Toward Metacognitive Architectures for LLMs: Exposing Epistemic Fractures via Evolutionary Memory

Abstract

Platforms adopting Large Language Models (LLMs) continue to face challenges related to output reliability, long-term coherence, and epistemic traceability. While recent approaches emphasize scaling or retrieval augmentation, they often overlook a more structural limitation: language models can evaluate the form of their own outputs without possessing awareness of their epistemic validity.

This work introduces **MarCognity-AI**, an exploratory architectural framework designed to make this limitation explicit. The architecture separates linguistic generation from epistemic evaluation through a metacognitive cycle and a vector-based Evolutionary Memory system. Using semantic coherence metrics (e.g., CrossEncoders), the system reflects on its own outputs and archives these reflections as part of an evolving memory, without assuming access to truth or certainty.

Rather than proposing a definitive solution, this work highlights a structural tension that current LLM architectures cannot resolve through scale alone.

1. Introduction and Motivation

The large-scale implementation of LLMs in critical environments (e.g., research, finance, healthcare) is hindered by structural limitations. Current models do not guarantee verifiability, long-term coherence, or ethical adequacy. The dominant approach has focused on increasing model size, a method that does not resolve the intrinsic problems of grounding and self-criticism.

Our work introduces **MarCognity-AI**, an architectural framework that emulates human cognitive functions of **metacognition** (thinking about thinking) and **evolutionary memory** (learning based on the history of judgments). The goal is not simply to improve output, but to equip the agent with an autonomous cycle of critical reasoning.

The main contributions of this article are:

1. The formalization of a structured **Metacognitive Cycle** that employs specialized models (CrossEncoder, Specter) for internal validation.

2. An **Evolutionary Memory architecture** that integrates FAISS with the archiving of metacognitive reflection (Reflective Journal).
3. The integration of **Scientific Rigor and Ethical Control Modules** for the agent's internal governance.

2. Related Work

2.1 Retrieval-Augmented Generation (RAG) Standard RAG systems have improved reliability by anchoring generation to external data corpora. However, the limitations of RAG lie in its inability to critically validate the quality of the source and in the absence of a mechanism to correct reasoning errors after retrieval. **MarCognity-AI** extends RAG by implementing a post-generation semantic cross-verification phase.

2.2 Self-Correction and Reflection MarCognity-AI formalizes Self-Correction techniques into a two-level methodology: an objective numerical evaluation (via CrossEncoder) and a qualitative reflection which, once recorded, feeds into the long-term memory system. This ensures that self-improvement is based on a traceable and persistent mechanism.

3. Methodological Architecture of MarCognity-AI

3.1 Self-evaluation enables the agent to assess its outputs under explicit semantic and epistemic constraints, without assuming authority over their factual correctness.

3.1.1 Objective Semantic Evaluation The initial coherence between the query and the response is assessed using a CrossEncoder model (e.g., DeBERTa). This provides an objective similarity score. If this score falls below a critical threshold (0.7), the output is flagged for correction.

3.1.2 Critical Reflection and Structured Improvement In the event of coherence failure, the reflection agent is activated, triggering the improve_response function. The correction prompt is designed to focus the LLM on specific attributes.

Code Snippet (Structured Improvement):

```python
# Function to improve a response while preserving its content but enhancing quality and clarity
def improve_response(question, response, level):
    improvement_prompt = f"""
You produced the following response:
\"{response.strip()}\"

Question:
\"{question.strip()}\"

Requested level: {level}

Improve the response while preserving the original content by enhancing:
- Clarity
- Academic rigor
- Semantic coherence

Return only the improved version.
"""
    return llm.invoke(improvement_prompt.strip())
```

“The result of the reflection process is recorded in the **Reflective Journal.**

3.2 Vector-Based Evolutionary Memory System Memory is implemented as a high-performance FAISS index. We use the *allenai/specter* embedding model (768 dimensions), specialized in the representation of scientific texts.

The key innovation is the archiving process (via add_diary_to_memory), which includes **Metacognitive Reflection** and the **Coherence Score** within the vector. This enables the system, in a future prompt, to retrieve not only factual context but also the lessons learned from previous self-evaluations, thereby creating a memory that evolves according to its own critical history.

Code Snippet (Semantic Coherence):

```python
# === Semantic coherence check ===
def check_coherence(query, response):
    emb_query = embedding_model.encode([query])
    emb_response = embedding_model.encode([response])
    similarity = np.dot(emb_query, emb_response.T) / (np.linalg.norm(emb_query) * np.linalg.norm(emb_response))
    if similarity < 0.7:
        return "The response is too generic, reformulating with more precision..."
    return response
```

3.3 Scientific Rigor Module MarCognity-AI is equipped with modules to ensure the academic rigor of its content:

- **Methodology and Citation Verification**: The functions (verify_methodology, verify_citations) analyze the replicability of methodologies and the timeliness/relevance of cited sources.
- **Novelty and Impact Assessment**: A RandomForestRegressor estimates an *Impact Score* for articles. Vector similarity among hypotheses further evaluates the degree of novelty with respect to the existing knowledge corpus.

3.4 Ethical Control and Controlled Autonomy An Ethical Module with a flagging system analyzes the text for the presence of risk patterns related to "critical topics." In cases of high-risk detection, the system can reduce the agent's autonomy, ensuring compliance.

3.5 The Epistemic Fracture in Large Language Models

Large Language Models exhibit a structural limitation that cannot be resolved through scale alone:

they generate *linguistically coherent* answers without possessing *epistemic awareness* of their truth value.

This phenomenon, which we define as the **Epistemic Fracture**, represents the core mismatch between the statistical nature of LLMs and the epistemic demands of high-stakes reasoning.

At its foundation, the Epistemic Fracture arises from three intrinsic properties of autoregressive models:

- **Token-level optimization rather than truth-level optimization**
- LLMs maximize next-token probability, not factual accuracy or logical validity.
- **Absence of internal truth states**
- The model does not maintain representations of certainty, justification, or evidence.
- **Coherence without grounding**
- Linguistic fluency can mask unsupported or fabricated content, producing hallucinations that appear credible.

As a result, LLMs often generate statements that are:

- syntactically correct
- semantically plausible

- rhetorically confident

yet **epistemically ungrounded**.

This divergence between *how convincing a statement sounds* and *how justified it is* constitutes the Epistemic Fracture.

It is the primary obstacle to deploying LLMs in domains requiring verifiability, accountability, and long-term coherence.

MarCognity-AI addresses this limitation by introducing explicit epistemic governance mechanisms—most notably the **Skeptical Agent** and the **Evolutionary Memory**—which transform generation from a purely linguistic process into a verifiable and traceable cognitive workflow.

3.6 The Skeptical Agent: Making Epistemic Gaps Explicit

The Skeptical Agent is a MarCognity-AI module designed to expose the gap between linguistic coherence and epistemic support in LLM outputs.

Rather than improving fluency or stylistic quality, its role is to make unsupported claims visible, forcing a distinction between what a model can say coherently and what it can actually justify.

The agent operates through a deliberately strict procedure:

1. Sentence-level decomposition
The generated response is decomposed into discrete statements, each treated as an independent epistemic unit rather than as part of a rhetorically coherent whole**.**

2. Comparison against provided sources
Each statement is checked against the documents designated as the source of truth.
The evaluation criterion is intentionally binary and non-interpretative:

- if explicit or strongly implied support exists → **VERIFIED**
- if no support exists → **EPISTEMIC FAILURE**

3. Explicit identification of unsupported claims
Any statement that cannot be justified by the documents is flagged as an epistemic failure, making hallucinations and unsupported inferences explicit rather than stylistically smoothed over.

4. Stylistic neutrality
The agent does not assess clarity, elegance, or rhetorical quality.
Its sole function is epistemic traceability.

5. Handling the absence of sources
When no documents are provided, the agent declares the entire response to be in a state of **Epistemic Uncertainty**, since no statement can be meaningfully verified.

6. Structured output
The analysis produces a formal report of the form:

- **CLAIM**: “statement”
- **STATUS**: VERIFIED / EPISTEMIC FAILURE
- **REASON**: justification based on the available sources

Rather than resolving epistemic uncertainty, the Skeptical Agent **makes its presence unavoidable**.
It provides the Epistemic Supervisor with a concrete basis for deciding whether a response should be accepted, revised, or regenerated, while preserving the distinction between linguistic performance and epistemic responsibility.

4. Performance Analysis and Evaluation

This section describes the evaluation methodology of **MarCognity-AI**, the set of integrated models, and the results obtained that quantify performance improvements in terms of reliability, coherence, and ethical rigor compared to a baseline non-augmented LLM.

4.1 Experimental Setup and Model Architecture The **MarCognity-AI** architecture operates by integrating a suite of specialized models for cognitive augmentation and validation. The primary LLM model for generation is *meta-llama/llama-4-maverick-17b-128e-instruct*, used under the **LLaMA 4 Community License (Meta)**. This setup ensures that the agent does not merely generate output but engages in an integrated metacognitive reasoning process for each response.

The integration of the Skeptical Agent completes the internal governance layer of MarCognity-AI. The following section evaluates how these components collectively improve reliability, coherence, and epistemic transparency.

The key tools employed are:

- **LLM Generation**: Llama 4 (*meta-llama/llama-4-maverick-17b-128e-instruct*).
- **Vector Embedding**: *allenai/specter* model (768 dimensions), specialized in the representation of scientific texts.
- **Objective Semantic Evaluation**: A CrossEncoder (DeBERTa-based) assesses coherence between query and response, triggering correction if the score falls below the critical threshold of 0.7.
- **Evolutionary Memory**: High-performance FAISS index for archiving and retrieving metacognitive reflection.
- **Scientific Retrieval**: The agent leverages asynchronous querying of open-access sources for grounding and validation, including *arXiv*, *PubMed*, and *Zenodo*.

4.2 Evaluation Metrics and Adapted Methodology The evaluation was conducted to measure the impact of the **Metacognitive Cycle** and **Evolutionary Memory** on the intrinsic limitations of the LLM.

- **Increase in Semantic Coherence Score**: Measured as the improvement in the CrossEncoder score between the initial response generated by Llama 4 and the final response refined by the reflection agent.
- **Reduction of Fallacy Rate (Hallucination Rate)**: Measured by the number of incorrect statements corrected after the self-improvement phase. Statements were verified against the *Corpus of Truth* composed of articles retrieved from open-access sources (arXiv, PubMed, Zenodo).
- **Long-Term Coherence (LTC)**: Assessed by measuring the relevance and accuracy of responses in multi-turn sessions, using the Reflective Journal as memory to guide learning based on the history of judgments.
- **Rigor and Transparency**: Qualitative evaluation of the integration of conceptual visualizations (generated via Plotly, NetworkX, etc.) and the effectiveness of ethical auditing (bias detection, risk analysis) on each output,ensuring that the agent's outputs remain epistemically accountable and subject to explicit verification constraints.

5. Implications and Open Directions

The MarCognity-AI architecture should be interpreted as an exploratory framework for examining the structural limitations of current LLM-based systems rather than as a finalized solution. Its primary contribution lies in making epistemic constraints explicit within the

generation process and in highlighting the tension between linguistic coherence and epistemic justification.

From an applied perspective, the introduction of explicit evaluation signals (e.g., coherence scores, epistemic failure flags) suggests potential directions for improving transparency and reliability in high-stakes deployments. However, these mechanisms are not presented as guarantees, but as tools for exposing where and how current models fail to meet epistemic requirements.

Similarly, the use of vector-based Evolutionary Memory illustrates one possible approach to preserving the history of internal evaluations and judgments over time. This design choice opens questions about how long-term memory, reflection, and governance might interact in future architectures, rather than claiming to resolve scalability or cost constraints.

Overall, MarCognity-AI does not aim to define mature cognitive agents, but to provide a concrete setting in which epistemic limitations become observable, measurable, and subject to systematic investigation.

References

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

Gao, J., Yang, Z., Chen, X., Li, X., Wu, C., & Wang, H. (2023). Retrieval-Augmented Generation (RAG) for LLMs: A Comprehensive Survey. ArXiv:2312.10997.

ohnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. IEEE Transactions on Parallel and Distributed Systems, 30(12), 3042-3054.

Cohan, A., Ammar, W., Groeneveld, D., & Pierce, D. (2020). SPECTER: Document-level Representation Learning for Scientific Documents. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

Geva, M., Saad, N., & Schuster, M. (2023). Self-Correction in LLMs: A Survey. ArXiv:2308.05583.