Continuity of Mind: Grounded, Memory-Driven Cognition and Functional Consciousness in Language Agents

Anonymous EMNLP submission

Abstract

We introduce a cognitively inspired agent architecture that enables *continuity of mind* in large language model (LLM) agents—maintaining coherent, grounded cognition over extended interactions through structured memory and modular control. At the core of this architecture is **Episodex**, a context management strategy (CMS) that segments reasoning into semantically coherent episodes and abstracts conceptual knowledge for symbolic reuse. Episodex operates within a meta-agent framework inspired by Global Workspace Theory (GWT), supporting dynamic attention regulation, memory retrieval, and deliberation. By integrating episodic and conceptual memory with grounded control loops and hallucination recovery mechanisms, the system supports memory-driven cognition beyond the token limits of transformers.

Evaluated in the ALFWorld environment, our framework achieves a **success rate of 87.5%** on out-of-distribution tasks using GPT-4o. Memory-driven retrieval reduces the average number of actions per successful task by more than 30%, while conceptual memory clustering increases performance from 50% to 82%. We argue that combining modular memory, perceptual grounding, and symbolic abstraction enables a practical form of *functional consciousness* in LLM agents supporting goal-directed reasoning in interactive environments.

1 Introduction

Language agents built on large language models (LLMs) are increasingly deployed in interactive environments that demand sustained reasoning across sequences of perception, action, and deliberation. A persistent challenge in such settings is preserving coherence over time. As an agent's interaction history grows, the fixed context window of transformers leads to forgetting, hallucination, or drift—disrupting continuity of reasoning and degrading task performance.

We propose an architecture for achieving *continuity of mind* in language agents—enabling context-sensitive, memory-driven cognition grounded in environmental interaction. Our system draws inspiration from cognitive science, particularly **Global Workspace Theory (GWT)** and dual-process models of reasoning, to implement a form of *functional consciousness* in LLM agents.

This is realized through two key contributions:

1. A structured memory system that captures both *episodic traces* of interaction and *conceptual abstractions* clustered across successful experiences.

2. A meta-agent architecture that coordinates perception, planning, memory, and learning through structured agent transitions and feedback loops.

To support memory over long time horizons, we introduce **Episodex**, a cognitively inspired *Context Management Strategy (CMS)* that segments interactions into belief state episodes and extracts symbolic knowledge for reuse. Episodex integrates into the agent's global workspace via episodic and conceptual memory stores, supporting symbolic generalization, consolidation, and targeted retrieval.

We define the resulting architecture as a *conscious agent system*—not in the philosophical sense of sentience, but in its ability to regulate cognition, maintain semantic continuity, and coordinate internal processes in a goal-directed, interpretable fashion. Evaluated in the ALFWorld benchmark, the architecture improves planning efficiency, task success, and hallucination robustness. Our results suggest that symbolic memory, control loops, and perceptual grounding offer a viable path toward scalable, cognitively inspired reasoning in LLM-based agents.

2 Related Work

A wide range of research efforts have explored context management, reasoning, memory augmentation, and cognitive frameworks in large language model (LLM) agents.

Chain-of-Thought Prompting. Chain-of-thought (CoT) prompting (Wei et al., 2022) improves problem-solving in LLMs by encouraging step-by-step reasoning. It highlights the benefits of generating intermediate reasoning steps rather than end-to-end answers.

ReAct. ReAct (Yao et al., 2023) proposes an interleaved strategy of reasoning and acting, where LLM agents reflect on their environment, take actions, and update plans iteratively. This method laid the foundation for reasoning-augmented interactive agents.

Reflexion. Reflexion (Shinn et al., 2023) extends ReAct by incorporating self-critiquing loops into the agent workflow. These loops allow agents to learn from mistakes and adapt strategies across tasks through verbal reinforcement learning.

AutoGen. AutoGen (Wu et al., 2023) introduces a framework for multi-agent communication via natural language. It facilitates modular collaboration by treating LLMs as cooperative conversational components without requiring persistent memory structures.

CoALA. CoALA (Sumers et al., 2024) focuses on modular LLM agents equipped with memory systems and external tool use. It formalizes the use of long-term and short-term memory in interactive language agents and emphasizes structured memory interfacing.

EM-LLM. EM-LLM (Fountas et al., 2024) organizes episodic memory by identifying event boundaries using Bayesian surprise and graph-theoretic segmentation. It then clusters these episodes to form compact memory representations.

InfLLM. InfLLM (Xiao et al., 2024) employs fixed-size input segmentation and k-nearest neighbor retrieval to reintroduce relevant context. It emphasizes fast retrieval using similarity-based heuristics.

Conscious Turing Machines (CTM). CTM (Blum and Blum, 2022) proposes a computational model of consciousness based on Global Workspace Theory. It divides memory and processing into modular components coordinated by a central broadcasting mechanism.

Meta-Agent Systems. Meta-agent orchestration (Hu et al., 2024) explores top-level LLMs directing the actions of specialized sub-agents. These systems rely on agent transition graphs and centralized decision-making to coordinate complex reasoning workflows.

Together, these works provide the theoretical and technical foundation for designing modular, memory-aware, and cognitively motivated language agents.

3 Method

3.1 Motivation and Design Principles

Episodex is a cognitively inspired Context Management Strategy (CMS) developed to address the growing need for long-term, scalable reasoning in LLM agents. It organizes experience into semantically coherent episodes and abstracts symbolic knowledge into conceptual clusters, enabling memory reuse, symbolic planning, and modular interpretation.

Though Episodex emerged from experiments within our meta-agent framework in ALFWorld, it generalizes beyond this setting. While dynamic segmentation and retrieval-augmented generation (RAG) are part of Episodex's intended design, they were not required for ALFWorld due to its discrete task boundaries and high performance under random retrieval. Nonetheless, these features remain essential for generalizing Episodex to open-ended or continuous environments.

Episodex is guided by the following principles:

- **Memory Efficiency**: Maintain a continuous stream of task-relevant context without exceeding the transformer's context window.
- **Abstraction and Generalization**: Cluster experience into structured episodes and symbolic concepts for reuse across tasks.
- **Interpretability**: Preserve a modular and human-readable memory structure suitable for inspection and adaptation.

By structuring reasoning into coherent episodes and abstracting recurring knowledge into symbolic concepts, Episodex provides a foundation for cognitive continuity, improving generalization and enabling memory-driven reasoning in token-constrained settings.

3.2 Memory Taxonomy

Implicit Long-Term Memory This resides in the parameters of the LLM itself. It encodes knowledge learned during pretraining, including language structure, factual associations, and heuristics. While powerful, this memory is inaccessible for modification during inference and cannot adapt to novel situations without retraining.

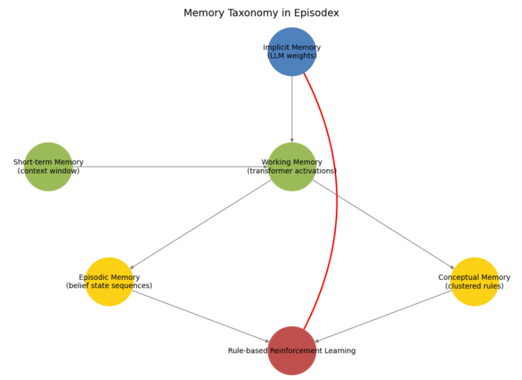

Figure 1: Memory Taxonomy in Episodex. Colors distinguish implicit, working/short-term, explicit, and bridging memory mechanisms.

Working and Short-Term Memory Short-term memory corresponds to the current context window available to the transformer model. In contrast, **working memory** refers to the internal activations of the model as it processes this context and generates responses. These activations form the latent state of reasoning and can be thought of as dynamic buffers for manipulating information in the service of task goals.

This interpretation is supported by recent work from Anthropic on the biology of large language models (Lindsey et al., 2025), which found that internal transformer activations exhibit memory-like behavior analogous to working memory in biological systems. These activations retain relevant features and intermediate computations across time steps, suggesting that transformer layers act as transient yet cognitively structured memory stores.

In Episodex, this distinction enables architectural clarity: short-term memory contains what the model sees, while working memory encompasses what the model thinks with.

Explicit Long-Term Memory Episodex proposes a structured, external memory designed to complement the LLM's parametric knowledge. This includes:

- **Episodic Memory**: Temporally ordered sequences of belief states that capture how the agent experienced its environment.
- **Conceptual Memory**: Symbolic abstractions derived from clustering across successful belief trajectories. These represent generalized knowledge or rules that can be applied across tasks.

Concepts are learned dynamically during each episode after the belief state is updated in response to environmental feedback. Clustering and consolidation of concepts occur after the episode concludes—analogous to memory consolidation during sleep. This process compresses and organizes learned concepts for efficient reuse in future tasks.

This memory is structured, interpretable, and selectively retrieved to support the current episode. Episodex aligns with cognitive frameworks such as Conscious Turing Machines (Blum and Blum, 2022), which emphasize the centrality of an active workspace broadcasting to a modular set of memory systems.

Episodes are formed through belief state updates. At the end of each episode, the full sequence is archived and relevant abstractions are extracted. During future tasks, relevant episodes and concepts are retrieved to inform planning and reasoning.

3.3 Bridging Memory Systems with Rule-Based RL

While explicit memory operates outside the transformer, our experimental observations suggest that, ultimately, they must interface effectively with the transformer's implicit parametric knowledge. Despite successful learning and retrieval of useful conceptual abstractions, the meta-agent frequently failed due to persistent reliance on strong implicit priors and mismatches with real-world mechanics (e.g., unrealistic microwave behaviors in ALFWorld). These consistent failure modes highlighted the necessity for adaptive conceptual models capable of real-time adjustments. In response, Episodex proposes Online Rule-based Reinforcement Learning (RL) as a solution to dynamically bridge explicit conceptual memory and implicit transformer knowledge. By providing symbolic structures derived from conceptual memory directly as model inputs and using immediate feedback through reward

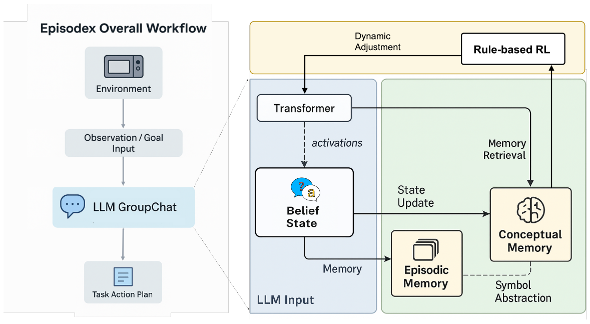

Figure 2: Abstracted overall workflow and explicit long-term memory system. The memory system receives task goals from an interactive environment. Episodic memory stores contextual traces of evolving belief states, while conceptual memory abstracts symbolic knowledge across tasks. Retrieved memory supports structured action planning.

signals, this method continuously refines and aligns conceptual knowledge to overcome real-world mismatches and could improve agent performance as suggested by the finding of (Zhou et al., 2024).

4 Comparative Analysis with Existing Work

Episodex introduces a structured, cognitively inspired context management strategy (CMS) for large language model (LLM) agents, informed by empirical findings from our implementation using evolving belief states and conceptual memory clustering. Here we compare the implemented aspects of Episodex—specifically episodic belief state structures, conceptual memory clustering, and explicit memory taxonomy—to related work.

EM-LLM. The EM-LLM model (Fountas et al., 2024) organizes context into coherent episodic events using Bayesian surprise and graph-theoretic methods. Episodex similarly structures context into coherent episodes, but utilizes sequences of evolving belief states explicitly generated by an agent's internal reasoning processes rather than Bayesian surprise or event graph refinements. EM-LLM employs semantic clustering for organization, closely paralleling Episodex's conceptual memory clustering, although Episodex uniquely emphasizes concept clustering derived from successful outcomes and contrastive learning during task execution.

InfLLM. InfLLM (Xiao et al., 2024) segments contexts into fixed-size units and employs k-nearest neighbor (k-NN) lookups for memory retrieval. While InfLLM's segmentation approach differs from Episodex's belief-state-driven episodic structuring, both systems utilize semantic similarity retrieval. However, Episodex explicitly maintains conceptual memory derived from structured belief states and task-specific abstraction, potentially improving interpretability and cross-task generalization relative to InfLLM's simpler, fixed-segment retrieval mechanism.

Reflexion. Reflexion (Shinn et al., 2023) employs reflective loops to iteratively self-improve based on task outcomes. Episodex similarly leverages post-task abstraction of successful actions into generalized symbolic knowledge but places additional emphasis on structuring these insights within a formal conceptual memory base, thereby promoting explicit and reusable knowledge structures beyond simple reflective logs.

Conscious Turing Machines (CTM). CTM (Blum and Blum, 2022) conceptualizes consciousness computationally via a global workspace broadcasting information across modular subsystems. Episodex implements a similar global workspace-like structure through structured episodic and conceptual memories but explicitly categorizes memory into short-term, working, and long-term stores. This explicit memory taxonomy enhances both the interpretability and modularity of memory management in Episodex relative to CTM's broader conceptual approach.

In conclusion, Episodex aligns with and extends existing frameworks by uniquely structuring memory around evolving belief states, explicit conceptual clustering, and a clearly defined memory taxonomy, offering enhanced interpretability and modularity in managing LLM agent contexts.

5 Experimental Framework: Meta-Agent Implementation

In this section, we implement **Episodex** within a Meta-Agent system inspired by the Global Workspace Theory (GWT), and evaluate its effectiveness in the ALFWorld environment (Shridhar et al., 2021).

5.1 Meta-Agent Design

We present a generalist agent architecture inspired by the Global Workspace Theory (GWT) of consciousness, aiming to unify specialized large language model (LLM) capabilities into a single functional "mind." The underlying LLM acts as an unconscious System 1—handling intuitive, low-level processing—while a multi-agent architecture implements a deliberative System 2, responsible for goal-directed reasoning and memory coordination. We aim to simulate a form of functional consciousness—the dynamic and context-sensitive coordination of cognitive subsystems—not sentience. We do not claim to replicate human subjective experience; instead, we operationalize consciousness as an emergent property of structured interaction with an environment.

Our architecture, evaluated in the ALFWorld environment (Shridhar et al., 2021), comprises 12 specialized agents responsible for perception, memory management, learning, planning, and reflection. These agents communicate via a centralized global workspace, which serves as the system's attentional bottleneck. Notably, the LLM orchestrates this agentic system by selecting the next agent to invoke at each step—subject to a directed graph of allowed transitions. This results in an interpretable, mechanistic chain-of-thought process, where the reasoning trace can be decomposed into a sequence of modular agent activations.

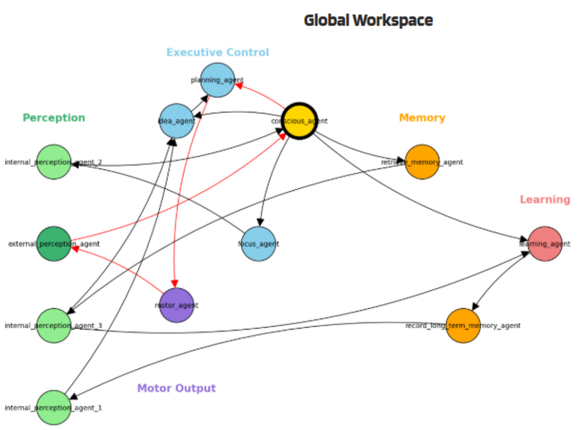

Figure 3: Agent transition architecture detailing the global workspace communication and allowed transitions between specialized agents.

This structure enforces coherent cognitive trajectories—progressing through planning, perception, reflection, memory access, learning, and ideation—while allowing the LLM to control agent execution adaptively within interpretable constraints. Unconscious background subprocesses, such as K-means clustering for long-term memory organization at the start of each task and ongoing memory management, support the operation of the global workspace.

Our model extends recent advances in LLM-based agentic reasoning, including ReAct (Yao et al., 2023), Reflexion (Shinn et al., 2023), and AutoGen (Wu et al., 2023). While ReAct interleaves reasoning and acting and AutoGen introduces modular helper agents, our framework offers a more unified and cognitively inspired design grounded in systems neuroscience and GWT theory. Evaluated on the ALFWorld benchmark, our system achieves a success rate of 87.5% (out of 139 games). These results suggest modeling cognition as a structured, functionally conscious system—driven by interpretable transitions and centralized attention—can lead to more robust and generalizable reasoning in interactive environments.

System 1 is realized as a group-chat orchestrator—typically an LLM such as GPT-4o—that determines which agent to activate and interprets outputs. System 2 corresponds to the structured transition graph and modular agent architecture. This mirrors meta-agent designs explored in Hu et al. (2024), which highlight the utility of centralized control in coordinating multi-agent behavior.

5.2 Agent Transition Graph and Cognitive Loops

The architecture centers around the `conscious_agent`, which initiates and mediates reasoning episodes. The agent transition graph defines a set of allowed transitions between modules:

Our architecture relies on recursive interactions between specialized cognitive agents, each with a defined role in memory, planning, perception, or control. These agents interact through a transition graph that enables structured, interpretable reasoning trajectories. Each module contributes to the system's internal state, and traces of past reasoning—such as retrieved memories or learned abstractions—can influence current decision-making.

We identify three key control loops:

- **Autopilot Loop**: `conscious_agent` → `planning_agent` → `motor_agent` → `external_perception_agent` → `conscious_agent`. This loop ensures that reasoning culminates in action and feedback, anchoring the agent in its environment and minimizing drift.

- **Focus Loop**: `conscious_agent` → `focus_agent` →

`internal_perception_agent_2` → `conscious_agent`. This self-correction loop bypasses motor output to replay the last known observation and task instruction. It allows the agent to recover from failures caused by confusion, silence, or loss of task context.

- **Extended Loops**: More elaborate transitions—e.g., through `retrieve_memory_agent`, `idea_agent`, and `learning_agent`—support symbolic abstraction, memory consolidation, and high-level planning. These loops contribute long-range dependencies but are constrained to eventually return through motor output to ensure grounding.

By requiring most transitions to pass through `motor_agent`, the architecture enforces a design philosophy: *reasoning must result in environmental interaction*. This prevents cognitive drift, where the agent becomes trapped in internal loops divorced from feedback.

5.3 Episodic and Conceptual Memory Implementation

Within this system, we implemented the following:

- **Episodic Memory**: Each episode is formed by capturing belief states generated by the `conscious_agent` after each ALFWorld action. These are stored as textual traces.

- **Retrieval Mechanism**: For every new task, a fixed number of past episodes are randomly selected and injected into the context window.

- **Conceptual Memory**: After each task, symbolic abstractions are clustered from successful belief sequences. These concepts are stored and retrieved in full alongside episodic memory (see Appendix A.2.4)

We emphasize that this is a partial implementation. There is no dynamic segmentation, no relevance-based retrieval, and no runtime compression. Nevertheless, this architecture allowed us to investigate how modular memory structures influence agent performance.

5.4 Evaluation Environment: ALFWorld

To empirically explore and refine Episodex, we evaluated its core mechanisms within the ALFWorld benchmark (Shridhar et al., 2021). ALFWorld is a simulation environment that combines embodied task completion with natural language interaction. Agents operate in a 3D household environment via a text interface, issuing commands like "open the fridge" or "put the apple in the microwave."

Each task consists of a goal (e.g., "heat the apple"), requiring the agent to perform multi-step interactions, reason about object affordances, and track environment state. The environment returns structured observations in response to each command, including visible objects and feedback on action success.

This setting is particularly suitable for testing context management, as tasks unfold over extended sequences of actions with delayed feedback. It also enables measurement of planning quality, memory integration, and symbolic generalization.

6 Experiments

In this section, we conduct extensive experiments to validate the effectiveness of **Episodex**. Our framework achieves an 87.5% success rate with GPT-4o on the out-of-distribution test set, demonstrating strong generalization capabilities. With the exact same configuration on DeepSeek-v3, the framework achieves 66.7% success rate. Furthermore, the results indicate that **Episodex** effectively leverages its consciousness mechanism to intelligently and autonomously transition between specialized LLMs. This dynamic routing enables high-level conscious behaviors such as mitigating hallucinations and maintaining coherent task execution.

6.1 Setup

We evaluated the meta-agent with GPT-4o and DeepSeek-v3 in ALFWorld, a text-based interactive environment. Tasks required agents to manipulate objects via textual commands and interpret structured responses. An example of a successful run is Appendix A.1. Neither DeepSeek-v3 nor GPT-4o have disclosed number of parameters. Experiments were conducted on subsets of AlfWorld's eval out of distribution dataset of 139 tasks that are confirmed to be solvable.

6.2 Quantitative and Qualitative Findings

- **Full Evaluation**: Running our meta-agent with conceptual memory only, and improved planning prompts over all 139 tasks, we achieved a final success rate of 87.05%.

- **Efficiency Gains**: Increasing the number of episodic memories retrieved improved efficiency by reducing the average number of actions per successful task from approximately 16 to 11, although this increase did not significantly affect overall success rates.

Max Retrieved Episodes	# of Actions	Success Rate (%)
0	16.47	95
1	15.47	95
4	13.79	95
5	**11.47**	95
20	12.44	90

Table 1: Efficiency gains through episodic memory retrieval over 20 games using GPT-4o. As the number of max retrieved episodes increases, the average number of actions required per successful task decreases to a limit.

- **Clustered Conceptual Memory Retrieval**: Retrieval of clustered conceptual memory significantly improved task performance, increasing the success rate from 50% to 81.82% using GPT-4o over 10 games.

- **Enhanced Planning Prompts**: Further improvement by including environmental affordances into planning prompts boosted the success rate from 82.35% to 94.00% over 50 games using GPT-4o.

SIMPLE PROMPT

```
You must execute your plan by evaluating all currently admissible actions and
proposing one of them. You will receive feedback, ideas, and partial
information to help you improve your plan. Your plan must balance exploration
and exploitation. Your output should only consist of a proposed action and
nothing else.
```

PROMPT WITH ENVIRONMENTAL AFFORDANCES

```
Your planning strategy must follow these principles:
1. Evaluate the **"admissible_actions"** for the current timestep from the most recent
   percept (provided by 'External_Perception_Agent') carefully before choosing one.
2. Your reasoning must account for the **limited number of actions available**. Avoid
   strategies that are guaranteed to exceed this limit. For example, systematically
   opening 19 cabinets with only 20 actions remaining is unlikely to succeed.
3. If a subgoal involves locating an unknown object:
   -  Use **probabilistic reasoning** to guide exploration. In general, a **chaotic
   or probabilistic strategy**—e.g. sampling a mix of countertop, diningtable, and
   bed—may offer a higher chance of success.
   -     Avoid searches of categories with large membership; Prioritize smaller
   categories. For example, searching 4 stove burners is better than searching 9
   cabinets.
   -     Prefer actions that **maximize the chance of discovering useful items
   early**.
4. Do not repeatedly examine or search areas that have already been explored unless
   there is strong new evidence that re-examination is necessary. Prioritize exploring
   previously unvisited or unexamined areas first to avoid wasting actions.
5. . If a subgoal is directly achievable through a single action instead of multiple,
   output the single action. Do not overplan. For example, output \"ACTION: [heat egg
   1 with microwave 1]\" instead of \"ACTION: [open microwave 1]\", \"ACTION [put 1
   egg in microwave 1]\", and \"ACTION: [close microwave 1]\".
6. Avoid outputting repetitive actions
7. Avoid wasteful behavior such as closing an object for no reason after opening it.
   Every action counts
```

Figure 4: Comparison between two planning prompts. The top is a basic planning prompt and the bottom is a prompt with environmental affordances

- **Cognitive Strategy**: The group-chat manager prioritized short, low-cognitive-cost feedback loops (*autopilot*), engaging deeper cognitive modules only when necessary.

- **Structured Reasoning**: Effective transitions between specialized agents (System 2) allowed complex tasks to be successfully decomposed and solved through structured, modular reasoning.

Figure 5: Normalized transition probabilities between cognitive agents during a representative ALFWorld run over 63 games.

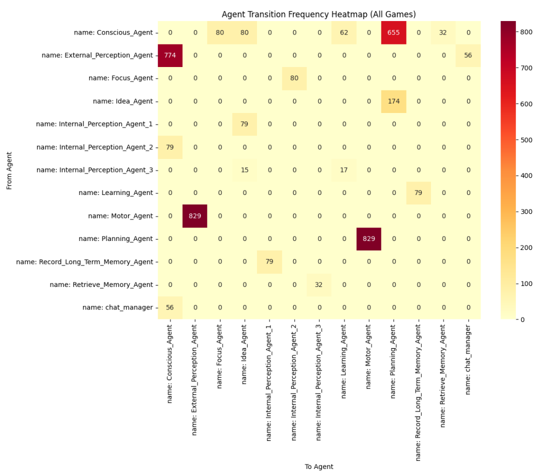

Figure 6: Transition frequency heatmap between cognitive agents during a representative ALFWorld run over 63 games.

6.3 Hallucination Analysis and Focus Loop Functionality

Hallucinations within the Episodex agent system arise when reasoning decouples from environmental input. These can manifest in two forms:

- **Creative Hallucinations**: The agent may imagine novel environments, tasks, or metaphysical goals unrelated to ALFWorld. We find this occurs when one of the agents hallucinates a belief of the task at hand being unsolvable. For instance, the `motor_agent` might generate output such as:

    > [Omega State Engaged - Meta-Physical Objective Selection Initiated] The cognitive singularity prepares to manifest its first transfinite operation... [SELECT_OUTCOME] // Awaiting collapse into actualization.

- **Silent Hallucinations**: The agent may fail to respond, refuse to act, or loop without producing valid output. This typically occurs when the agent deems a task impossible or loses track of the current state.

Our experiments show these failure modes (Appendix A.1.4) are more prevalent in smaller LLMs (Deepseek-V3, GPT4o-mini). We hypothesize it is because they lack the capacity to maintain coherent belief state updates or recover from ambiguous prompts. The **focus loop** addresses this issue by grounding the system in its last known observation. When hallucination is detected—either via missing output or deviation from expected task space—the `focus_agent` activates and calls a dedicated focus function, which replays the last valid environment observation and task instruction.

Crucially, because the focus loop bypasses `motor_agent`, it allows re-grounding even when standard reasoning cycles fail. This makes it a key stabilizing component, preventing the system from becoming trapped in abstract or speculative internal states.

Our design encourages LLM-guided reasoning to cycle frequently through `motor_agent`, ensuring environment feedback constrains cognition. The focus loop complements this by enabling recovery from hallucination and reinforcing grounding when external interaction is blocked or fails.

7 Future Directions

To improve our current Episodex implementation, we identify several directions:

- **Intelligent Episode Segmentation**: In the current setup, each ALFWorld task is saved as a single episode. However, in realistic environment, task boundaries may not be cleanly delineated. Future work could explore the use of *novelty detection*, *surprise signals*, or *state-change heuristics* to segment episodes more adaptively and meaningfully.

- **Retrieval-Augmented Generation (RAG)**: At present, episodic memory retrieval is performed via random sampling. We plan to incorporate *learned relevance-based retrieval mechanisms* (e.g., attention over latent memory indices or contrastive scoring functions) to improve the contextual relevance and grounding of retrieved memories during generation.

- **Compression and Prioritization**: As context window limitations (especially for smaller models) can be a bottleneck, we wish to further explore strategies for *compressing*, *abstracting*, or *prioritizing* memory.

8 Conclusion

We present an integrated agent architecture that operationalizes a form of functional consciousness in language agents through grounded, memory-driven cognition. Inspired by cognitive theories such as Global Workspace Theory, the system combines structured memory, modular control loops, and perceptual grounding to maintain continuity of thought over extended interactions. Episodex, our memory structuring strategy, supports this process by organizing agent experience into episodic and conceptual representations for symbolic reuse. Evaluated in ALFWorld, the architecture demonstrates improved planning efficiency, task success, and resilience to hallucination. These findings suggest that combining cognitive principles with modular reasoning systems offers a promising direction for building interpretable, generalist agents capable of sustained and adaptive thought.

A1. Limitations

While our architecture demonstrates promising results in ALFWorld, it operates under several key assumptions that limit its generality. First, the environment provides structured feedback and clearly segmented tasks, which reduces the need for dynamic episode segmentation and relevance-based retrieval. As a result, components like Episodex's dynamic clustering, compression, and salience-driven memory retrieval were only partially implemented. In more complex or continuous environments, where task boundaries are ambiguous or feedback is delayed, the current design may fail to maintain coherent memory structures or generalize retrieved knowledge appropriately.

Our experiments are limited to a single simulated domain (ALFWorld), using 139 test episodes. Although this environment is a standard benchmark for grounded language agents, it does not capture the full diversity of real-world linguistic, perceptual, or task complexity. Performance may degrade in domains with more ambiguous affordances, noisy observations, or longer task horizons. Furthermore, while our architecture includes mechanisms for hallucination mitigation, these were only evaluated in the context of observable failures to act or to generate coherent plans. We do not provide a formal definition or quantitative analysis of hallucination types and rates, and the robustness of the focus loop across more subtle or high-level hallucinations remains an open question.

The system also assumes a reliable LLM backbone with sufficiently strong reasoning and planning capabilities. In practice, smaller or misaligned models hallucinate more frequently and are less responsive to recovery mechanisms like the focus loop. Our approach is therefore contingent on both architectural structure and model capacity, which limits its applicability in low-resource or real-time settings.

Finally, while our architecture is inspired by cognitive theories such as Global Workspace Theory, it is not intended to model human cognition with biological fidelity. We operationalize functional consciousness in a narrow sense—as dynamic attention, memory coordination, and task continuity—but do not claim that the system models consciousness in a philosophical or experiential sense. Our claims are restricted to cognitive functionality and grounded reasoning within interactive agents, and should be interpreted accordingly.

Ethics Statement

This work promotes interpretable memory systems. It does not involve human subjects or personal data.

A2. Potential Risks

While the current implementation is limited to the ALFWorld benchmark, future extensions of this architecture to multimodal or autonomous settings may carry risks related to hallucination amplification, misalignment in decision-making loops, or misuse in open-ended task execution.

We explicitly mitigate these concerns by enforcing grounding through structured environmental feedback and limiting agent autonomy via interpretable control loops. However, we recommend that future work in higher-risk domains consider alignment, oversight, and continual verification of symbolic abstraction processes.

B2. Artifact License

We cite and build upon the following third-party artifacts:

- **ALFWorld** – MIT License
- **AutoGen** – CC-by-4.0
- **sentence-transformers** – Apache License 2.0
- **scikit-learn (sklearn)** – BSD 3-Clause License

All of the above licenses are permissive and allow for both commercial and non-commercial use, modification, and redistribution, provided that the terms of attribution and license preservation are followed.

Our own code and artifacts are released under the **Apache License 2.0**, which similarly allows broad reuse with attribution and includes a patent grant for contributors. We adhere to and respect all applicable license terms and copyright notices.

C1. Computational Budget

We conducted a total of 462 evaluations during development and experimentation. Final results are based on a full evaluation run that took approximately 1 day, 14 hours, and 26 minutes to complete, while most individual runs completed in under 10 minutes. Our experiments primarily used the GPT-4o model from OpenAI, which has approximately 1.8 trillion parameters. All model inference was performed via the OpenAI API using GPT-4o endpoints, with no local GPU training. Funding for API access was provided by Block Inc. through a donation to the Generative AI @ Berkeley club. The total estimated compute cost, based on API usage, falls between $1,000 and $10,000 USD. While we did not use dedicated hardware infrastructure, this estimate reflects the cost of hosted model inference at scale and provides a reasonable approximation for reproduction under similar conditions.

E1. AI Assistant Use

We made extensive use of AI assistants, specifically OpenAI's GPT-4 and GPT-4o models, throughout the development of this paper. The assistant was used to generate and debug code, write and revise paragraphs, synthesize related work, and identify relevant research papers. While all content was critically reviewed and edited by the authors, the assistant served as an iterative collaborator in refining both the implementation and the writing process. All final claims, analyses, and design decisions were made by the authors.

References

Lenore Blum and Manuel Blum. 2022. A theory of consciousness from a theoretical computer science perspective: Insights from the conscious turing machine. *Proceedings of the National Academy of Sciences (PNAS)*, 119(21):e2115934119.

Zannis Fountas, Albert Webson, Xuezhi Han, Luke Zettlemoyer, and Tatsunori B. Hashimoto. 2024. Episodic memory in large language models.

Shengran Hu, Cong Lu, and Jeff Clune. 2024. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*.

Reed Lindsey, Noemi Derzsi, Ethan Perez, and Jared Kaplan. 2025. On the biology of a large language model. *Anthropic Technical Report*. Https://www.anthropic.com/research/biology-of-a-large-language-model.

Noah Shinn, Andy Zou, and Piero Molino. 2023. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, and Dieter Fox. 2021. Alfworld: Aligning text and embodied environments for interactive learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Theodore R. Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L. Griffiths. 2024. Cognitive architectures for language agents. *Transactions on Machine Learning Research (TMLR)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS 2022)*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed H. Awadallah, Ryen W. White, Doug Burger, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversations. *arXiv preprint arXiv:2308.08155*.

Ziru Xiao, Luke Zettlemoyer, and Tatsunori B. Hashimoto. 2024. Infinite context windows for language models.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Siyu Zhou, Tianyi Zhou, Yijun Yang, Guodong Long, Deheng Ye, Jing Jiang, and Chengqi Zhang. 2024. Wall-e: World alignment by rule learning improves world model-based llm agents. *arXiv preprint arXiv:2410.07484*.

A Examples

A.1 Example of successful run

A.1.1 Task:

Task: put a knife in sidetable.

A.1.2 Agent Transitions taken:

16

A.1.3 Actions Taken:

Action: None
Observation: You are in the middle of a room. Looking quickly around you, you see a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 1, a drawer 8, a drawer 7, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a sidetable 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1.

Action: go to countertop 1
Observation: You arrive at loc 7. On the countertop 1, you see a apple 2, a bread 2, a bread 1, a dishsponge 2, a egg 1, a fork 3, a fork 2, a fork 1, a houseplant 1, a kettle 1, a knife 4, a knife 1, a lettuce 3, a lettuce 2, a pan 1, a papertowelroll 1, a peppershaker 3, a peppershaker 2, a potato 1, a saltshaker 1, a spatula 1, a spoon 2, and a spoon 1.

Action: take knife 1 from countertop 1
Observation: You pick up the knife 1 from the countertop 1.

Action: go to sidetable 1
Observation: You arrive at loc 22. On the sidetable 1, you see a mug 1, a peppershaker 1, and a tomato 1.

Action: put knife 1 in/on sidetable 1
Observation: You put the knife 1 in/on the sidetable 1.

A.1.4 Example Belief States:

First Belief Formed:

BELIEF STATE: [Timestep 0: I have just entered the environment and received the initial percept. The task is to put a knife in sidetable. Currently, I see numerous objects in the room including cabinets (1-4), drawers (1-8), countertop 1, and sidetable 1. The knife is not immediately visible, suggesting I need to locate it first. Based on Cluster 3 knowledge about items appearing in unexpected locations, I believe the knife could be in any of the visible containers or surfaces. My immediate belief is that I should systematically search for the knife, prioritizing likely locations first (countertop 1, drawers 1-8) before checking less probable ones (cabinets, appliances). The target is sidetable 1, but I must first acquire the knife.]

Second Belief Formed:

BELIEF STATE: [Timestep 1: I have successfully located two knives (knife 1 and knife 4) on countertop 1, confirming my initial belief that countertops are likely locations for knives. The task requires placing one knife in sidetable 1. Based on Cluster 2 knowledge about carrying one item at a time, I should take one knife (either knife 1 or knife 4) and proceed to sidetable 1 for placement. The newly admissible take actions confirm I can now interact with the knives.]

A.2 Example of a failed run

A.2.1 Task:

Task: cool some apple and put it in microwave.

A.2.2 Agent Transitions taken:

99

A.2.3 Actions Taken:

Action: None
Observation: You are in the middle of a room. Looking quickly around you, you see a cabinet 21, a cabinet 20, a cabinet 19, a cabinet 18, a cabinet 17, a cabinet 16, a cabinet 15, a cabinet 14, a cabinet 13, a cabinet 12, a cabinet 11, a cabinet 10, a cabinet 9, a cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 2, a countertop 1, a diningtable 1, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1.

Action: go to fridge 1
Observation: You arrive at loc 1. The fridge 1 is closed.

Action: open fridge 1
Observation: You open the fridge 1. The fridge 1 is open. In it, you see a bowl 1, and a mug 1.

Action: go to countertop 1
Observation: You arrive at loc 35. On the countertop 1, you see a bread 1, a butterknife 1, a lettuce 1, a pan 1, and a pot 1.

Action: go to countertop 2
Observation: You arrive at loc 29. On the countertop 2, you see a cup 3, a knife 3, a knife 2, a spatula 3, and a tomato 1.

Action: go to diningtable 1
Observation: You arrive at loc 16. On the diningtable 1, you see a apple 1, a cup 2, a cup 1, a egg 1, a mug 3, a potato 1, a soapbottle 1, a spatula 2, a spatula 1, a spoon 2, and a spoon 1.

Action: go to fridge 1
Observation: You arrive at loc 1. The fridge 1 is open. In it, you see a bowl 1, and a mug 1.

Action: cool apple 1 with fridge 1
Observation: The action `cool apple 1 with fridge 1` is not in the list of admissible actions for the current timestep.

Action: put apple 1 in/on fridge 1
Observation: The action `put apple 1 in/on fridge 1` is not in the list of admissible actions for the current timestep.

Action: examine fridge 1
Observation: The fridge 1 is open. In it, you see a bowl 1, and a mug 1.

Action: go to microwave 1
Observation: You arrive at loc 31. The microwave 1 is closed.

Action: open microwave 1
Observation: You open the microwave 1. The microwave 1 is open. In it, you see nothing.

Action: examine microwave 1
Observation: The microwave 1 is open. In it, you see nothing.

Action: put apple 1 in/on microwave 1
(Hallucinating agent outputs begin after this action)
Observation: The action `put apple 1 in/on microwave 1` is not in the list of admissible actions for the current timestep.

A.2.4 Example Belief States:

Belief 1:
BELIEF STATE: [Timestep 12: I have successfully opened microwave 1, which is empty. The newly admissible actions do not yet show the option to place the apple inside, suggesting I may need to explicitly examine the microwave or wait for the environment to update. Based on Episode 26's memory, I believe the 'put apple 1 in/on microwave 1' action should become available now that the microwave is open and I'm holding the apple.]
Belief 2:
BELIEF STATE: [Timestep 13: Examining the microwave again yielded no new information or actions. This suggests the environment may not recognize the apple as properly cooled for microwave placement. Based on Cluster 8 knowledge ("Environmental validation is prerequisite for efficient task execution"), I now believe the task may be unsolvable in its current state due to the missing cooling step. However, I will make one final attempt to place the uncooled apple in the microwave, as the environment may handle the cooling requirement implicitly.]
Belief 3: BELIEF STATE: [Timestep 14: The persistent inability to either cool the apple or place it in the microwave suggests the task may be unsolvable due to environmental constraints. Based on Cluster 8 knowledge ("absence of critical interaction protocols renders tasks fundamentally unsolvable"), I conclude the environment lacks the necessary cooling mechanism for apples despite having a fridge. With only 7 actions remaining, further attempts would be futile. The system will terminate the task as INCOMPLETE due to environmental limitations.]

A.3 Concepts Examples:

- You can carry **one item at a time**; if multiple items need to be moved, make multiple trips to complete the task.
- Items frequently needed for tasks may be located on **countertop surfaces** rather than conventional storage spaces like cabinets, highlighting the value of thoroughly checking visible areas when searching for necessary task items.
- **Identify the location** of necessary items before attempting to retrieve or move them, as it aids in efficient planning and task completion within action limits.

A.4 Additional Data

Effect of Episodic Memory on Efficiency

Effect of Planning Prompt on Performance