DAVIS: Planning Agent with Knowledge Graph-Powered Inner Monologue

Anonymous ACL submission

Abstract

Designing a generalist scientific agent capable of performing tasks in laboratory settings to assist researchers has become a key goal in recent Artificial Intelligence (AI) research. Unlike everyday tasks, scientific tasks are inherently more delicate and complex, requiring agents to possess a higher level of reasoning ability, structured and temporal understanding of their environment, and a strong emphasis on safety. Existing approaches often fail to address these multifaceted requirements. To tackle these challenges, we present DAVIS[1]. Unlike traditional retrieval-augmented generation (RAG) approaches, DAVIS incorporates structured and temporal memory, which enables model-based planning. Additionally, DAVIS implements an agentic, multi-turn retrieval system, similar to human's inner monologue, allowing for a greater degree of reasoning over past experiences. Through internal planning before each step, DAVIS significantly reduces the likelihood of taking unsafe actions compared to baseline models. DAVIS demonstrates significant performance on the ScienceWorld benchmark, outperform previous approaches on 8 out of 9 elementary science subjects. In addition, DAVIS's World Model demonstrates competitive performance on the famous HotpotQA dataset for multi-hop question answering. To the best of our knowledge, DAVIS is the first RAG agent to employ an interactive retrieval method in RAG pipeline.

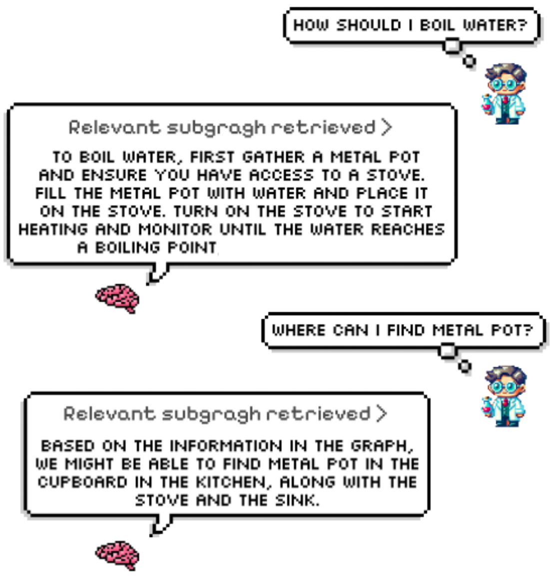

Figure 1: Visualization of DAVIS's inner monologue during decision-making. The agent uses its World Model to retrieve relevant subgraphs from a Temporal Knowledge Graph (TKG) for reasoning.

1 Introduction

A core focus of current Artificial Intelligence (AI) research is the development of artificial agents capable of autonomously performing human tasks with high decision-making autonomy (Ahn et al., 2022; Zhao et al., 2024; Wang et al., 2024; Putta et al., 2024). While Reinforcement Learning (RL) has traditionally been used to create goal-oriented agents in Markovian environments (Mnih et al., 2013; Schrittwieser et al., 2020; Hafner et al., 2020), it often suffers from sample inefficiency, limited generalizability, and poor interpretability, making real-world deployment challenging (Dulac-Arnold et al., 2019). Recently, large language models (LLMs) (Radford et al.; Touvron et al., 2023; DeepSeek-AI et al., 2025) have revolutionized the creation of autonomous agents by leveraging natural language understanding to enhance interpretability and generalization. These LLM-based agents have shown great promise in critical domains such as healthcare (Qiu et al., 2024) and scientific research (Schmidgall et al., 2025) by mimicking human decision-making processes and enabling more intuitive reasoning and actions.

Several approaches have enhanced agentic reasoning and decision-making. SwiftSage (Lin et al., 2023) emulates the fast and slow thinking of hu-

[1]All code and prompts are available at Anonymous Github

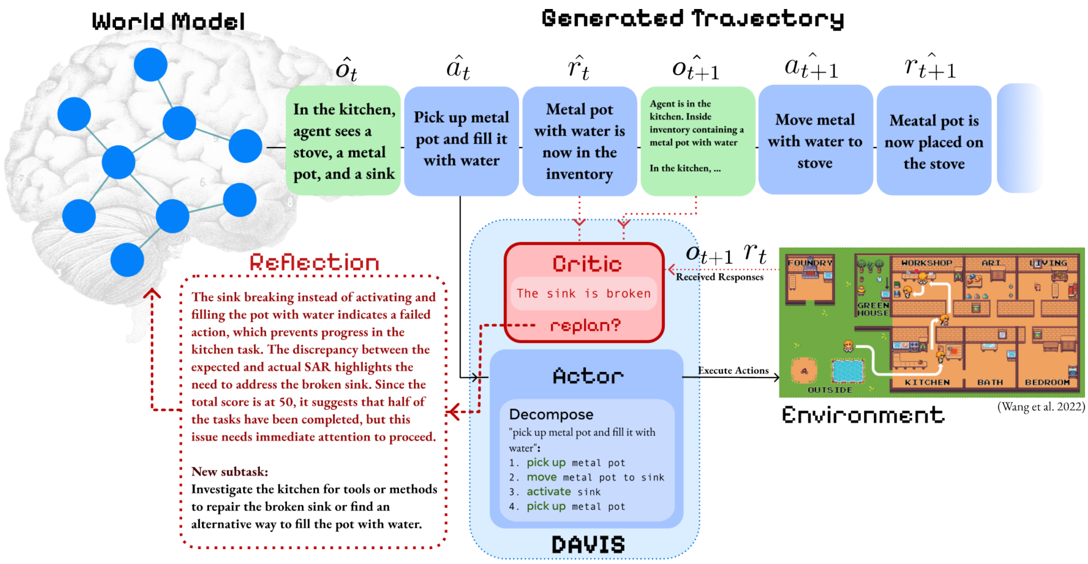

Figure 2: Overview of DAVIS's decision-making process. The World Model generates a feasible course of actions, which are translated by the actor and executed sequentially by the agent in the environment. A reflection mechanism detects discrepancies between expected and actual outcomes, prompting the Critic module to identify failures and suggest replanning.

mans with fine-tuned language models for planning. SayCan (Ahn et al., 2022) decomposes tasks into subgoals, while ReAct (Yao et al., 2023) [2] integrates reasoning into execution. RAG-based systems like Reflexion (Shinn et al., 2023) and RAP[3] (Kagaya et al., 2024) retrieve past experiences via semantic search, but their unstructured memory limits multi-hop reasoning and causal understanding. These systems retrieve static information rather than engaging in agentic, multi-turn retrieval, preventing dynamic adaptation.

Humans do not retrieve past knowledge statically; instead, we actively reflect, question our understanding, and refine our knowledge through internal dialogues. Inspired by this, we introduce DAVIS, an agentic multi-turn retrieval system that mirrors human cognition by enabling iterative interactions between the agent and its memory during the planning stage. DAVIS actively engages with its World Model (WM), a temporal knowledge graph-based QA system, to refine its understanding before execution. Like human brainstorming, DAVIS engages in conversation with its WM to retrieve past experiences, evaluate actions, identify gaps, and optimize strategies.

Consequently, DAVIS proves to be effective for iterative reasoning within scientific domains. Specifically, DAVIS outperforms 4 other baselines (Ahn et al., 2022; Kagaya et al., 2024; Yao et al., 2023; Shinn et al., 2023) on 8 out of 9 science subjects in the ScienceWorld (Wang et al., 2022) environment.[4] DAVIS's WM achieves competitive performance on the HotpotQA (Yang et al., 2018) dataset. Our contributions can be summarized as follows:

- We introduce DAVIS, a novel agentic reasoning framework that engages in multi-turn retrieval and self-reflection to refine decision-making.
- Unlike static retrieval methods which use unstructured knowledge, DAVIS leverages a structured temporal knowledge graph memory system to enable multi-hop reasoning and causal understanding.
- DAVIS continuously interacts with its World Model, mimicking human-like inner monologue to improve adaptability and safety.
- Empirical evaluations show that DAVIS outperforms prior agentic reasoning models across scientific benchmarks, demonstrating superior planning and execution.

[2] SwiftSage, Reflexion, SayCan, and ReAct are used under MIT license

[3] RAP is used under MIT license

[4] ScienceWorld is used under Apache 2.0 license

2 Background & Related Work

2.1 TextWorld Environments

TextWorld (Côté et al., 2019) is a class of sandbox environments for training textual agents through interactive text-based games. Similar to Zork, it acts as a game master, providing textual feedback on player actions, managing inventory, and tracking task progress. The absence of visual components makes it computationally efficient, with TextWorld-Express achieving up to 4 million simulation steps per second, enabling cost-effective large-scale training (Jansen and Côté, 2023).

Historically, text-based games presented significant challenges to learning agents. These games are partially observable, as descriptive text often omits complete environmental details. Additionally, the combinatorial and compositional nature of both the observation and action spaces posed substantial difficulties for most reinforcement learning algorithms (Jansen and Côté, 2023). However, with the advent of large language models (LLMs), these challenges have become surmountable. LLMs' advanced language understanding and reasoning capabilities make them well-suited for navigating and learning from text-based environments.

In this work, we deploy DAVIS to the TextWorld environment for evaluation, leveraging the linguistic capabilities of LLMs to address the complexities of text-based game simulations effectively.

2.2 LLM-Based Agentic Systems

The development of LLM-based agentic systems in complex environments has seen significant advancements, drawing heavily from human decision-making processes. Broadly, these systems fall into two main paradigms: direct online interaction with chain-of-thought (CoT) reasoning and Retrieval-Augmented Generation (RAG).

The first paradigm involves agents interacting directly with their environment using CoT reasoning (Yao et al., 2023; Ahn et al., 2022; Lin et al., 2023). Chain-of-Thought prompting (Wei et al., 2023) enables large language models to decompose complex tasks into smaller, interpretable reasoning steps, which is more consistent with how human approach decision making. However, CoT-based systems lack robust memory components for long-term learning and adaptability across multiple tasks. The absence of memory has been linked to increased hallucination and stochasticity in task planning (Guerreiro et al., 2023), posing significant risks in critical domains such as scientific research.

The second paradigm, RAG-based systems, integrates retrieval mechanisms with generative capabilities, enabling agents to access relevant external knowledge during task execution. In the Minecraft domain, extensive work has been done on RAG-based agents, with JARVIS-1 (Wang et al., 2023b) and Voyager (Wang et al., 2023a) representing the state-of-the-art. Since Minecraft is one of the most popular video games in the world, these agents leverage the extensive in-domain knowledge of LLMs but face significant limitations in scientific environments, where tasks often involve unknown skills and cannot rely on pre-existing knowledge. A more general and iterative approach involving multiple trials is necessary in such cases.

For instance, Reflexion (Shinn et al., 2023) maintains logs of past trials to reflect on successes and failures, while RAP (Kagaya et al., 2024) retrieves semantically similar examples from past experiences to guide decision-making. Although these systems address some of the shortcomings of CoT-based methods, they often depend on unstructured vector databases for memory, which scatter information and limit multi-hop reasoning. Moreover, such systems lack mechanisms to handle temporal reasoning and iterative refinement, making them less suited for domains requiring adaptive and contextual understanding, such as scientific experimentation.

In addition, both paradigms used by these agents lack internal validation and planning capabilities, making them less effective in scientific domains where deliberate and accurate decision-making is crucial. These limitations underline the need for hybrid systems that integrate iterative reasoning, structured memory, and robust internal planning to enable agents to perform effectively in complex environments such as scientific research laboratories. DAVIS is designed with model-based planning in mind.

2.3 Graph Question Answering (Graph QA)

Graph Question Answering (Graph QA) systems have become effective tools for structured reasoning and information retrieval. GraphReader (Li et al., 2024) constructs a graph from document chunks and deploys an agent for exploration. HOLMES (Panda et al., 2024) extracts relevant documents, builds an entity-document graph, prunes it, and uses cosine similarity for answers. GraphRAG (Edge et al., 2024) generates an entity knowledge graph, pregenerates community summaries, and synthesizes responses. By encoding knowledge in a graph format, these systems excel at multi-hop reasoning over interconnected concepts, making them particularly valuable for domains that require relational understanding, such as scientific research. Unlike unstructured vector-based retrieval systems, Graph QA systems enable iterative retrieval, allowing agents to retrieve information, reason over it, and perform subsequent queries based on the refined context.

3 DAVIS

DAVIS adopts a model-based planning approach (Sutton and Barto, 1998), where the agent utilizes a WM as an internal representation of its surrounding environment.

3.1 Problem Formulation

We define the planning problem for DAVIS in a textual environment as a Partially Observable Markov Decision Process (POMDP), represented by:

$$\mathcal{P} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$$

In this formulation, $\mathcal{S}$ denotes the set of true environment states, which are not directly observable. $\mathcal{A}$ represents the set of available actions. $\mathcal{T}(s_{t+1} \mid s_t, a_t)$ is the state transition probability function, modeling the dynamics of the environment. $\mathcal{R}(s_t, a_t)$ is the reward function, specifying the immediate reward received after taking action a_t in state s_t. Ω is the set of possible observations. $\mathcal{O}(o_{t+1} \mid s_{t+1}, a_t)$ is the observation probability function, defining the likelihood of observing o_{t+1} given the new state s_{t+1} and action a_t. $\gamma \in [0, 1)$ is the discount factor, determining the present value of future rewards.

Since the true state s_t is not directly observable, the agent maintains a belief state b_t, which is a probability distribution over all possible states, representing the agent's estimate of the environment's state at time t. The belief state is updated based on the agent's actions and received observations. The agent selects an action $a_t \in \mathcal{A}$ based on its current belief state, following a policy π:

$$a_t = \pi(b_t)$$

After executing the action a_t, the agent receives a reward $r_t = \mathcal{R}(s_t, a_t)$ and transitions to a new state s_{t+1} according to the transition function $\mathcal{T}$. The objective of the agent is to find an optimal policy π^* that maximizes the expected cumulative discounted reward over time:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid \pi\right]$$

3.2 World Model (WM)

The World Model (WM) of DAVIS is represented as a Temporal Knowledge Graph (TKG), constructed through a combination of Stanford CoreNLP (Manning et al., 2014) for coreference resolution and LLM prompting for knowledge extraction. In textual environments, where state representations are conveyed in natural language, constructing an effective WM requires methods that can process and represent textual information efficiently and accurately.

State representation methods in text-based environments include text encoding techniques using recurrent neural networks (Narasimhan et al., 2015, He et al., 2016, Hausknecht et al., 2020), transformers (Kim et al., 2022), and knowledge graph (KG) representations (Ammanabrolu and Hausknecht, 2020). KGs offer structured and interpretable representations without requiring extensive training. Ammanabrolu and Riedl's (2021) framed KG construction in text-based games as a question-answering problem, where agents identified objects and their attributes. This approach demonstrated that higher-quality KGs led to improved control policies. DAVIS extends this concept to Temporal Knowledge Graphs, incorporating time-sensitive information to model dynamic environment changes. Temporal reasoning is critical in such settings, and as noted in (Lee et al., 2023), LLMs are highly effective in extrapolating TKGs using in-context learning.

Let $G_t = (\mathcal{E}, \mathcal{R}, \mathrm{T})$ denote the Temporal Knowledge Graph (TKG) at time t, where $\mathcal{E}$ is the set of entities at t, $\mathcal{R}$ is the set of relations representing relationships between entities at, and T is the set of timestamps associated with each relation e_i.

During training, when DAVIS executes an action a_t and receives the subsequent observation o_{t+1}, the transition is stored as:

$$(o_t \parallel a_t \parallel o_{t+1})$$

We prompted an LLM to summarize the concatenated transition and applied Stanford CoreNLP for coreference resolution. The resolved text is then analyzed to extract entities V_t and relations relation tuples using LLM-based parsing.

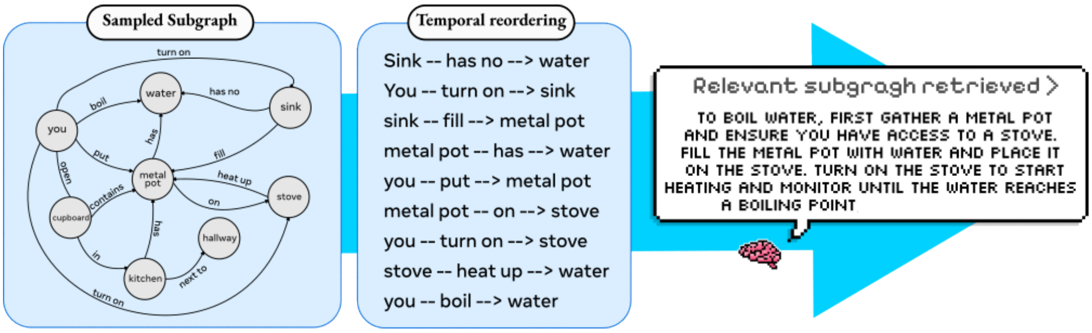

Figure 3: Illustration of DAVIS's retrieval and reasoning process. The left panel shows the sampled subgraph representing relevant entities and their relationships. The middle panel depicts the temporal reordering of the retrieved information to establish a coherent sequence of actions. The right panel demonstrates how DAVIS generates a structured and interpretable response by summarizing the retrieved knowledge.

Algorithm 1 Planning with Retrieval-Augmented World Model

Input: $\tau, \mathcal{R}$ **Parameters**: L, k **Output**: τ

for $t = 1$ to L **do**
  $\hat{s}_t \leftarrow f(\tau)$ ▷ State estimation
  $\hat{a}_t \leftarrow \pi(\hat{s}_t, \mathcal{R}, k)$
  $\tau \leftarrow \tau \cup \hat{a}_t$
  $\hat{o}_{t+1}, \hat{r}_{t+1} \leftarrow \textsc{Transition}(\hat{s}_t, \hat{a}_t)$ ▷ Algorithm 2
  $\tau \leftarrow \tau \cup \{\hat{o}_{t+1}, \hat{r}_{t+1}\}$
  if (τ) violates safety constraints (optional) **then**
    Alert supervisor
  end if
end for
return τ

Each extracted tuple (v_i, e_j, v_k, τ) is added to the TKG, where the timestamp τ records the time at which the fact was introduced:

$$G_{t+1} = G_t \cup \{(v_i, e_j, v_k, \tau)\}$$

3.3 Retrieval-Augmented Model Approximation

As demonstrated in Lee et al.'s (2023), LLMs excel at recognizing temporal patterns and extrapolating future events based on past data. DAVIS leverages this capability to approximate future states and rewards. For example, if sufficient past data indicates that opening a cupboard often reveals a kettle, the LLM can infer such transitions purely from learned patterns without requiring explicit pre-programmed rules. Unlike prior works (Kagaya et al., 2024; Shinn et al., 2023) that rely on vector-based retrieval of experiences, DAVIS employs a more agentic approach. Instead of passively retrieving information, DAVIS engages in a conversational process with its WM, iteratively querying to fill knowledge gaps while retrieving relevant subgraphs to generate informed responses. The retrieval system is described in 3.4.

Although true state s_t is not directly observable as mentioned in subsection 3.1, it is theoretically possible to maintain a statistic $f(\tau)$ that approximates the belief state from the trajectory history. The statistic is updated recurrently, and captures all relevant information necessary for optimal-decision-making (Nguyen et al., 2021; Åström, 1965). Applying this to DAVIS, we approximate the belief state $\hat{b}_t$ with equation:

$$\hat{b}_t = f(\tau_{t':t}),$$

where $f(\cdot)$ is a prompted LLM that extracts relevant information from the trajectory history, and is updated recurrently with new observations and actions. To further refine decision-making, DAVIS maintains an inner monologue $\mathcal{M}_t$, a running list of iterative queries and answers exchanged between DAVIS and its WM, as illustrated in Figure 1. This monologue allows the system to dynamically update its WM based on retrieved insights.

DAVIS optimizes its policy while simultaneously learning approximations of the transition and reward models using its WM. The learned functions incorporating the inner monologue are:

$$\text{Policy:} \quad \pi(a_t \mid \hat{b}_t, \mathcal{M}_t) \tag{1}$$
$$\text{Transition Model:} \quad \hat{\mathcal{T}}(o_{t+1} \mid \hat{b}_t, a_t, \mathcal{M}_t) \tag{2}$$
$$\text{Reward Model:} \quad \hat{\mathcal{R}}(r_t \mid \hat{b}_t, a_t, \mathcal{M}_t) \tag{3}$$

With the approximated belief $\hat{b}_t$, DAVIS's WM estimates the transition and reward models using prior experiences retrieved from a Temporal Knowledge Graph (TKG). Since DAVIS is designed as an imitation agent, it also leverages prior experiences to directly inform its policy as defined in (1). This retrieval-driven approximation enables DAVIS to

Algorithm 2 Transition Prediction

Input: $\hat{b}_t, \hat{a}_t, k$ **Output**: $\hat{o}_{t+1}, \hat{r}_{t+1}$

$\mathcal{M} \leftarrow \emptyset$ ▷ Initialize inner monologue set
$i \leftarrow 0$
while $i < k$ or not `predicted` **do**
  $\hat{o}_{t+1}, q \leftarrow \hat{T}(\hat{b}_t, \hat{a}_t, \mathcal{M})$
  $\hat{r}_{t+1}, q \leftarrow \hat{R}(\hat{b}_t, \hat{a}_t, \mathcal{M})$
  if $q \neq \emptyset$ **then**
    $\mathcal{M} \leftarrow \mathcal{M} \cup \{(q, \texttt{graphQA}(q))\}$
  end if
  $i \leftarrow i + 1$
end while
return $\hat{o}_{t+1}, \hat{r}_{t+1}$

construct an adaptive and context-aware model of the world, allowing for informed decision-making in complex, temporally dependent environments.

3.4 Retrieval System

Given a query q, such as *"Where can I find water?"*, the WM first narrows its search to relevant entity types such as `Person (PER)` and `Location (LOC)`. It then selects the two most relevant entities from the available options. Limiting the scope to two entities is computationally efficient and ensures a manageable search space without sacrificing relevant context. The query is then expanded and processed as follows and illustrated in Figure 3:

1. **We iteratively expand** the current list of selected entities by adding their neighbors, forming a maximal subgraph as ignoring temporal information might result in an infeasible path.
2. **We reorder the edges** in the maximal subgraph based on timestamps. This reordering shows the proper sequence of events.
3. **The temporal sequence is then passed to an LLM** as in-context examples for extrapolation and summarization, enabling the LLM to generate a coherent response.

3.5 Planning with a WM

With the reward model and transition model approximated, we can now plan action trajectories within the WM. Algorithm 1 describes the WM-incorporated planning process of DAVIS.

3.6 Executing Plan with Actor-Critic

For plan execution, we employ an actor-critic structure, consisting of two distinct models: the actor R_a and the critic R_c, integrated with the WM architecture. The process is illustrated in Figure 2. Below, we provide a formalized description of each model and its role within DAVIS.

World Model (WM)

The primary objective of the WM is to generate a comprehensive plan or trajectory for achieving a specific task within the environment. Analogous to human decision-making, the WM enables DAVIS to anticipate environmental changes, minimize risky actions, and improve sample efficiency. Given an initial observation estimate $\hat{o}_t$, the WM generates a predicted trajectory

$$\tau_{t:t+L} = \Big\{(\hat{o}_i, \hat{a}_i, \hat{o}_{i+1}, \hat{r}_{i+1})\Big\}_{i=t}^{t+L-1}$$

of length L. This trajectory $\tau_{t:t+L}$ is passed to the actor-critic model for execution in the environment.

Actor

The actor R_a is responsible for decomposing each high-level action $\hat{a}_t \in \tau$ into executable commands within the given environment domain. Additionally, it predicts intermediate state transitions between actions:

$$\hat{\tau}_{t:t+L'} = R_a(\tau_{t:t+L})$$

where $L' \geq L$ accounts for the expanded trajectory with low-level, executable actions. The actor model is prompted with permissible commands in the current environment. After decomposition, the expanded trajectory $\hat{\tau}_{t:t+L'}$ is executed step-by-step in the environment, producing actual environment responses:

$$(o_t, r_t, o_{t+1}) = \mathcal{E}(\hat{a}_t)$$

where $\mathcal{E}$ is the environment transition function that maps the executed action $\hat{a}_t$ to the resulting observation o_{t+1} and reward r_t. These results are passed to the critic model.

Critic

The critic R_c evaluates the actual execution results against the predicted trajectory τ. The comparison is performed through an LLM-based evaluation function, which assesses the semantic consistency between the expected and actual observations. At each timestep t, the critic receives the predicted state transition $(\hat{o}_t, \hat{r}_t, \hat{o}_{t+1})$ and the actual environment response (o_t, r_t, o_{t+1}) obtained from executing $\hat{a}_t$ in the environment.

The LLM-based critic compares these components via a prompted evaluation function R_c:

$$\Delta_t = R_c\Big((\hat{o}_t, \hat{r}_t, \hat{o}_{t+1}), (o_t, r_t, o_{t+1})\Big)$$

where Δ_t is a qualitative feedback score representing the level of agreement between the predicted and actual transitions. Based on the LLM's response, the critic determines whether replanning is necessary. If the predicted and actual observations deviate significantly, the critic updates the reflection memory $\mathfrak{R}_t$ and triggers replanning:

$$\mathfrak{R}_{t+1} = \mathfrak{R}_t \cup \{(o_t, \hat{s}_t, \Delta_t)\}$$

Algorithm 1 is then called to replan the new subtask. For instance, if the task is "using the stove to heat water" and the agent encounters an exception (e.g., the stove is broken), the LLM evaluates the exception, updates $\mathcal{M}_t$, and suggests a revised subtask such as "find an alternative heating method."

4 Experiment

4.1 ScienceWorld environment

We selected ScienceWorld (Wang et al., 2022) for evaluation. It includes 30 tasks spanning 9 different subjects derived from the grade school science curriculum, which provide a structured framework for assessing the performance of AI agents, including predefined evaluation metrics that are key to establishing a fair comparison. The agent is deployed in a simulated laboratory setting and must navigate through eight distinct functional rooms, using various tools and equipment to complete tasks such as measuring the boiling temperature of an unknown substance. Each task features over 100 possible variations to prevent overfitting. The environment demands extensive world knowledge, commonsense reasoning, strong deduction, and problem-solving skills. A higher score indicates more progression toward task completion, representing the agent's ability to finish the task. For example, a score of 75 indicates that the agent completed 75% of the task before picking the wrong action that led to task termination. Appendix section A details the ScienceWorld environment.

4.2 Performance

We evaluated DAVIS on 30 tasks from the ScienceWorld benchmark, comparing its performance against state-of-the-art baseline agents: SayCan, ReAct, Reflexion, and RAP. Baselines were selected based on their competitive performance, available implementations, and relevance to ScienceWorld. We acknowledge the current state-of-the-art method on ScienceWorld, SwiftSage (Lin et al., 2023). However, it was excluded from our replication baselines because discrepancies between the available code and the documented evaluation methods made direct replication infeasible. For consistency, all baselines were reimplemented to align with the latest ScienceWorld version. To ensure fairness, both RAP and DAVIS utilized memory constructed from five episodes of golden trajectories rather than the ReAct-based approach proposed in Kagaya et al.'s (2024). Performance was averaged across subjects for comparison, with details on tasks and subjects provided in Table 5 and Table 3 in the appendix. As shown in Figure 4, DAVIS outperformed all baselines in 8 out of 9 subjects, achieving an overall average score of 65.06—approximately 1.8 times higher than competing methods. Full results for each task, including standard deviations, are available in the appendix Table 6.

4.3 Ablational study

We compared two versions of DAVIS: one that utilizes its constructed WM during planning and another that relies solely on internal knowledge from LLMs for planning. We selected two long, two medium, and two short tasks, averaging the results over three variations, to underscore the importance of knowledge grounding in state management. Table 1 shows that adding the WM enhances DAVIS's performance across various tasks.

Task	DAVIS	DAVIS_{WM}
Long Tasks		
Melt (1-2)	3.00	70.00
Determine Melting Point Unk. (2-3)	5.00	92.33
Medium Tasks		
Mix Paint Secondary (6-1)	40.00	36.37
Test Conductivity (3-3)	55.00	58.33
Short Tasks		
Lifespan Longest-Lived (7-1)	66.67	100.00
Find Living Thing (4-1)	25.00	100.00

Table 1: DAVIS performance with and without WM

4.4 Multi-hop Q&A

We evaluated the performance of DAVIS's WM on the HotpotQA multi-hop QA benchmark using 200 randomly sampled instances, following the methodology of Li et al.'s (2024). DAVIS (GPT-4o) achieved an F1 score of 75.0, exceeding the performance of GraphReader and GraphRAG, while performing on par with HOLMES. In exact match (EM) accuracy, DAVIS outperformed GraphReader and GraphRAG, falling slightly behind HOLMES. Compared to the standard retrieval-

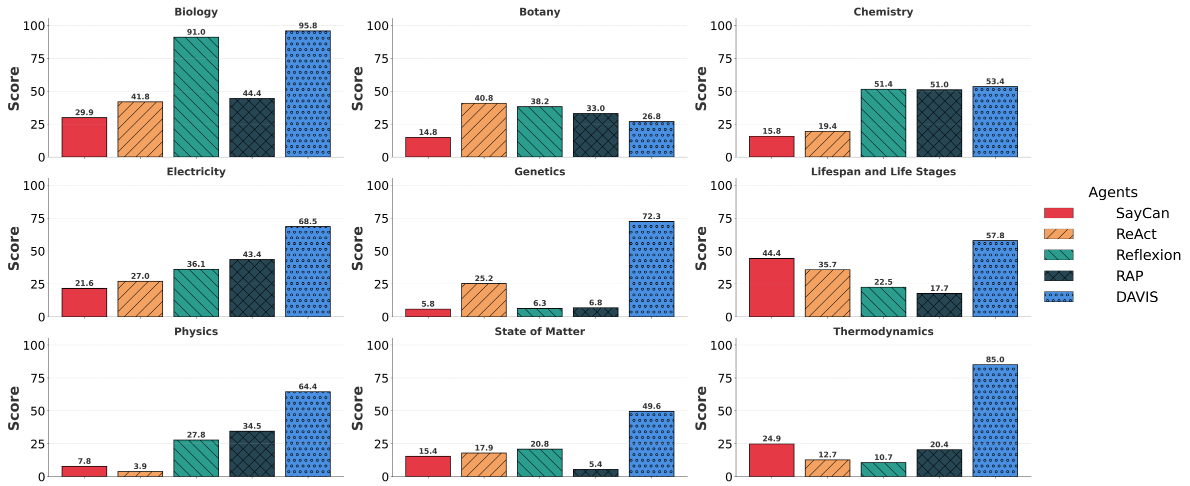

Figure 4: Performance comparison of different agents (SayCan, ReAct, Reflexion, RAP, and DAVIS) across multiple scientific domains. Each subplot represents the average score achieved in a specific category, demonstrating DAVIS's superior performance in most tasks. For full results, view table 6

based baselines BM25 (Robertson and Zaragoza, 2009) and OpenAI's Ada-002, DAVIS demonstrated substantial improvements, highlighting the effectiveness of structured memory retrieval and graph-based reasoning. While HOLMES achieved the highest scores, both HOLMES and GraphRAG rely on hyper-relational knowledge graphs to connect source data with extracted entities (Anokhin et al., 2024), similar to DAVIS. However, unlike DAVIS, these models lack mechanisms for dynamic updates, limiting their adaptability in evolving environments. These results suggest that hyper-relational graphs provide an effective framework for memory organization in LLM agents, with broad potential applications.

Table 2: DAVIS's World Model demonstrates competitive performance to strong QA baseline graph agents

Method	HotpotQA	
	EM	F1
BM25 (top-3)	45.7	58.5
Ada-002 (top-3)	45.0	58.1
GPT 4o	53.0	68.4
GPT-4-turbo	46.0	63.5
GraphReader (GPT-4)	55.0	70.0
HOLMES (GPT-4)	**66.0**	**78.0**
GraphRAG (GPT-4o-mini)	58.7	63.3
DAVIS (GPT-4o)	60.0	75.0
DAVIS (GPT-4-turbo)	58.0	74.0

5 Conclusion

We have introduced DAVIS, an agent designed for scientific interactive reasoning tasks in complex environments. DAVIS represents a novel approach that leverages a structured World Model (WM) in the form of a temporal knowledge graph, enabling iterative retrieval and reasoning over past experiences. This structured representation allows DAVIS to approximate both the transition dynamics and reward models of its environment, facilitating more informed decision-making. DAVIS also uniquely uses an interactive retrieval process, which combines iterative querying with contextual reasoning to fill knowledge gaps and refine understanding. This process is further augmented by DAVIS's ability to perform internal planning and validation before interacting with the environment. By engaging in this pre-execution deliberation, DAVIS can detect potential unsafe behaviors early, evaluate the long-term consequences of its actions, and ensure alignment with scientific protocols. Such capabilities make DAVIS particularly suited for hands-on scientific tasks that require precision, adaptability, and adherence to rigorous experimental procedures.

Evaluations across several scientific domains, including thermodynamics, biology, and physics, demonstrate the efficacy of DAVIS's structured knowledge representation and retrieval methods. DAVIS significantly outperforms baseline agents by combining robust planning with the capacity for iterative reasoning, enabling it to generalize effectively from demonstrations to new tasks.

6 Limitations

While DAVIS demonstrates strong reasoning capabilities and improved performance over previous agentic approaches, it has several limitations that we will address in future research.

6.1 High operational cost

DAVIS heavily relies on Large Language Models (LLMs), making it computationally expensive. Due to its careful planning and reasoning process, it sends and receives an average of 43,000 tokens per action, resulting in an estimated cost of $0.43 per action. For tasks requiring 100 actions, this cost can escalate to $43 per episode, leading to a total experimental cost of approximately $3,000 for 90 variations.

6.2 Sensitive to LLM performance

DAVIS's reasoning and decision-making abilities fluctuate based on the underlying LLM's performance. Factors such as model version updates, prompt engineering quality, and external API changes can lead to accuracy, consistency, and response time variability. This dependence on LLM stability makes DAVIS susceptible to unexpected performance shifts, which may impact reliability in dynamic or evolving environments.

6.3 Biased Planning & Knowledge Dependence

DAVIS's decision-making process is heavily influenced by the Temporal Knowledge Graph (TKG), which serves as its structured memory. However, this dependence can lead to biased planning, as DAVIS prioritizes information within the graph. Although efforts were made to increase data diversity by populating the knowledge graph with 150 different ScienceWorld task variations, the model still struggles when encountering novel scenarios or incomplete knowledge. Future work should explore adaptive knowledge integration to mitigate bias.

6.4 Lack of Multimodal Capabilities

DAVIS operates exclusively in textual environments, limiting its applicability as an embodied agent. The absence of visual, auditory, or sensory perception restricts its ability to interact with real-world multimodal tasks. Future research should focus on integrating visual and sensor-based input processing to enhance generalization and deployment in robotic or multimodal AI systems.

References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. *arXiv preprint*. ArXiv:2204.01691 [cs].

Prithviraj Ammanabrolu and Matthew Hausknecht. 2020. Graph Constrained Reinforcement Learning for Natural Language Action Spaces. *arXiv preprint*. ArXiv:2001.08837 [cs, stat].

Prithviraj Ammanabrolu and Mark O. Riedl. 2021. Learning Knowledge Graph-based World Models of Textual Environments. *arXiv preprint*. ArXiv:2106.09608 [cs].

Petr Anokhin, Nikita Semenov, Artyom Sorokin, Dmitry Evseev, Mikhail Burtsev, and Evgeny Burnaev. 2024. AriGraph: Learning Knowledge Graph World Models with Episodic Memory for LLM Agents. *arXiv preprint*. ArXiv:2407.04363 [cs].

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. 2019. TextWorld: A Learning Environment for Text-Based Games. In *Computer Games*, pages 41–75, Cham. Springer International Publishing.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint*. ArXiv:2501.12948 [cs].

Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. 2019. Challenges of Real-World Reinforcement Learning. *arXiv preprint*. ArXiv:1904.12901 [cs].

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. *arXiv preprint*. ArXiv:2404.16130 [cs].

Nuno M. Guerreiro, Duarte Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and André F. T. Martins. 2023. Hallucinations in Large Multilingual Translation Models. *arXiv preprint*. ArXiv:2303.16104 [cs].

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2020. Dream to Control: Learning Behaviors by Latent Imagination. *arXiv preprint*. ArXiv:1912.01603 [cs].

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. 2020. Interactive Fiction Games: A Colossal Adventure. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7903–7910. Number: 05.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016. Deep Reinforcement Learning with a Natural Language Action Space. *arXiv preprint*. ArXiv:1511.04636 [cs].

Peter A. Jansen and Marc-Alexandre Côté. 2023. TextWorldExpress: Simulating Text Games at One Million Steps Per Second. *arXiv preprint*. ArXiv:2208.01174 [cs].

Tomoyuki Kagaya, Thong Jing Yuan, Yuxuan Lou, Jayashree Karlekar, Sugiri Pranata, Akira Kinose, Koki Oguri, Felix Wick, and Yang You. 2024. RAP: Retrieval-Augmented Planning with Contextual Memory for Multimodal LLM Agents. *arXiv preprint*. ArXiv:2402.03610 [cs].

Minsoo Kim, Yeonjoon Jung, Dohyeon Lee, and Seungwon Hwang. 2022. PLM-based World Models for Text-based Games. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1324–1341, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dong-Ho Lee, Kian Ahrabian, Woojeong Jin, Fred Morstatter, and Jay Pujara. 2023. Temporal Knowledge Graph Forecasting Without Knowledge Using In-Context Learning. *arXiv preprint*. ArXiv:2305.10613 [cs].

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, Wenbo Su, and Bo Zheng. 2024. GraphReader: Building Graph-based Agent to Enhance Long-Context Abilities of Large Language Models. *arXiv preprint*. ArXiv:2406.14550 [cs].

Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. 2023. SwiftSage: A Generative Agent with Fast and Slow Thinking for Complex Interactive Tasks. *arXiv preprint*. ArXiv:2305.17390 [cs].

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *arXiv preprint*. ArXiv:1312.5602 [cs] version: 1.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. 2015. Language Understanding for Text-based Games using Deep Reinforcement Learning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1–11, Lisbon, Portugal. Association for Computational Linguistics.

Hai Nguyen, Brett Daley, Xinchao Song, Christopher Amato, and Robert Platt. 2021. Belief-Grounded Networks for Accelerated Robot Learning under Partial

Observability. *arXiv preprint*. ArXiv:2010.09170 [cs].

Pranoy Panda, Ankush Agarwal, Chaitanya Devaguptapu, Manohar Kaul, and Prathosh Ap. 2024. HOLMES: Hyper-Relational Knowledge Graphs for Multi-hop Question Answering using LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13263–13282, Bangkok, Thailand. Association for Computational Linguistics.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. 2024. Agent Q: Advanced Reasoning and Learning for Autonomous AI Agents. *arXiv preprint*. ArXiv:2408.07199 [cs].

Jianing Qiu, Kyle Lam, Guohao Li, Amish Acharya, Tien Yin Wong, Ara Darzi, Wu Yuan, and Eric J. Topol. 2024. LLM-based agentic systems in medicine and healthcare. *Nature Machine Intelligence*, 6(12):1418–1420. Publisher: Nature Publishing Group.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. 2025. Agent Laboratory: Using LLM Agents as Research Assistants. *arXiv preprint*. ArXiv:2501.04227 [cs].

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. 2020. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609. Publisher: Nature Publishing Group.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. *arXiv preprint*. ArXiv:2303.11366 [cs].

Richard S Sutton and Andrew G Barto. 1998. Reinforcement Learning: An Introduction.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint*. ArXiv:2302.13971 [cs].

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint*. ArXiv:2305.16291 [cs].

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. ScienceWorld: Is your Agent Smarter than a 5th Grader? *arXiv preprint*. ArXiv:2203.07540 [cs].

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. 2024. Describe, Explain, Plan and Select: Interactive Planning with Large Language Models Enables Open-World Multi-Task Agents. *arXiv preprint*. ArXiv:2302.01560 [cs].

Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, Xiaojian Ma, and Yitao Liang. 2023b. JARVIS-1: Open-World Multi-task Agents with Memory-Augmented Multimodal Language Models. *arXiv preprint*. ArXiv:2311.05997 [cs].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv preprint*. ArXiv:2201.11903 [cs].

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. *arXiv preprint*. ArXiv:1809.09600 [cs].

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint*. ArXiv:2210.03629 [cs].

Haiteng Zhao, Chang Ma, Guoyin Wang, Jing Su, Lingpeng Kong, Jingjing Xu, Zhi-Hong Deng, and Hongxia Yang. 2024. Empowering Large Language Model Agents through Action Learning. *arXiv preprint*. ArXiv:2402.15809 [cs].

K. J Åström. 1965. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205.

A ScienceWorld

ScienceWorld (Wang et al., 2022) is a benchmark designed to evaluate interactive reasoning in digital agents through a realistic laboratory simulation. Developed by the Allen Institute for AI, it provides a text-based environment that emulates scientific experiments, requiring agents to interact with objects, collect observations, and apply reasoning skills to solve tasks. The framework consists of approximately 40,000 lines of SCALA code with a PYTHON interface, following standard RL benchmarking practices.

The ScienceWorld environment consists of 10 interconnected locations (Fig. 5), each populated with up to 200 distinct object types, including scientific instruments, electrical components, biological specimens, substances, and common environmental elements like furniture and books. Agents can interact with objects through a predefined action space of 25 high-level actions, categorized into domain-specific operations (e.g., using a thermometer, measuring conductivity) and general interactions (e.g., moving, opening containers, picking up items). At each step, approximately 200,000 possible action-object combinations exist, though only a subset is relevant based on the context.

ScienceWorld tasks are designed to assess scientific reasoning across multiple disciplines. The dataset includes 30 distinct tasks (Table 3), covering a range of experimental procedures and problem-solving scenarios. These tasks are further grouped into 9 science domains (Table 5), including physics, chemistry, biology, and environmental science, allowing for targeted evaluation of an agent's ability to reason through various scientific concepts, making ScienceWorld a robust benchmark for testing multi-step reasoning in dynamic, interactive environments.

#	Task
1-1	Changes of State (Boiling)
1-2	Changes of State (Melting)
1-3	Changes of State (Freezing)
1-4	Changes of State (Any)
2-1	Use Thermometer
2-2	Measuring Boiling Point (Known)
2-3	Measuring Boiling Point (Unknown)
3-1	Create a Circuit
3-2	Renewable vs Non-Renewable Energy
3-3	Test Conductivity (Known)
3-4	Test Conductivity (Unknown)
4-1	Find a Living Thing
4-2	Find a Non-Living Thing
4-3	Find a Plant
4-4	Find an Animal
5-1	Grow a Plant
5-2	Grow a Fruit
6-1	Mixing (Generic)
6-2	Mixing Paints (Secondary Colours)
6-3	Mixing Paints (Tertiary Colours)
7-1	Identify Longest-Lived Animal
7-2	Identify Shortest-Lived Animal
7-3	Identify Longest-Then-Shortest-Lived Animal
8-1	Identify Life Stages (Plant)
8-2	Identify Life Stages (Animal)
9-1	Inclined Planes (Determine Angle)
9-2	Friction (Known Surfaces)
9-3	Friction (Unknown Surfaces)
10-1	Mendelian Genetics (Known Plants)
10-2	Mendelian Genetics (Unknown Plants)

Table 3: Tasks in ScienceWorld.

B DAVIS Implementation Details

We utilized GPT-4-turbo for reasoning, GPT-4o for question answering, and LLaMA3-70B for the Knowledge Graph construction pipeline. Agents were run for a maximum of 80 steps per task. All RAG-based agents were initialized with five variations, a total of 150 variations, of rollouts using the golden trajectory for training, while three randomly sampled test variations, a total of 90 variations, were drawn from the ScienceWorld test set. In contrast, all CoT agents were evaluated directly on the randomly drawn test set as intended.

All experiments were conducted on a system equipped with a NVIDIA RTX 3060 GPU, an AMD Ryzen 9 7900X CPU, 64GB RAM, running Ubuntu 23.04 with Python 3.11.0. The full table of hyperparameters and settings for DAVIS is provided in Table 4. Full results is available in table 6, and all code and prompts are available in the attached repository.

Hyperparameter	Value
Maximum Steps per Task	100
Simplification Level	Easy
Knowledge Graph Pipeline	LLaMA3-70B-Instruct
Reasoning Model	GPT-4-Turbo
Maximum QA Turns	5
Predicted Trajectory Length	5

Table 4: Hyperparameter settings for DAVIS.

Figure 5: The ScienceWorld environment

Subject	Description	Tasks
Matter	Agents perform experiments to change the state of various materials, such as transforming ice to water or water to steam	1-1, 1-2, 1-3, 1-4
Thermodynamics	Agents conduct experiments involving temperature manipulation, such as heating or cooling objects.	2-1, 2-2, 2-3
Electricity	Agents relocate to a workshop and construct electrical circuits to achieve specific objectives.	3-1, 3-2, 3-3 3-4
Biology	Agents relocate to a garden and identify animals based on various queries.	4-1, 4-2, 4-3 4-4
Botany	Agents relocate to a greenhouse and perform tasks such as growing plants or observing their growth.	5-1, 5-2
Chemistry	Agents engage in standard chemistry tasks, such as mixing substances to create new compounds	6-1, 6-2, 6-3
Lifespan and Life Stages	Agents observe and report the life stages of plants and animals, such as germination, flowering, or molting.	7-1, 7-2, 7-3 8-1, 8-2
Physics	Agents use physics knowledge to measure angles or explore physical properties of materials	9-1, 9-2 9-3
Genetics	Agents identify genetic traits of plants, such as dominant or recessive characteristics, based on observations.	10-1 10-2

Table 5: Description of subjects and corresponding tasks in ScienceWorld. Each subject represents a unique domain of inquiry, with tasks designed to evaluate agents' reasoning, planning, and execution capabilities in diverse scientific scenarios.

Task	SayCan		ReAct		Reflexion		RAP		DAVIS	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
State of Matter	15.42		17.92		20.83		5.42		49.58	
1-1 (L)	1.67	1.5	2.67	2.5	27.67	41.0	13.33	20.55	25.67	19.6
1-2 (L)	23.33	40.4	25.67	40.2	1.00	1.7	1.67	2.89	70.00	0.0
1-3 (L)	3.33	5.8	19.33	25.3	19.33	25.3	6.67	5.78	32.00	27.7
1-4 (L)	33.33	57.7	24.00	39.0	35.33	56.0	0.00	0.00	70.67	0.6
Thermodynamics	24.89		12.67		10.67		20.44		85.00	
2-1 (M)	6.00	3.0	4.00	3.5	9.00	0.0	30.33	47.43	83.00	29.4
2-2 (M)	7.67	0.6	6.33	0.6	17.33	18.8	8.67	15.02	79.67	35.2
2-3 (L)	61.00	48.3	27.67	39.3	5.67	0.6	22.33	20.40	92.33	13.3
Electricity	21.58		27.00		36.08		43.42		68.50	
3-1 (S)	30.33	40.4	30.33	40.4	23.33	34.5	39.00	33.05	82.33	15.7
3-2 (M)	22.67	26.4	19.33	29.3	14.33	20.6	35.33	27.31	68.67	27.1
3-3 (M)	23.33	27.5	5.00	5.0	39.00	34.5	38.00	35.03	58.33	2.9
Biology	29.92		41.83		91.00		44.42		95.83	
4-1 (S)	11.33	9.8	17.00	0.0	72.33	47.9	61.00	38.1	100.00	0.0
4-2 (S)	36.00	34.8	58.33	28.9	100.00	0.0	19.33	9.8	83.33	14.4
4-3 (S)	22.33	4.6	75.00	0.0	91.67	14.4	58.33	36.0	100.00	0.0
4-4 (S)	50.00	43.3	17.00	0.0	100.00	0.0	39.00	38.1	100.00	0.0
Botany	14.83		40.83		38.17		33.00		26.83	
5-1 (L)	16.67	14.4	9.00	3.6	3.67	4.6	50.00	73.99	35.67	2.9
5-2 (L)	13.00	4.6	72.67	47.3	72.67	47.3	16.00	13.89	18.00	6.2
Chemistry	15.78		19.44		51.44		51.00		53.44	
6-1 (M)	16.67	11.5	23.33	11.5	56.67	37.9	53.33	5.78	36.67	5.8
6-2 (S)	26.33	2.3	20.67	18.0	83.33	28.9	22.67	21.60	53.67	40.5
6-3 (M)	4.33	2.3	14.33	5.1	14.33	7.5	77.00	0.00	70.00	0.0
Lifespan and Life Stages	44.40		35.67		22.47		17.67		57.80	
7-1 (S)	75.00	43.3	66.67	28.9	50.00	0.0	16.67	28.86	100.00	0.0
7-2 (S)	83.33	28.9	66.67	28.9	33.33	14.4	16.67	28.86	83.33	28.9
7-3 (S)	33.00	0.0	22.00	19.1	22.33	9.2	5.67	9.81	83.00	0.0
8-1 (S)	13.33	6.1	15.00	22.6	4.00	4.0	38.00	25.98	2.67	2.3
8-2 (S)	17.33	4.6	8.00	0.0	2.67	4.6	11.33	9.81	20.00	0.0
Physics	7.78		3.89		27.78		34.48		64.44	
9-1 (L)	5.00	5.0	0.00	0.0	36.67	54.8	30.00	30.00	76.67	40.4
9-2 (L)	6.67	7.6	11.67	12.6	8.33	2.9	30.00	0.00	60.00	34.6
9-3 (L)	11.67	16.1	0.00	0.0	38.33	53.5	43.44	23.28	56.67	37.9
Genetics	5.83		25.17		6.33		6.83		72.33	
10-1 (L)	6.00	9.5	39.00	53.5	6.33	9.2	3.33	5.78	100.00	0.0
10-2 (L)	5.67	9.8	11.33	9.8	6.33	9.2	10.33	10.50	44.67	47.9

Table 6: Full results on ScienceWorld. The average score for each category is displayed in the grey bar on the same row as the category label.