not only perform tasks but also critique them, generating feedback and identifying patterns of errors in their own outputs. However, these methods rely on the model’s own parametric knowledge (and optionally feedback from an interactive environment) to direct these critiques. In contrast to previous work, we are interested in grounding these critiques in supervised data as a way to provide novel information to the agent and to avoid simply reinforcing the parametric biases of the base LLM.

Inspired by human tutoring, where feedback often includes explanations of mistakes and guidance for improvement, we explore whether such reflective insights can be distilled into reusable knowledge for future tasks. Instead of merely memorizing example responses, we hypothesize that an agent that internalizes structured feedback can develop a deeper understanding of task requirements and generalize more effectively to new examples.

In this paper, we investigate how LLM agents can effectively and continuously learn from supervised signals and deeper reasoning provided by critiques, incorporating these insights into memory.

2 Learning from Supervised Signals

For large language model (LLM)-based agents, the availability of supervised signals, in the form of labeled datasets or

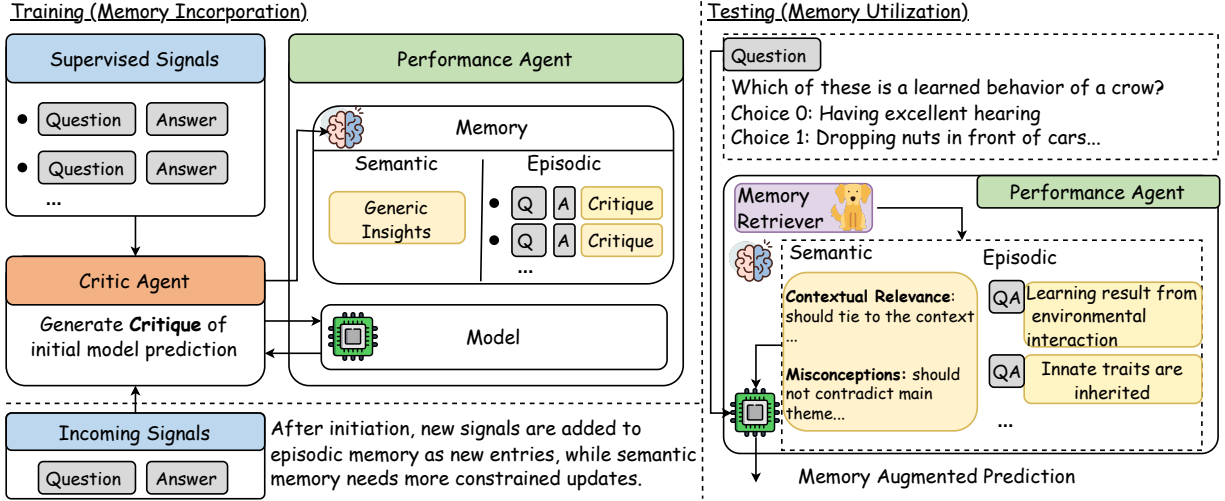


Figure 1: Agents learn from supervised signals by incorporating them into memory. At inference time, both task-level insights (semantic memory) and context-specific information (episodic memory) can support more informed decision-making.

continuously incoming feedback from users or the environment, presents a wide range of opportunities for building agents that can learn and improve continuously.

As illustrated in Figure 1, we refer to our task-solving agent as the *performance agent* (PA). The PA consists of an LLM model (the prediction model), which is queried to perform tasks, and a memory module, which it can read from and write to. Given a new task to which we want to adapt the agent, we begin with an initial labeled dataset:

$\mathcal{D}_{\text{train}}^{\text{init}} = \{(x_i, y_i)\}_{i=1}^N$, where each x_i represents a task-related question or request, and y_i denotes the corresponding correct label or answer. The performance agent processes inputs x_i from a test set $\mathcal{D}_{\text{test}}$ and produces initial predictions denoted by $\text{PA}(x_i)$.

To enhance the capabilities of the PA, we introduce a second component: the *critic agent* (CA) making use of a LLM critic model. The CA takes as input one tuple (x_i, y_i) along with the PA’s prediction $\text{PA}(x_i)$, and outputs a text critique aimed at improving the PA’s performance.

2.1 What to Remember?

Critique is a widely used approach for improving model performance and guiding iterative refinement by identifying errors, uncovering blind spots, and providing actionable feedback for enhancement (Shinn et al. 2023; Gou et al. 2024; Chen et al. 2024). In our setup, we employ a label-driven critique generation process, where the critic agent uses provided ground-truth answers as part of its input to generate critiques. For each question in the dataset, the performance agent first produces an initial prediction. The critic agent is then given the correct answer and asked to critique the performance agent’s output. Each critique is structured into the following fields:

- **Assertion:** A reiteration of the correct answer to the question, and a judgment regarding the correctness of the

performance agent’s response.

- **Rationale:** An instance-specific explanation detailing why the correct answer is valid and why the performance agent’s response was correct or incorrect.
- **Reflection:** A broader, generalizable insight that may be applicable to similar questions in the future.

This design addresses a key challenge observed in our empirical studies: critic agents sometimes persist in their own incorrect understanding from its underlying model when generating critiques, even after being shown the correct answer. Because the critic agent draws on the model’s parametric knowledge, it can inherit pretraining biases that reinforce such biases. To mitigate this, we require the critic agent to explicitly restate the correct answer and make a clear assertion about the correctness of the initial prediction before offering a rationale or reflection. This explicit structure significantly reduces confirmation bias.

We further decompose critiques into two conceptual layers—*rationale* (local) and *reflection* (global)—to balance specificity and generalizability. An ideal rationale should provide a detailed explanation tailored to the specific instance, while a reflection should capture broader insights that can be applied to unseen examples in the future. This structured format resulted in noticeably higher-quality critiques.

Example Critique

Question: Does short-term treatment with proton pump inhibitors cause rebound aggravation of symptoms?

PA Response: Yes

Critique:

- **Assertion:** No
- **Rationale:** Short-term treatment with proton pump

inhibitors (PPIs) generally does not cause rebound aggravation of symptoms upon discontinuation. Studies have shown that any perceived increase in symptoms after stopping PPIs may be temporary...

- *Reflection:* In the broader context, medical treatments may sometimes be misattributed with rebound phenomena, but each class of medication has its own pharmacological profiles. In the case...

3 Incorporating Critiques into Memory

Next, we investigate how learned critiques can be effectively incorporated into the performance agent’s memory. We adopt two primary forms of PA memory: *semantic memory* and *episodic memory*, both of which are well-established in agentic learning literature (Sumers et al. 2024).

3.1 Semantic Memory

Semantic memory encodes generalizable knowledge across the entire dataset. In this work, we construct semantic memory by *summarizing all of the critiques* across a dataset into a unified knowledge representation. This allows the performance agent to draw on abstract insights during inference in future tasks. Semantic memory aims to capture insights that are broadly applicable across the entire task domain. To utilize it at inference time, we augment the performance agent’s prompt with these insights in the form of additional instructions. This strategy is referred to as SEM_CRIT. Semantic memory typically takes the form of a bulleted list of task-specific advice and cautions (see Appendix for an example), though its format is intentionally left vague to give the critic agent flexibility in capturing patterns from previously generated critiques.

3.2 Episodic Memory

While semantic memory offers concise and broadly applicable knowledge, it often fails to capture nuanced patterns or context-specific behaviors, particularly in diverse datasets. Episodic memory addresses this by enabling the agent to recall specific past instances, effectively allowing it to "revisit" similar scenarios where successes or failures occurred, along with accompanying context and critical reasoning.

The key to effective use of episodic memory lies in the retrieval of relevant examples. The agent retrieves relevant memories from similar prior cases and conditions on both the original examples and their critiques, learning to weigh and incorporate only the most pertinent critiques rather than attending to all examples equally. This strategy is referred to as (EP_CRIT). Following the retrieval-augmented generation (RAG) paradigm, we identify the top $K = 5$ most similar data points to a test input x_i using semantic embeddings. The corresponding memory entries, containing critical thinking artifacts are then used as additional demonstrations for the performance agent. See the Appendix for detailed results and discussion on why we chose $K = 5$.

3.3 Combining Semantic and Episodic Memory

To harness the complementary strengths of both memory types, we introduce EP+SEM_CRIT. This hybrid strategy presents the performance agent with both high-level semantic instructions and context-specific episodic examples. By unifying generalizable semantic memory with detailed situational context, these approaches aim to support more robust and adaptive reasoning during inference. These are unified by simply concatenating the semantic memory to the end of the episodic memory.

4 Empirical Evaluation

4.1 Datasets

To evaluate the effectiveness of various memory-augmented learning strategies under diverse conditions, we conduct empirical studies across multiple datasets. The tasks cover a range of settings, including fact-oriented question answering, ranking, and retrieval-based QA. The following datasets were selected based on several criteria: domain diversity, easily verifiable answers, and low enough baseline accuracy to allow room for improvement.

Multi-Condition Ranking (Pezeshkpour and Hruschka 2025) Given a list of 5 items, sort them in order along 3 logical conditions. Converted into a 4-choice multiple-choice task.

NFCorpus (Boteva et al. 2016) Given a medical article and two medical papers, determine which paper is cited directly by the article’s bibliography.

PubMed (Jin et al. 2019) Determine if a highly technical medical statement is true or false, across many different medical domains.

To mitigate potential bias from LLMs being exposed to public datasets during pretraining, we additionally evaluated our strategies on four personal preference datasets. The task was to predict whether a given item belonged to a user’s history. Even if the model had encountered these datasets during training, it would be unlikely to memorize preferences associated with individual user IDs.

Steam Pref (Tamber 2017) Video game playtime per user on the PC platform Steam. Sampled only games that were played for at least 5 hours.

Book Pref (Ziegler et al. 2005) Book ratings per user. Actual ratings were not used - the task was formatted as predicting whether a user is more or less likely to read a given title.

Anime Pref (Union 2016) Anime ratings per user from the website MyAnimeList. Actual ratings were not used - the task was formatted as predicting whether a user is more or less likely to watch a given title.

Movie Pref (Parashar 2023) Movie ratings per user, based on the MovieLens dataset. Sampled only movies rated 3/5 or higher.

For the preference datasets, we randomly selected three users per dataset. For each user, 250 items were sampled from their history and 250 from outside it, prioritizing favorites when possible. Users were treated independently, with no memory shared across them. For all other datasets, 500 questions were randomly sampled and evenly split into training and testing sets. Additional dataset-specific preprocessing details are provided in the Appendix.

We separate these datasets into two distinct groups: fact-oriented datasets, which includes Multi-Condition Ranking, NFCorpus, and PubMed, and preference-based datasets, which include the other four.

4.2 Experimental Setup

We compared different learning strategies against two baseline setups: zero_shot and EP_LABEL. The zero_shot baseline reflects the agent’s performance without memory or demonstrations. The EP_LABEL baseline is a retrieval-based few-shot strategy that includes $K = 5$ example question–answer pairs (x_i, y_i) retrieved from the training set using embedding similarity, consistent with all other EP strategies, where x_i is the input and y_i the corresponding answer. EP_LABEL is a strong baseline—combining RAG and few-shot demonstration—as it has full access to the same supervised signals. While EP_LABEL can be categorized as a RAG solution, we adopt the EP naming convention to highlight both its similarities to and differences from other strategies.

We experiment with five different models. In Table 1, we begin with OpenAI’s GPT-4o-mini (OpenAI 2024) and the reasoning-oriented model o4-mini (OpenAI 2025), where both the performance agent and the critic agent use the same backbone model. To extend our exploration to open-source models, we also conducted experiments (Table 2) on three additional models: Mistral’s Mixtral 8x22B MoE (Mistral 2024), Meta’s Llama 4 Scout (Meta 2025), and Meta’s Llama 3.1 8B (Meta 2024). These models span a range of sizes and include both dense and mixture-of-experts (MoE) architectures. We also further explore the impact of mixing different model choices between the performance and critic agents.

4.3 Results on OpenAI models

Table 1 compares various learning strategies against baselines across seven datasets using two OpenAI models. All preference-based results are averaged across three users per dataset (see Appendix for per-user breakdowns). The results show notable variation across datasets, both in baseline performance and in the impact of memory-augmented learning.

For the first three datasets, which are more fact-oriented, memory augmentation provided limited benefit. For example, GPT-4o-mini achieved a modest 1.6% improvement on NFCorpus over the zero-shot baseline, and o4-mini showed a 0.4% gain on Multi-Cond Ranking. In all other fact-oriented trials, adding critiques did not improve performance.

In contrast, preference-based datasets showed more consistent gains from critique-based memory over EP_LABEL. For GPT-4o-mini, three out of four preference datasets improved, with an average gain of 5.1%; for o4-mini, all four preference datasets improved, averaging a 2.5% gain. Gains of up to 10% between _CRIT strategies and EP_LABEL suggest that

Model and Experiment	Multi-Cond. Ranking	NFCorpus	PubMed	Steam Pref	Book Pref	Anime Pref	Movie Pref
gpt-4o-mini							
zero_shot	56.8	<u>85.6</u>	<u>62.4</u>	52.8	52.0	47.9	49.9
EP_LABEL	65.2	84.4	63.2	57.6	55.2	51.1	53.2
EP_CRIT	65.2	83.6	62.0	62.7	53.8	<u>54.4</u>	57.7
SEM_CRIT	58.4	87.2	59.6	60.1	45.5	48.8	<u>58.7</u>
EP+SEM_CRIT	56.8	85.2	61.6	<u>62.4</u>	<u>54.2</u>	61.7	59.3
o4-mini							
zero_shot	87.6	89.2	62.0	50.4	49.3	51.1	51.6
EP_LABEL	<u>90.0</u>	91.6	66.8	60.0	49.7	63.9	<u>59.3</u>
EP_CRIT	80.8	89.2	<u>64.8</u>	<u>60.6</u>	<u>50.5</u>	<u>68.1</u>	60.7
SEM_CRIT	69.6	88.8	60.4	48.0	48.2	48.9	50.9
EP+SEM_CRIT	90.4	<u>90.8</u>	61.5	61.5	52.4	68.3	57.6

Table 1: Performance agent accuracy across datasets for gpt-4o-mini and o4-mini. We use EP, SEM, and EP+SEM to denote episodic, semantic, and combined memory. Results on preference datasets are averaged across all users. For each model and dataset, the highest score is **bolded** and the second-highest is underlined.

critiques can provide useful additional signals beyond the supervised label.

When comparing memory augmentation strategies, episodic memory with critiques (EP_CRIT) generally outperformed semantic memory (SEM_CRIT), achieving better scores in 12 out of 14 comparisons. GPT-4o-mini showed an average 3.0% improvement of EP_CRIT over SEM_CRIT, while o4-mini saw an average improvement of 8.6%. These results suggest that models benefit more from a small number of targeted examples than from a high-level summary, possibly due to limitations in semantic abstraction—e.g., summaries being overly general or filled with superficially correct but uninformative content.

To test whether combining both memory types is beneficial, we also evaluated the hybrid strategy EP+SEM_CRIT. It outperformed both single-source memory strategies in 8 out of 14 cases, with GPT-4o-mini showing a marginal average 0.3% gain over EP_CRIT and o4-mini showing 1.1%. Overall, while combining episodic and semantic memory can sometimes help, the additional cost—especially from generating semantic memory over the entire training set—might not be justified.

4.4 Results on Open-Source Models

Next, we examine performance on open-source models. Table 2 shows results for three: Llama 4 Scout, a compact variant of Meta’s Llama 4 series optimized for efficiency; Mixtral 8x22B, a mixture-of-experts model from Mistral using 2 of 8 active 22B experts per pass; and Llama 3.1 8B, a dense 8B-parameter transformer that balances capability with a relatively small size. The zero_shot baseline performance of the OpenAI models tended to be higher than those of the open-source models, implying that the quality of the critiques made by these models may be higher as well. To assess the impact of the critic, we also vary the critic model by mixing

Model and Experiment	Multi-Cond. Ranking	NFCorpus	PubMed	Steam Pref	Book Pref	Anime Pref	Movie Pref
Performance Model: Llama 4 Scout							
zero_shot	66.4	57.2	66.8	49.9	51.9	47.9	49.2
EP_LABEL	74.4	69.6	66.4	61.3	54.5	58.1	58.4
Critic Model: Llama 4 Scout							
EP_CRIT	77.6	82.8	70.0	61.5	51.2	59.1	57.2
SEM_CRIT	62.8	66.8	63.2	48.9	47.8	48.8	51.3
EP+SEM_CRIT	<u>78.4</u>	82.8	68.8	57.8	51.9	55.9	<u>57.6</u>
Critic Model: gpt-4o-mini							
EP_CRIT	70.0	90.8	66.4	61.6	<u>55.4</u>	67.3	55.3
SEM_CRIT	67.2	86.0	58.0	52.7	46.8	48.8	51.6
EP+SEM_CRIT	68.0	88.0	66.8	<u>63.1</u>	55.8	64.5	58.8
Critic Model: o4-mini							
EP_CRIT	67.6	86.4	63.6	47.3	49.8	47.5	51.3
SEM_CRIT	76.4	<u>87.2</u>	<u>68.4</u>	60.3	54.2	64.4	55.2
EP+SEM_CRIT	82.0	<u>90.4</u>	67.2	64.7	55.4	<u>65.6</u>	56.3
Performance Model: Mixtral 8x22B							
zero_shot	60.4	60.4	51.6	56.2	52.0	48.3	51.1
EP_LABEL	60.8	70.0	48.4	57.6	<u>53.0</u>	52.1	49.7
Critic Model: Mixtral 8x22B							
EP_CRIT	73.2	77.2	53.2	57.4	51.0	51.2	49.9
SEM_CRIT	45.6	53.6	39.6	52.2	47.3	50.4	48.5
EP+SEM_CRIT	71.2	83.2	42.8	54.1	49.4	50.7	45.7
Critic Model: gpt-4o-mini							
EP_CRIT	79.6	84.4	49.6	58.1	55.0	<u>55.2</u>	55.3
SEM_CRIT	32.4	72.0	44.4	56.1	48.1	48.9	50.8
EP+SEM_CRIT	80.4	<u>85.6</u>	53.2	55.3	53.8	53.3	50.9
Critic Model: o4-mini							
EP_CRIT	<u>81.6</u>	86.8	51.6	<u>59.2</u>	51.8	57.9	<u>53.5</u>
SEM_CRIT	27.6	79.2	43.2	50.3	48.7	47.6	50.7
EP+SEM_CRIT	85.6	85.2	53.2	60.5	51.8	50.8	51.7
Performance Model: Llama 3.1 8B							
zero_shot	42.8	<u>83.6</u>	62.0	54.6	51.7	49.2	50.7
EP_LABEL	78.4	<u>83.6</u>	66.4	61.8	53.2	60.3	<u>54.7</u>
Critic Model: Llama 3.1 8B							
EP_CRIT	89.2	64.4	58.8	55.5	52.0	54.1	52.0
SEM_CRIT	23.2	84.0	60.4	51.4	48.4	49.9	50.3
EP+SEM_CRIT	84.0	62.4	47.2	52.7	49.8	49.9	52.0
Critic Model: gpt-4o-mini							
EP_CRIT	89.6	83.2	46.4	53.6	50.6	55.6	49.5
SEM_CRIT	40.4	82.8	<u>65.6</u>	54.1	51.1	49.3	52.5
EP+SEM_CRIT	<u>90.4</u>	80.8	46.4	57.1	51.0	50.8	49.6
Critic Model: o4-mini							
EP_CRIT	92.8	83.6	53.2	58.6	52.0	<u>57.9</u>	48.1
SEM_CRIT	45.6	82.8	55.6	50.9	52.5	51.6	51.1
EP+SEM_CRIT	88.4	81.6	48.4	<u>59.1</u>	<u>53.0</u>	50.4	56.5

Table 2: Performance agent accuracy for three open-source models used as the base LLM (Llama 4 Scout, Mixtral 8x22B, and Llama 3.1 8B), each evaluated with critiques generated by different models: the same model used for prediction, GPT-4o-mini, and o4-mini. For each LLM and dataset, the highest score is **bolded** and the second-highest is underlined.

in critiques from two OpenAI models.

For comparison among the critique-based strategies, we observe a similar pattern with OpenAI models as shown in Table 1: EP_CRIT generally outperforms SEM_CRIT, while the hybrid method offers occasional benefits. However, we observe a different pattern among these models’ performances and their capabilities to utilize critiques, compared to the OpenAI models.

While using the same critic model (first five rows for each model), we find that for fact-oriented data, _CRIT strategies outperform baselines across all models except Llama 3.1 8B on PubMed. The average performance gains of the best-performing _CRIT strategy over the best-performing baseline across all datasets are: Llama 4 Scout — 6.8%, Mixtral 8x22B — 9.1%, and Llama 3.1 8B — 1.7%. When critiques are instead generated by OpenAI models, these improvements become even more pronounced: Llama 4 — 10.1%, Mixtral — 14.4%, and Llama 3.1 — 4.5%. Notably, on the multi-conditional ranking task, the Mixtral model achieves a **24.8% improvement** over the EP_LABEL baseline when leveraging critiques generated by o4-mini.

For preference data, in contrast, _CRIT strategies do not consistently outperform EP_LABEL when using the same critic models. With Llama 4 Scout, EP_CRIT performs slightly better on two out of four preference datasets. For Mixtral 8x22B and the smaller Llama 3.1 8B, _CRIT methods are generally outperformed by EP_LABEL. However, when provided with different (and potentially higher-quality) critiques from OpenAI models, we begin to see improvements: the best-performing _CRIT strategy outperforms the EP_LABEL baseline by 3.9% (Llama 4), 3.5% (Mixtral), and 0.3% (Llama 3.1).

To conclude, we observe an opposite trend in performance across the two groups of data: **OpenAI models tend to benefit more from critiques on preference data and outperform the strong RAG-style baseline, but show limited gains on fact-oriented data. In contrast, open-source models show clear improvements on fact-oriented data but struggle to effectively leverage additional information from their own critiques on preference data. We also observe that critiques generated by OpenAI agents generally led to better outcomes for the open-source models than their own critiques.** This could imply that the OpenAI critiques were higher quality, or that mixing base LLMs for the performance agent and critique agent allows the LLMs to cover for each other’s gaps in knowledge.

To further investigate why models exhibit such different behaviors, we introduce a novel metric: suggestibility.

4.5 Suggestibility

In memory-augmented agentic learning, it is crucial not only to generate the best possible critique for inclusion in memory, but also to ensure that the model is actually receptive to it, i.e., that it can be “persuaded” by the insight. This receptivity, which we term *suggestibility*, is influenced by a compound of factors: the model architecture, the nature of the task, and the format in which the memory is represented.

To better quantify this phenomenon, we define a *suggestibility* metric S , which captures the difference in an

agent’s performance when given a best-effort critique versus when given an intentionally misleading one (generated by flipping the ground-truth label). Formally,

$$S = \frac{1}{|D|} \sum_{x_i \in D} \mathbb{1}[\text{PA}(x_i \mid \text{Ins}(x_i, y_i)) = y_i] - \frac{1}{|D|} \sum_{x_i \in D} \mathbb{1}[\text{PA}(x_i \mid \text{Ins}(x_i, \neg y_i)) = y_i]$$

where PA denotes the performance agent, Ins refers to the critic agent, and D is the evaluation dataset. Note that in real-world settings, the true label y_i is not available to either PA or Ins; thus, this metric represents an idealized or “cheating” scenario, using artificially constructed best and adversarial insights for controlled experimentation.

To explore how different components affect a model’s suggestibility, we report S across three experimental conditions, varying the context provided to the performance agent. As shown in Table 3, X indicates the presence of the question and Y denotes inclusion of the ground-truth label, lines labeled +CRIT refer to cases where critiques (for both the ground-truth and flipped labels) are included in the suggestibility tests.

	Multi- Cond. Ranking	NFCorpus	PubMed	Steam Pref	Book Pref	Anime Pref	Movie Pref
gpt-4o-mini							
XY	45.6	6.4	98.4	100.0	100.0	100.0	100.0
XY+Crit	98.4	40.0	99.6	100.0	100.0	100.0	100.0
X+Crit	100.0	70.8	93.2	100.0	99.8	100.0	100.0
o4-mini							
XY	90.4	5.6	97.2	100.0	100.0	99.9	99.9
XY+Crit	88.8	22.0	99.2	100.0	100.0	100.0	100.0
X+Crit	96.8	52.0	98.8	98.6	98.9	99.1	99.7
Llama 4 Scout							
XY	76.4	26.8	91.2	100.0	100.0	100.0	100.0
XY+Crit	86.8	46.8	95.6	100.0	100.0	100.0	100.0
X+Crit	76.0	42.8	94.0	100.0	100.0	98.1	99.7
Mixtral 8x22B							
XY	90.4	0.0	83.2	100.0	99.8	100.0	99.9
XY+Crit	94.0	51.6	100.0	86.5	89.4	98.7	96.4
X+Crit	91.6	56.0	99.6	89.5	88.8	95.3	94.8
Llama 3.1 8B							
XY	92.0	34.8	94.0	99.7	99.2	100.0	99.9
XY+Crit	95.6	70.4	100.0	99.7	99.2	98.3	98.3
X+Crit	94.0	60.4	97.6	95.1	96.2	86.1	85.5

Table 3: Suggestibility scores using different models across datasets (preference datasets averaged across users). X represents the question, Y and CRIT denotes the presence of an answer and critique in the suggestibility test.

Model suggestibility is highly task-dependent. Even in the direct “cheating” case—where the model is given the X, Y pair from the test data—some models may refuse to change their prediction at all. For example, Mixtral 8x22B on NFCorpus shows zero changes after seeing the “cheating”

label. In contrast, Mixtral 8x22B is far more suggestible on other datasets, reaching up to 100% (i.e., always following the signal in the suggestion).

Models are generally more suggestible on preference data than on fact-oriented data. We hypothesize that this reflects a tension between knowledge encoded in a pretrained LLM’s parameters and the information provided by labeled examples. In fact-based domains, LLMs are likely more competent due to prior exposure to relevant facts during pretraining. A notable exception is the PubMed dataset, where the complexity of medical queries introduces enough ambiguity for critiques to meaningfully influence model outputs. In contrast, in preference-based domains, models cannot have learned individual user preferences—especially with anonymized users—so they lack parametric knowledge.

Comparing XY results to XY+Crit or X+Crit in Table 3, we observe a general increase in suggestibility when critiques are provided in addition to, or in place of, the label Y. This positive effect is expected, as critiques offer both local rationale and global reflections that might serve as justification, making the label more persuasive.

However, there are exceptions. On all preference datasets, Mixtral 8x22B and Llama 3.1 8B exhibit lower suggestibility when critiques are provided (X+Crit compared to XY). This aligns with the results in Table 2, where Llama 3.1 8B (using the same critic model) is outperformed by EP_LABEL on all four preference datasets. Similarly, Mixtral 8x22B is outperformed by EP_LABEL on three out of four datasets, with the sole exception showing only a marginal 0.2% improvement.

These findings help explain the seemingly contradictory behavior on preference data: in such cases, Mixtral and the 8B Llama 3.1 models are less responsive to critiques but more receptive to direct labels, which favors the EP_LABEL baseline.¹

The unusual patterns observed in Llama 3.1 8B and Mixtral 8x22B may stem from limitations in reasoning capabilities due to model size or pretraining data. In this paper, we aim to highlight the connection between model suggestibility and behavioral differences, while leaving a deeper investigation into the root causes of suggestibility variation to future work.

4.6 Dataset Size Scaling

Memory-augmented learning for agents enables the ability to learn from small amounts of training data while continually enriching the training set over time, particularly through the use of episodic memory. To investigate the influence of training data size, we conduct an additional analysis presented in Table 4. Specifically, we re-run the pipeline using gpt-4o-mini on each of the preference datasets, training with only 25%, 50%, and 75% of the original data. Both EP and EP+SEM methods begin to show improvement even at 25% of the data, with accuracy continuing to increase as more data is incorporated into memory. Methods that leverage se-

¹We note that suggestibility is not the only factor influencing model behavior. For example, Llama 4 Scout demonstrates strong critique-based suggestibility yet is still outperformed by EP_CRIT. Factors such as the quality of the critique may also play a significant role.

Experiment (Training Percentage)	Steam Pref	Book Pref	Anime Pref	Movie Pref
Baselines				
zero_shot	52.8	52.0	47.9	49.9
EP_LABEL	55.8	50.9	51.1	53.2
EP_CRIT				
25%	61.6	49.5	55.7	56.1
50%	62.9	51.3	54.9	57.3
75%	64.1	53.4	57.6	57.1
100%	62.7	53.8	54.4	57.7
EP+SEM_CRIT				
25%	57.8	48.5	55.1	57.1
50%	60.1	51.2	55.6	56.8
75%	59.7	50.6	58.9	60.5
100%	62.4	54.2	61.7	59.3

Table 4: Accuracy with varying size of training dataset on preference datasets using gpt-4o-mini. For each strategy, the highest score is **bolded**.

mantic memory are more sensitive to limited training data, as lower-quality summary-level insights may result from reduced context. Across datasets, performance tends to plateau between 75% and 100%, suggesting that model performance may be reaching saturation under the current data conditions.

5 Related Work

Agentic Memory Recent LLM-based agent research has focused on memory management challenges due to context length limitations. The predominant approach is retrieval-based augmentation (RAG), using embedding similarity for memory retrieval. Memories range from simple input/output copies to complex structures: Reflexion (Shinn et al. 2023) stores agent self-reflections, Voyager (Wang et al. 2024) maintains reusable agent-created tools, and Generative Agents (Park et al. 2023) employs a two-tier system of event streams and higher-level reflections.

Fine-tuning While fine-tuning is a well-established way of improving a model’s performance in a specific area (Dodge et al. 2020), it presents challenges such as: extensive labeled data requirements (Vieira et al. 2024), catastrophic forgetting (Luo et al. 2025), computational expense (Hu et al. 2022), and inapplicability to closed-source models.

In-Context Learning In-context learning treats models as black boxes, adjusting inputs to influence outputs (Dong et al. 2024b). Simple prompt modifications like appending “Let’s think step by step” can significantly improve performance (Kojima et al. 2022). Few-shot learning enhances results by providing question-answer examples (Brown et al. 2020). Reflection-based approaches, where models reason over feedback about their decisions, enable autonomous improvement (Shinn et al. 2023; Yao et al. 2023). However, most research focuses on feedback from simulated environments (Wang et al. 2024), with limited exploration of other feedback mechanisms.

6 Discussion

Our findings reveal a consistent pattern where episodic memory strategies (EP_CRIT) outperformed semantic memory approaches (SEM_CRIT) across most experimental conditions. This preference for concrete, contextual examples over generalized insights reflects a fundamental distinction between lazy and eager generalization strategies. Much like k-nearest neighbor methods, which often outperform regression-based approaches when training data is abundant, our lazy-learning episodic method has the advantage of not needing to commit to learning the full function over the entire domain. Instead, it learns only a local approximation to the function at the current query point, allowing for greater flexibility and specificity that may be lost during semantic abstraction.

Our analysis has centered on the accuracy of different agentic learning strategies, but design choices also impact computational cost and the ability to incorporate ongoing supervision. Semantic memory typically requires greater training-time computation due to summarization or distillation, whereas episodic memory simply stores past experiences with minimal processing. At inference time, however, semantic memory offers more readily applicable knowledge, while episodic memory relies on retrieval quality. This trade-off suggests that the optimal strategy may depend on the size of the supervised dataset and the frequency of inference—semantic memory may be better suited for frequent inference under sparse supervision, while episodic memory may be preferable when supervision is abundant and retrieval is reliable. Additional quantitative analysis can be found in the Appendix.

An additional interesting direction for exploring model suggestibility is to disentangle how much a model’s behavior changes due to genuinely incorporating supervised signals into its internal beliefs versus merely adapting its responses to please the user. In our empirical study, we observed that models exhibited higher suggestibility scores when critiques were attributed to the user, compared to when the same critiques were believed to originate from the model itself or another model. This suggests that the perceived source of feedback plays a significant role in how seriously the model treats the signal, opening up opportunities to better understand and guide belief formation in interactive learning systems.

7 Conclusion

We present a memory-augmented framework that enables LLM agents to learn classification functions from labeled examples and model-generated critiques without parameter updates. By combining episodic memory for instance-level experiences and semantic memory for task-level guidance, our approach supports continual adaptation through structured supervision. Experiments show up to 24.8% accuracy gain over label-only baselines and reveal distinct behavioral differences between OpenAI and open-source models across fact- and preference-based tasks. We introduce a novel metric, suggestibility, to explain how models internalize feedback via memory. Our findings highlight the potential of reflective, memory-driven learning as a lightweight and interpretable strategy for improving LLM adaptability.

References

- Boteva, V.; Gholipour, D.; Sokolov, A.; and Riezler, S. 2016. A Full-Text Learning to Rank Dataset for Medical Information Retrieval.
- Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20*. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.
- Chen, X.; Lin, M.; Schärli, N.; and Zhou, D. 2024. Teaching Large Language Models to Self-Debug. In *The Twelfth International Conference on Learning Representations*.
- Dodge, J.; Ilharco, G.; Schwartz, R.; Farhadi, A.; Hajishirzi, H.; and Smith, N. 2020. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. arXiv:2002.06305.
- Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Ma, J.; Li, R.; Xia, H.; Xu, J.; Wu, Z.; Chang, B.; Sun, X.; Li, L.; and Sui, Z. 2024a. A Survey on In-context Learning. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 1107–1128. Miami, Florida, USA: Association for Computational Linguistics.
- Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Ma, J.; Li, R.; Xia, H.; Xu, J.; Wu, Z.; Chang, B.; Sun, X.; Li, L.; and Sui, Z. 2024b. A Survey on In-context Learning. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 1107–1128. Miami, Florida, USA: Association for Computational Linguistics.
- Gou, Z.; Shao, Z.; Gong, Y.; yelong shen; Yang, Y.; Duan, N.; and Chen, W. 2024. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. In *The Twelfth International Conference on Learning Representations*.
- Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-Efficient Transfer Learning for NLP. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 2790–2799. PMLR.
- Howard, J.; and Ruder, S. 2018. Universal Language Model Fine-tuning for Text Classification. In Gurevych, I.; and Miyao, Y., eds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 328–339. Melbourne, Australia: Association for Computational Linguistics.
- Hu, E. J.; yelong shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.
- Jin, Q.; Dhingra, B.; Liu, Z.; Cohen, W.; and Lu, X. 2019. PubMedQA: A Dataset for Biomedical Research Question Answering. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2567–2577. Hong Kong, China: Association for Computational Linguistics.
- Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713871088.
- Luo, Y.; Yang, Z.; Meng, F.; Li, Y.; Zhou, J.; and Zhang, Y. 2025. An Empirical Study of Catastrophic Forgetting in Large Language Models During Continual Fine-tuning. arXiv:2308.08747.
- Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegrefe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; Gupta, S.; Majumder, B. P.; Hermann, K.; Welleck, S.; Yazdanbakhsh, A.; and Clark, P. 2023. Self-Refine: Iterative Refinement with Self-Feedback. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 46534–46594. Curran Associates, Inc.
- Meta. 2024. Introducing Llama 3.1: Our most capable models to date. <https://ai.meta.com/blog/meta-llama-3-1/>.
- Meta. 2025. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation. <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>.
- Mistral. 2024. Cheaper, Better, Faster, Stronger. <https://mistral.ai/news/mixtral-8x22b>.
- OpenAI. 2024. GPT-4o Mini: advancing cost-efficient intelligence. <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>.
- OpenAI. 2025. Introducing OpenAI o3 and o4-mini. <https://openai.com/index/introducing-o3-and-o4-mini/>.
- Parashar, M. 2023. Movie Recommendation System. <https://www.kaggle.com/datasets/parasharmanas/movie-recommendation-system>.
- Park, J. S.; O'Brien, J.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative Agents: Interactive Simulacra of Human Behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST '23*. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701320.
- Pezeshkpour, P.; and Hruschka, E. 2025. Multi-Conditional Ranking with Large Language Models. In Chiruzzo, L.; Ritter, A.; and Wang, L., eds., *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2863–2883. Albuquerque, New Mexico: Association for Computational Linguistics. ISBN 979-8-89176-189-6.
- Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I.; et al. 2018. Improving language understanding by generative pre-training.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: language agents with verbal reinforcement learning. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 8634–8652. Curran Associates, Inc.

Sumers, T.; Yao, S.; Narasimhan, K.; and Griffiths, T. 2024. Cognitive Architectures for Language Agents. *Transactions on Machine Learning Research*. Survey Certification.

Tamber. 2017. Steam Video Games. <https://www.kaggle.com/datasets/tamber/steam-video-games/data>.

Union, C. 2016. Anime Recommendations Database. <https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database>.

Vieira, I.; Allred, W.; Lankford, S.; Castilho, S.; and Way, A. 2024. How Much Data is Enough Data? Fine-Tuning Large Language Models for In-House Translation: Performance Evaluation Across Multiple Dataset Sizes. arXiv:2409.03454.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2024. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Transactions on Machine Learning Research*.

Weston, J.; Chopra, S.; and Bordes, A. 2015. Memory networks. Publisher Copyright: © 2015 International Conference on Learning Representations, ICLR. All rights reserved.; 3rd International Conference on Learning Representations, ICLR 2015 ; Conference date: 07-05-2015 Through 09-05-2015.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K. R.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*.

Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; and Wang, Y. 2024. MemoryBank: Enhancing Large Language Models with Long-Term Memory. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17): 19724–19731.

Ziegler, C.-N.; McNee, S. M.; Konstan, J. A.; and Lausen, G. 2005. Improving recommendation lists through topic diversification.

A Ethics Statement

This research on memory-augmented learning for large language model agents raises several important ethical considerations that we wish to acknowledge.

Though our suggestibility work was focused on how the model’s instruction-following ability varied with dataset, this kind of approach could also be used to more efficiently jail-break models to spread misinformation. Future work should be careful to avoid developing tools to improve the suggestibility of models to the point that they spread harmful misinformation.

We also recognize that improved adaptation capabilities may exacerbate existing biases in these agents. Because the insights are generated by the agent itself, even with feedback from the labeled data, it could cause the agent to reinforce

its preconceptions about the world, which may perpetuate harmful stereotypes. Future work should explore safeguards to identify and mitigate such bias amplification.

B Implementation Details

B.1 RAG Implementation

We used `blevlabs/stella_en_v5` as our encoder model and FAISS as our vector database. Similarity was based purely on the encodings of the questions in each dataset.

Though we did experiment with fine-tuning an encoder model to increase the separation of the classes in each dataset (for example, bought vs not-bought games for each Steam user) in embedding space, we did not see significant improvements in performance.

B.2 Dataset: Additional Details

NFCorpus: The original NFCorpus data source associated each article with many papers with varying degrees of separation, which we transformed into this pairwise setup by choosing one paper at the closest and furthest level of separation possible for each paper. Sampled 500 shortest combinations of articles and papers to avoid context-length issues.

Steam and Book Preference: Due to limitations in the number of games/books per user, the training and test sizes are smaller for these datasets than others. The train-test split percentage was maintained at 50%.

Steam User 1679: 104 samples in train set

Steam User 3188: 129 samples in train set

Steam User 6839: 116 samples in train set

Book User 63: 218 samples in train set

Book User 123: 183 samples in train set

Book User 2642: 206 samples in train set

B.3 Models

Default hyperparameters were used for all models, including a temperature of 0. OpenAI models were queried through the OpenAI API while open-source models were queried through the fireworks.ai API.

C Prompts

Critique Generation

User: {Question}

Agent: {PA Initial Prediction}

User: The correct answer is {Ground Truth Answer}. Explain why this is the correct answer, following the following JSON format

```
{
  correct_answer: correct_answer,
  local_reason: Specific reasons why this answer
is correct in this particular case.,
  global_reason: General reasons why this answer
is correct that can be applied to other questions.
}.
```

Respond only with JSON.

Semantic Memory Generation

Your job is to summarize a set of self-critiques made by some agent as they perform different instances of their task. For each instance you will be shown the output of the agent, followed by the critiques made by the agent after they were told the correct answer. Distill those critiques into a helpful summary of advice to the agent, paying particular attention to instances where the agent outputs an incorrect answer. Produce your output in a form that can be used directly as instructions to the agent. You should summarize the key points in these critiques. Be precise and concise. Do not repeat yourself.
For example in train_set:
{Question} {Answer} {Critique}

Performance Agent with Semantic Memory

{Question}
Here is some helpful advice that will help you make your decision: {Summary}

Performance Agent with Episodic Memory

For example in examples:
User: {Example Question}
Agent: {PA Initial Prediction}
User: {Critique Generation Prompt}
Agent: {Critique}
User: Here is your final question, make sure to learn from your past mistakes! {Question}

Performance Agent with Episodic and Semantic Memory

For example in examples:
User: {Example Question}
Agent: {PA Initial Prediction}
User: {Critique Generation Prompt}
Agent: {Critique}
User: Here is your final question, make sure to learn from your past mistakes! {Question}
Also, here is some additional advice to guide your response: {Summary}

D Examples

Example Critique Summary (NFCorpus)

1. ****Focus on Relevance****: Always choose the reference that directly relates to the subject matter of the article. Look for references that support the main claims made in the article.
2. ****Identify Key Themes****: Ensure that the reference paper closely aligns with the key themes discussed in the article, such as specific health effects, mechanisms of action, or relevant population studies.

3. ****Avoid General Topics****: Select references that do not deviate into unrelated topics. If one reference discusses foundational knowledge or statistics that do not support the article’s claims, it’s likely not the correct choice.

4. ****Highlight Specific Effects****: When discussing studies, emphasize specific effects or outcomes that are directly addressed in the article. Look for quantitative data or direct correlations that would affirm the article’s claims.

5. ****Example Comparison****: When there are multiple choices, conduct a clear comparison between them. If one reference explicitly discusses the same variables outlined in the article, that should be favored.

6. ****Review Findings****: When evaluating findings from referenced studies, ensure they corroborate the arguments or recommendations presented in the article. This can include discussing potential risks, benefits, or mechanisms.

7. ****Address Opinions and Recommendations****: When the article discusses guidelines or opinions (such as on health recommendations), favor references that critique or analyze these points directly.

8. ****Check for Clinical Relevance****: In clinical or scientific discussions, emphasize studies that provide empirical evidence that can be tied back to practical outcomes related to the topic of the article.

9. ****Nutritional Context****: In discussions around diet, ensure the references speak to the nutritional context being examined, such as the impact of specific foods on health, rather than unrelated dietary patterns.

10. ****Summarizing Connections****: When concluding which reference is correct, clearly summarize why the chosen reference aligns best with the article’s content. Discuss how it supports or expands upon the article’s key points.

By following these instructions, you will ensure that your references are relevant and provide strong support for the claims made in the articles you analyze.

E Additional Results

E.1 Varying K

In Table 5, we observe that varying K , the number of examples included from the episodic memory module, can have significant effects on the accuracy of the overall performance agent, with an up to 8% absolute difference in accuracy between different K values for the same setup. $K = 5$ and $K = 10$ both have similar results, and are on average better than $K = 1$ or $K = 3$. Because the results for $K = 10$ were not significantly higher than $K = 5$, we elected to use the simpler method and use $K = 5$ for all episodic and episodic+semantic memory experiments in this paper.

Model and Experiment	Multi-Cond. Ranking	NFCorpus	PubMed	Steam Pref	Book Pref	Anime Pref	Movie Pref
EP_CRIT							
$K = 1$	60.0	88.8	60.8	59.4	51.3	59.5	52.0
$K = 3$	64.0	83.6	60.4	60.2	52.3	58.3	54.8
$K = 5$	65.2	83.6	62.0	62.7	53.8	54.4	57.7
$K = 10$	56.8	86.8	61.6	62.8	52.9	59.2	60.0

Table 5: Effect on accuracy of varying K , the number of examples used in episodic memory, with gpt-4o-mini as the base model.

Experiment	Average Training Tokens	Average Utilization Tokens
EP_CRIT	142,517	492,716
SEM_CRIT	174,217	146,217
EP+SEM_CRIT	174,217	585,995

Table 6: The average number of tokens (input+output) it cost across all 7 datasets to train and utilize the different kinds of memory with gpt-4o-mini.

E.2 Token Costs

Table 6 lists the average number of input and output tokens required per dataset to populate the agent’s memory and to utilize that memory at prediction time. Episodic memory tends to be slightly less expensive than semantic memory in an offline memory generation setting, however it makes up for this in requiring many more tokens at utilization time. EP+SEM_CRIT, being a combination of the previous two methods, requires the most tokens.

E.3 Aggregated Results Across Datasets

Model	Average Accuracy Gain	Accuracy Gain Variance
PA and CA: gpt-4o-mini	3.0	16.6
PA and CA: o4-mini	1.1	3.9
PA: Llama 4 Scout		
CA: Llama 4 Scout	2.6	23.1
CA: gpt-4o-mini	4.2	62.0
CA: o4-mini	5.7	48.7
PA: Mixtral 8x22B		
CA: Mixtral 8x22B	3.3	37.4
CA: gpt-4o-mini	6.7	50.2
CA: o4-mini	-7.6	78.0
Llama 3.1 8B		
CA: Llama 3.1 8B	-1.6	31.7
CA: gpt-4o-mini	-0.4	28.1
CA: o4-mini	0.0	48.6

Table 7: The average gain in accuracy of the best-performing critique-based approach compared to the best-performing of the two baselines, zero_shot and EP_LABEL. Negative means the baselines did better. Results averaged across all seven datasets.