M2PA: A Multi-Memory Planning Agent for Open Worlds Inspired by Cognitive Theory

Yanfang Zhou[1]* Xiaodong Li[2]* Yuntao Liu[1]* Yongqiang Zhao[3] Xintong Wang[4]
Zhenyu Li[1] Jinlong Tian[2] Xinhai Xu[1]†

[1]Academy of Military Sciences
[2]National University of Defense Technology
[3]Peking University
[4]Department of Informatics, Universität Hamburg
yanfangzhou08@gmail.com, xuxinhai@nudt.edu.cn

Abstract

Open-world planning poses a significant challenge for general artificial intelligence due to environmental complexity and task diversity, especially in long-term tasks and lifelong learning. Inspired by cognitive theories, we propose M2PA, an open-world multi-memory planning agent. M2PA innovates by combining Large Language Models (LLMs) with human-like multi-memory systems, aiming to fully leverage the strengths of both while mitigating their respective limitations. By integrating the expansive world knowledge and language processing capabilities of LLMs with the perception and experience accumulation abilities of the human memory system, M2PA exhibits situation awareness, and experience generalization capabilities, as well as the potential for lifelong learning. In experiments, M2PA significantly outperforms current state-of-the-art agents across 50 Minecraft tasks in zero-shot learning. In exploratory lifelong learning experiments, M2PA demonstrates its continuous learning ability, achieving a **38.33%** success rate in the "ObtainDiamond" task. Our findings provide a novel paradigm for constructing more effective agents in open-world environments.

1 Introduction

Planning (Russell and Norvig, 2016), a core ability of intelligent agents, involves decomposing complex tasks into sub-goals and generating action sequences for each sub-goal. In open-world environments like Minecraft (Wikipedia contributors, 2024), planning faces new challenges due to complex environments and diverse tasks. Recent work (Wang et al., 2023; BAAI, 2023; Zhu et al., 2023; Liu et al., 2024; Qin et al., 2024) shows that Large Language Models (LLMs) (Achiam et al., 2023; OpenAI, 2023; Guo et al., 2024; Team, 2024) with
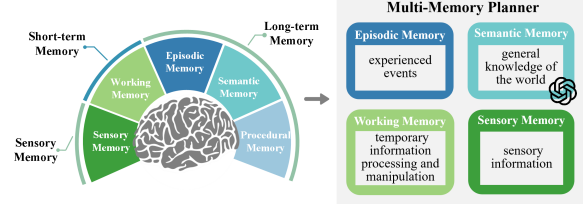

Figure 1: **Multi-Memory Planner Design Inspired by Human Memory System. (Left) Human multi-memory system**: sensory, short-term (working), and long-term (semantic, episodic, procedural) memory. **(Right) Proposed multi-memory planner**: sensory, working, semantic, and episodic memory components. Procedural memory is excluded from the scope of the current study.

their rich knowledge and reasoning capabilities, offer a new paradigm for open-world planning. However, current approaches directly use raw information as the external knowledge library (Wang et al., 2024) or focus only on single-modal memory (Zhu et al., 2023; Qin et al., 2024), failing to directly connect to agents' experiences or generalize knowledge to unfamiliar situations or novel tasks.

Human players in games combine the current situation with past experiences for planning and continuously accumulate experiences to improve task success rates. This observation aligns with research on human cognitive systems (Anderson et al., 2004; Laird, 2019; Kotseruba and Tsotsos, 2020; Wang et al., 2025), particularly memory mechanisms (Tulving et al., 1972; Squire, 2004). Neuroscience and cognitive psychology reveal that humans possess a complex multi-memory system (Figure 1) for information storage and recall. When faced with new problems, humans can flexibly invoke and integrate various memory types, such as semantic memory (general knowledge) and episodic memory (specific experiences) (Tulving, 2002), combining the knowledge required to solve problems with the experiences of the current situation to generate feasible plans. This enables humans to continuously

---

learn, and cope with unfamiliar situations or novel tasks, supporting lifelong learning. However, human semantic memory is inherently limited and leads to constrained planning capabilities.

Inspired by these, we propose the Multi-Memory Planning Agent (M2PA), integrating LLMs with human-like multi-memory systems. M2PA leverages LLMs' expansive world knowledge and language processing capabilities while incorporating perception, task-specific experiences, and lifelong learning capabilities from the multi-memory system. This integration enables M2PA to handle immediate contexts while maintaining consistency with long-term goals and past experiences. It emphasizes *what human multi-memory systems do best* (perception, dynamic knowledge integration, experience utilization, and continuous learning) and *what LLMs do best* (massive knowledge representation and efficient retrieval).

The M2PA design comprises a **semantic memory** module based on LLMs, serving as the primary knowledge source; an **episodic memory** module using vector databases to abstract successful plans into reusable experiences through innovative encoding, retrieval, and updating mechanisms; a **sensory memory** with a visual buffer capturing the current scene; a **working memory** prompter that integrates various memory and feedback, providing richer task prompts for LLMs. This design enables the agent to utilize semantic, episodic, sensory, and working memory for experience accumulation, situation-aware planning, and lifelong learning. Compared to agents based on external knowledge, M2PA offers strong dynamic adaptability (automatic update mechanisms), long-term consistency (decisions and actions are consistent with long-term goals and past experiences), and multimodal information management (seamless integration and utilization of multimodal information).

In experiments, M2PA significantly outperforms current state-of-the-art agents across 50 Minecraft tasks, which include short-term, medium-term, and long-term tasks. Notably, it demonstrates strong planning capabilities in long-term tasks and tackling new challenges without additional training. We also find that the LLM, with its richer world knowledge as semantic memory, boosts planning success rates. In addition, lifelong learning experiments show M2PA's potential for continuous learning through step-by-step exploration of the world and experience accumulation, enhancing its planning ability.

To summarize, our main contributions are:

- **M2PA innovates by integrating LLMs with human-like multi-memory systems, combining the strengths of both to offset the limitations of each:** It combines LLMs' world knowledge and language processing capabilities with human-like perception, experience utilization, and lifelong learning, offering a new paradigm for open-world agents.

- We design an **episodic memory** mechanism for LLM-based agents, abstracting successful plans as experiences using vector databases with an automatic update mechanism. Episodic memory enables experience generalization, supporting lifelong learning.

- **Multimodal dynamic working memory** automatically constructs prompts based on task instructions and environment information, overcoming LLMs' context window limitations and enhancing contextual connectivity between sub-goals.

2 Method

In this section, we first provide an overview of our proposed M2PA. Next, we elaborate on the design of the multi-memory components and then detail how these modules interact within M2PA.

2.1 Overview

Our M2PA comprises three modules: a **Multi-Memory Planner**, a **Reflector**, and a **Controller** (Figure 2 (a)). Unlike other LLM-based agent that use LLMs as central controllers, M2PA positions the LLM as semantic memory. It works in conjunction with an episodic memory based on a vector database, updating the episodic memory repository when plans are successful.

The Multi-Memory Planner aligns with human memory systems, integrating multi-memory information. This approach combines knowledge with experiences and perception with motor capabilities, enhancing planning abilities in open-world environments. Planning involves task decomposition, sub-goal planning, and plan generation. The process could be formulated as follows:

$$P=\{p_1,p_2,...,p_n\}=plan(E,T;\Theta,\mathcal{P}),$$
$$p_i=\{a_{i1},a_{i2},...,a_{it}\}=sub\text{-}plan(E,g_i;\Theta,\mathcal{P}), \quad (1)$$
$$\{g_1,g_2,...,g_n\}=decompose(E,T;\Theta,\mathcal{P}).$$
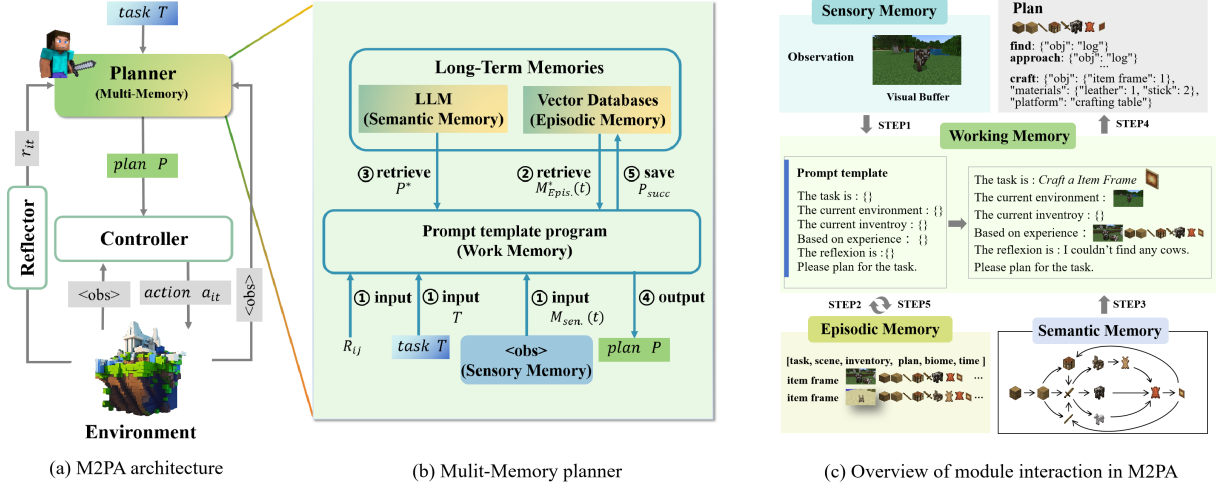
Figure 2: **M2PA Architecture and Module Interaction Overview.** (a) M2PA includes a multi-memory planner for producing plans, a reflector for providing execution feedback, and an action controller. (b) The Multi-memory planner incorporates a semantic memory module leveraging LLMs, an episodic memory module utilizing vector databases, a sensory memory with a visual buffer, and a working memory prompter. (c) After receiving the task instruction, the working memory integrates observations, task information, and planning experiences from episodic memory. Then, it uses a prompt template to guide the semantic memory in generating a plan. Successful plans are subsequently updated in the episodic memory for future reference.

We define E as the environment and T as the task. At time step t, g_i represents a sub-goal, p_i denotes the corresponding sub-plan, and a_{it} is the action taken. Θ and $\mathcal{P}$ represent the parameters of the LLM and the task prompts, respectively.

The Reflector triggers self-reflection on execution performance, either periodically or under specific conditions (such as encountering obstacles or achieving intermediate goals). During reflection, working memory processes and integrates knowledge and experiences from semantic and episodic memory, conducting comprehensive analysis. This design strengthens the agent's self-awareness and adaptive capabilities.

The Controller parses and executes action sequences from the planner. We replace low-level keyboard and mouse operations with high-level actions (such as find, craft, mine, etc.) (Details in Appendix B.2). This approach enables precise handling of environmental observations and operations, enhances the LLM understanding of situations and user intentions, and improves interaction efficiency.

2.2 Multi-Memory Planner

The Multi-Memory Planner, M2PA's core, comprises sensory memory, semantic memory, working memory, and episodic memory (Figure 2 (b)).

2.2.1 Sensory Memory.

M2PA uses self-centric RGB images from Minecraft as scene graphs in its sensory memory. These images capture the environment and entities, similar to human perception, supporting situation-aware planning. We use MineClip (Fan et al., 2022), which performs better than CLIP (Radford et al., 2021) at instruction following, to encode sensory memory for multimodal retrieval.

2.2.2 Semantic Memory

The LLM serves as semantic memory with extensive knowledge and language processing capabilities. Semantic memory, through interactive planning (Pallagani et al., 2024), iteratively decomposes complex tasks into a hierarchical sub-goal tree using environment information and experiences from sensory and episodic memory. Action sequences are generated for each sub-goal (Equation 1), forming a skill tree (Figure 3).

We provide a structured template for goal decomposition (Prompt can be found in Appendix D), including materials, tools, and a set of structured actions that can be executed by the controller. The LLM recursively decomposes tasks (e.g., "*stone pickaxe*" ⚒) using this template until the sub-goals are indivisible (e.g., log 🪵), forming a sub-goal structure tree and skill tree (Figure 3). This enhances plan readability and interpretability, aiding error localization during execution failure. For
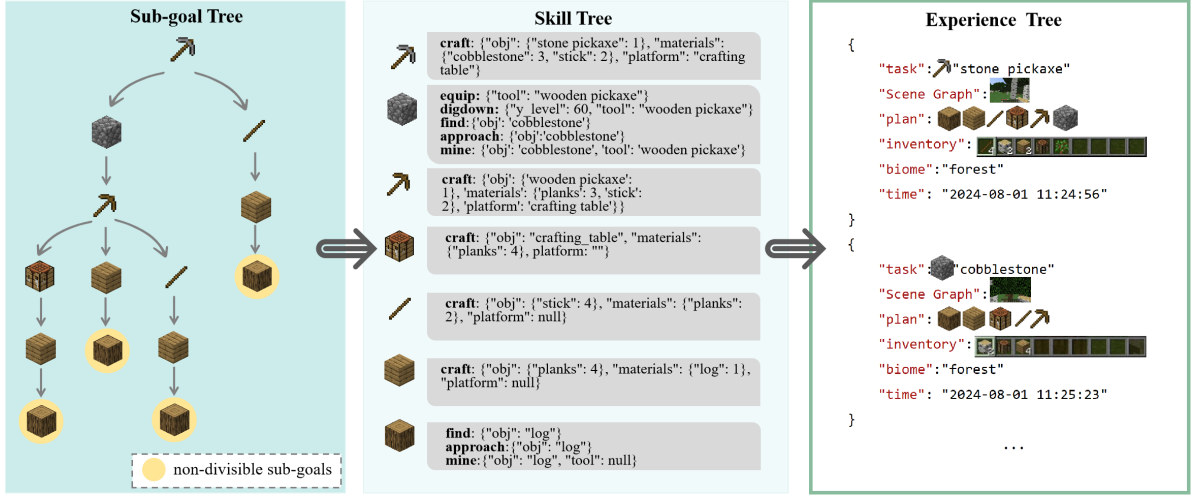
3

Figure 3: **Generating-experiences.** Working memory employs a goal-oriented prompting strategy, guiding the LLM to follow established human logical thinking, iteratively decomposing complex tasks into a sub-goal tree structure, and generating executable action sequences for each sub-goal, forming a skill tree. This skill tree's sequence is executable by the action controller. The experience tree, abstracted from successful plans, aids in guiding subsequent planning tasks.

more details about semantic memory, please check Appendix A.1.

2.2.3 Working Memory

The Working Memory Prompter simulates human short-term memory for processing visual information, retrieving and generating episodic memories, and integrating feedback from the reflector. We design a standardized prompt template that fuses multi-memory data and feedback (Figure 2 (c)). We introduce the "Reference for Selection" method (Gramopadhye and Szafir, 2023) to incorporate past experiences to autoregressively prompt the LLM, which can reduce the sub-goal space. Subsequently, we employ the "Evaluation of Sub-goal Achievement based on Sub-goal Results" method (Madaan et al., 2023) to integrate feedback, enhancing the self-correction capabilities of the LLM. This goal-oriented prompting strategy (Li et al., 2024) can guide the LLM to follow established human logical thinking, efficiently transforming sub-goal and skill trees into experience trees (Figure 3), and storing them in episodic memory for future tasks. See Appendix A.2 for details.

2.2.4 Episodic Memory

Episodic memory, a key M2PA component, stores experiences in vector form to support planning. As episodes accumulate, raw language data causes memory bloat and lower retrieval efficiency. An efficient episodic memory mechanism requires efficient coding, retrieval, and updating mechanisms.
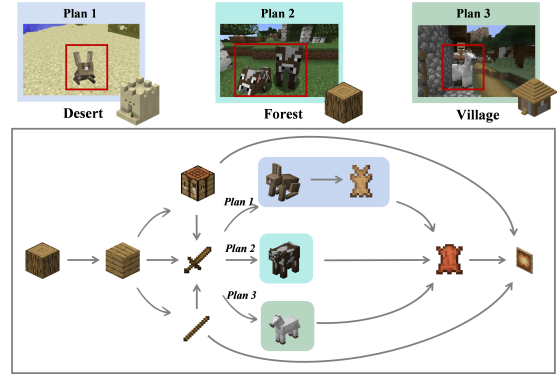

Figure 4: **Feasibility and Quality of Planning are Determined by Multiple Factors.** Including biome, surrounding resource distribution, and agent's current inventory status.

(1) Encoding. Open-world planning (e.g., Minecraft) heavily depends on environmental factors that determine plan feasibility and quality (Figure 4). We encode six key elements in multimodal vectors: [*goal, scene graph, plan, inventory, biome, and time*]. *Scene graphs* come from sensory memory. Cross-modal storage captures key features and relationships. Feature extraction reduces memory usage and improves retrieval efficiency. See Appendix A.4.1 for encoding details.

(2) Retrieval. Human planning primarily relies on situational cues and the recency effect. Situations provide key cues and trigger points for recall, with higher similarity memories more easily activated. We design an algorithm simulating the

4

human memory retrieval process. This algorithm integrates multimodal information from various elements (e.g., inventory, scene graph) through a weighted average approach, forming a comprehensive query vector. Values are then selected based on similarity. The process is described as follows:

$$q = \alpha \cdot q_{context} + (1 - \alpha) \cdot q_{image},$$
$$R = \text{top-k} \left((\arg\max)_{v \in D} \left(\text{similarity}\left(q, v \right) \right) \right). \quad (2)$$

Where the weight α (which is between 0 and 1) is used to fuse $q_{context}$ and q_{image} into a comprehensive query vector q. D represents the set of vectors in the vector database. v is a candidate vector, and R is the retrieved memory. For more details, please check the Appendix A.4.2.

(3) Updating. Memory accumulation may cause reasoning issues. Existing methods, such as ChatDB (Hu et al., 2023) and RET-LLM (Modarressi et al., 2023), use manual deletion and FIFO strategies, respectively, but fail to enable automatic memory replacement. Inspired by human memory reconsolidation and integration, we design two updating modes: real-time and periodic. For more details, please check the Appendix A.4.3.

- **Real-time updating**: New information replaces original memory when similarity reaches a threshold, ensuring timeliness and accuracy.

- **Periodic updating**: The entire memory repository undergoes restructuring and optimization when system resources permit.

M2PA's dual updating mechanism compresses storage space, enhances retrieval efficiency, and preserves core memory information. The formal description follows:

$$M_{update} = \begin{cases} f(M, N) & \text{if similarity} > \theta, \\ g\left(\{M_i\} \right) & \text{if } |\{M_i\}| > \delta. \end{cases} \quad (3)$$

Where $f(\cdot)$ and $g(\cdot)$ denote memory integration functions. M represents existing memories, and N represents new information. The thresholds θ and δ are introduced to represent decision criteria, where $|\{M_i\}|$ indicates the number of memories associated with the same task i.

2.3 Module Interaction

Let's take the example ("CraftFrame" 🗔 in Minecraft) shown in Figure 2 (b)(c) to demonstrate the workflow of module interactions in M2PA.

First, working memory retrieves relevant experiences $M^*_{Epis.}(t)$ from episodic memory $M_{Epis.}(t)$, combining observation $M_{Sen.}(t)$ from sensory memory and task instruction T. Then, working memory generates a multimodal query prompt $\text{Prompt}(t)$ by filling the template with $M^*_{Epis.}(t)$, $M_{Sen.}(t)$, T, and feedback R_{it}. Next, semantic memory $M_{Sem.}$ uses $\text{Prompt}(t)$ to prompt LLM to decompose the complex task and generate a plan P^* 🍖🍖/🪓✂️ 🏹🗡🪵🪵. If P^* succeeds, the corresponding information updates $M_{Epis.}$. If P^* fails, the reflector generates feedback R^*_{it}, triggering plan revision. This process repeats until the task is completed. In this closed-loop process, the LLM functions as both a planner and a reflector.

$$M^*_{Epis.}(t) = \text{retrieve}(M_{Sen.}(t), T; M_{Epis.}),$$
$$\text{Prompt}(t) = \text{combine}(M^*_{Epis.}(t), M_{Sen.}(t), T,$$
$$R_{it}; M_{Work}),$$
$$P^* = \text{plan}(\text{Prompt}(t); M_{Sem.}), \quad (4)$$
$$R^*_{it} = \text{execute}(a_{it}),$$
$$M_{Epis.} = \text{update}(P_{Succ}; M_{Epis.}).$$

3 Experiments

3.1 Experiments Setting

Environment. We leverage OpenAI's GPT-4o (Achiam et al., 2023) and GPT-3.5-turbo (OpenAI, 2023) APIs as M2PA's semantic memory core for text completion. Along with text-embedding-ada-002 (OpenAI, 2022) API for text embedding, enabling efficient storage and retrieval. Integration with MineClip (Fan et al., 2022) vision encoder enhances M2PA's in-game visual information parsing, yielding better results compared to directly using multimodal large models(Qin et al., 2024).

Task Setting. We select 50 tasks from MineDojo (Fan et al., 2022) (Minecraft benchmark) (Wikipedia contributors, 2024). Tasks vary in period and complexity, focusing on obtainable overworld items. Tasks are categorized into five difficulty levels (basic to complex) based on minimum required sub-goals (Appendix C).

Baselines. We compare M2PA with three representative baselines in the experiments. Plan4MC (BAAI, 2023), further integrates LLMs and Reinforcement Learning (RL) (Matsuo et al., 2022) without using memory for planning. DEPS (Wang et al., 2023), utilizes text-based memory as the initial plan, but lacks visual information integration

for planning. MP5 (Qin et al., 2024), combines visual information for situation-aware planning, but lacks scene-related elements when encoding memory.

Evaluation metrics. Agents start in survival mode with empty inventories. Success means obtaining targets within time limits, while death counts as a failure. Each task runs 30 times with random starting positions and seeds. All methods use standardized prompts, feedback templates, and controllers. We report average success rates and include game steps as metrics in lifelong learning experiments.

Task level	Task	Average Success Rate (%)			
		Plan4MC	DEPS	MP5	**M2PA**
Basic	🧰	53.33	80.00	100	**100.00**
	Average	64.33±9.7	91.00±3.3	97.00±2.8	**99.33±2.0**
Easy	🗡️	46.67	63.33	76.67	**93.33**
	Average	43.43±6.3	80.33±5.0	84.67±4.8	**99.27±1.3**
Medium	⛏️	36.66	76.67	73.33	**90.00**
	Average	25.53±8.7	53.67±18.1	71.33±6.5	**84.33±5.4**
Hard	▯	16.67	20.00	43.33	**50.00**
	Average	15.07±6.2	19.20±4.5	43.00±4.8	**52.00±7.2**
Complex	💠	2.00	6.00	16.00	**33.33**
	Average	0.20±0.6	0.80±1.8	9.67±5.9	**20.20±7.7**

Table 1: **M2PA and baselines performance on Minecraft tasks.** Detailed results for each task can be found in Appendix C.

3.2 Main Results

Table 1 shows experimental results across difficulty levels (Detailed accuracy in Appendix C). M2PA performs best in all categories. For Basic and Easy tasks, all methods perform similarly, with M2PA slightly ahead (average success rate over 99%). As task complexity increases, M2PA's advantage grows, with a success rate of 52%, nearly three times that of DEPS (Wang et al., 2023) and Plan4MC (BAAI, 2023). For Complex tasks, M2PA achieves a 20.20% success rate, twice that of MP5, and outperforms other methods. Its excellent performance in Hard and Complex tasks demonstrates strong long-term planning and environmental adaptability.

3.3 Ablation Study

We conduct ablation studies to evaluate the effectiveness of various modules in the multi-memory system (Table 2). Our analyses are as follows:

Semantic Memory. Using LLM alone achieves 33.33% success rate for "*Crafting Table*" 🧰, show-

Sema. Mem.	Sens. Mem.	Work Mem.	Epis. Mem.	Average Success Rate (%)				
				🧰	⛏️	⛏️	⛏️	⛏️
✓				33.33	16.00	0.00	0.00	0.00
✓	✓			66.67	33.33	0.00	0.00	0.00
✓	✓	✓		93.33	83.33	66.67	43.33	6.67
✓	✓	✓	✓	100.00	100.00	90.00	76.67	16.67

Table 2: **Ablation study.** Sema. Mem.: Semantic Memory; Sens. Mem.: Sensory Memory; Work Mem.: Working Memory; Epis. Mem.: Episodic Memory.

ing planning potential. **Sensory Memory.** Agents with sensory memory show higher success rates, demonstrating better environmental understanding. **Working Memory.** Agents with working memory first complete "*Diamond pickaxe*" ⛏️, highlighting its role in memory integration and feedback processing. It provides better prompts and helps correct errors, crucial for long-term tasks. **Episodic Memory.** Agents with episodic memory nearly triple the success rate for the "*Diamond pickaxe*" ⛏️, showing experience's significant impact on performance.

3.4 Exploring Lifelong Learning Capabilities

In this section, we further explore M2PA's lifelong learning capabilities.

3.4.1 The Compensatory Effect of Episodic Memory on Semantic Memory

We tested four M2PA configurations: GPT-3.5-turbo and GPT-4o, both with and without episodic memory. Results in Figure 5 show performance increases in order: GPT-3.5-turbo < GPT-4o < GPT-3.5-turbo + episodic memory < GPT-4o + episodic memory. Meanwhile, the play-game steps decreased in the same order. These findings demonstrate tha LLMs with richer world knowledge and language processing capabilities (serving as semantic memory) improve planning success and efficiency. Episodic memory provides complementary benefits to semantic memory, and together these memory mechanisms significantly enhance LLM planning capabilities.

3.4.2 The Impact of Situation-Awareness on Planning Efficiency

We find that tasks show higher success rates above ground than underground at the same difficulty level. Task difficulty increases with depth, likely due to higher scene similarity. Deeper environments are harder to distinguish, limiting planning effectiveness (Figure 6).
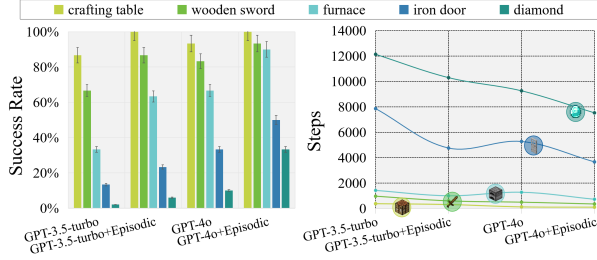
Figure 5: **Success rates and play-game steps for various configurations and task terms.** We find that episodic memory compensates for semantic memory limitations.

ground   y_level=50   y_level=16

Figure 6: **Screenshots of scenes with varying recognizability.** We present three screenshots of scenes at different depths above and below ground.

To quantify visual recognizability's impact, we compare CLIP (Radford et al., 2021) (simulating low-recognizability underground) and MineClip (Fan et al., 2022) (simulating high-recognizability above-ground). Analysis of equal-difficulty tasks confirms situation awareness significantly affects success rates (Table 3).

| Visual	Average Success Rate(%)				
Encode	Basic	Simple	Medium	Hard	Complex
Clip (Radford et al., 2021)	97.27	88.10	58.50	36.70	12.90
MineClip (Fan et al., 2022)	99.33	99.27	84.33	52.00	20.20

Table 3: Success rates for different visual encoding models.

3.4.3 Mechanistic Analysis of Episodic Memory Effectiveness

To investigate episodic memory mechanisms, we analyze three aspects:

(1) Success Rate with Episodic Memory. We introduced two metrics: overall success rate $R_{overall}$, and conditional success rate $R_{condition}$ (Bishop and Nasrabadi, 2006). The latter measures the success rate after the first successful execution, i.e., after acquiring episodic memory.

$$R_{overall} = \frac{N_s}{N_t} \, , \; R_{condition} = \frac{N_{s|c}}{N_{t|c}}. \qquad (5)$$

In this equation, N_s represents successful attempts, N_t denotes total attempts, $N_{s|c}$ refers to

successful attempts under condition c, and $N_{t|c}$ indicates total attempts under condition c.
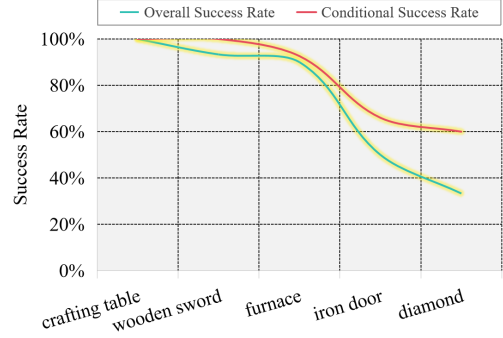
Figure 7: **Overall success rate and conditional success rate for typical items in each task group.** We report two metrics for typical items.

Figure 7 shows $R_{overall}$ significantly declines as task difficulty increases. However, $R_{condition}$ remains less affected by task difficulty, nearly doubling compared to $R_{overall}$ in "*ObtainDiamond*"🟢. This indicates episodic memory significantly improves overall planning stability.

(2) Impact of Scattered Memory Fragments on Comprehensive Success. Experiments on the "*ObtainDiamond*"🟢 (12 sub-goals) reveal a positive correlation between the number of sub-goals in episodic memory and success rate (Figure 8).

We do not directly store information related to "*ObtainDiamond*"🟢, but instead use the experiences of its sub-goals, encoding their key elements into episodic memories. This is not mere data reuse. The increased planning success rate observed with more sub-goals memories indicates that M2PA can solve unfamiliar scenarios and tasks through experience generalization.
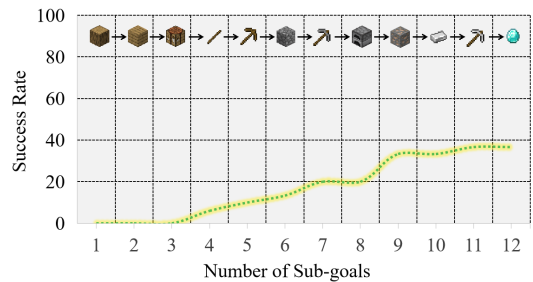
Figure 8: **Impact of a number of sub-goals in episodic memory on success rate.** Taking "*ObtainDiamond*"🟢 as an example, we report its success rate as the number of sub-goals increases.

(3) Impact of Episodic Memory Capacity. We assess episodic memory capacity's impact (representing different learning stages) on M2PA's perfor-

mance, setting a 300-entry memory limit. Figure 9 shows success rates increase with memory capacity across tasks. Short-term tasks (Basic, Easy, and Medium) exhibit a saturation effect, reaching high, stable success rates at specific memory thresholds, while long-term tasks (Hard and Complex) keep improving, even when reaching the capacity limit, with the task of "*ObtainDiamond*" 💠 reaching 38.33%. This improvement likely comes from M2PA's built-in memory update mechanism dynamically merges and compresses similar experiences. This shows M2PA's growing planning abilities and lifelong learning potential.
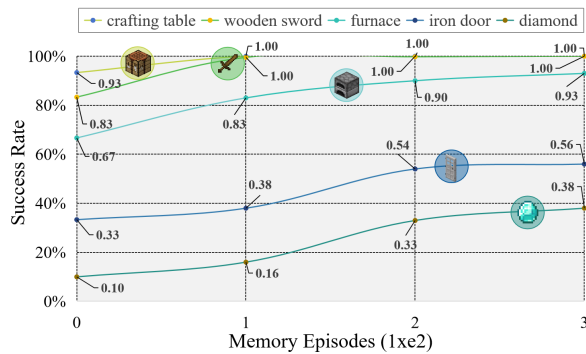
Figure 9: **The impact of memory capacity on M2PA success rates.** We report how the success rates of typical items change with varying memory capacity.

4 Related Work

4.1 Planning with LLM-based Agent

Traditional methods like symbolic planning have required environment modeling and have been limited to handling a single task in open worlds, restricting generalization (Benyamin et al., 2023) (Wichlacz et al., 2019). Reinforcement learning (RL) (Matsuo et al., 2022) has emerged as the dominant approach for creating game-playing agents, while RL agents struggle with open-world tasks due to their high computational overhead and limited task generalization (Baker et al., 2022; Hafner et al., 2023).

LLMs have leveraged their vast knowledge and language processing abilities to provide a new paradigm for planning. Recent work (Wang et al., 2023) (Liu et al., 2024) (BAAI, 2023) (Zhu et al., 2023) has incorporated LLMs as planning and reflection modules but has lacked visual information integration and experience utilization for planning. GITM (Zhu et al., 2023) and DEPS (Wang et al., 2023) have directly used text-based memory as the

initial plan, which limits the comprehensive utilization of experience, especially in situation-aware planning. Jarvis-1 (Wang et al., 2024) has utilized raw multimodal data but has lacked abstract integration of memory, resulting in insufficient utilization of experience. MP5 (Qin et al., 2024) combines visual information but lacks scene-related elements when encoding memory. Despite LLM-based agents having extensive world knowledge, they still face limitations in situation awareness, experience utilization, and lifelong learning.

4.2 Human Memory Systems

The human memory system consists of three main types: sensory memory (ultra-short-term), working memory (short-term), and long-term memory. Long-term memory (Tulving et al., 1972) has two components: semantic and episodic. Semantic memory stores general knowledge and supports basic reasoning (Collins and Quillian, 1969; McClelland and Rogers, 2003), which aligns with the capabilities of LLMs. Episodic memory stores personal experiences, and its absence can limit learning and adaptation (Rosenbaum et al., 2005). Human episodic memory and semantic memory work together, enabling the effective use of past experiences and broad knowledge when facing new challenges. Additionally, sensory memory captures immediate perceptual information, while working memory (Baddeley, 1992; Roth and Courtney, 2007) processes and integrates it with past experiences and knowledge for decision-making. This collaboration enables humans to continuously learn, supporting lifelong learning.

5 Conclusion

This paper presents M2PA, a multi-memory planning agent for open-world environments. M2PA aligns with human multi-memory functions, integrating semantic and episodic memory modules to leverage situation awareness, past experiences, and knowledge to enhance decision-making and planning abilities. Experiments demonstrate M2PA's efficiency across short-term, medium-term, and long-term Minecraft tasks. It significantly outperforms existing baselines in zero-shot learning. Furthermore, M2PA exhibits lifelong learning abilities through dynamic memory updates and optimization. Future research will explore M2PA's multi-modal capabilities. It will also refine memory integration and planning abilities, to advance

the development of general artificial intelligence.

6 Limitations

Our M2PA aims to enhance open-world planning performance by combining the strengths of human memory systems with LLMs. Although replanning rounds and token costs were not formally evaluated as primary metrics, preliminary testing indicates that M2PA performs comparably to baseline models. A key limitation stems from the probabilistic nature of LLMs' reasoning process, which causes prerequisite identification errors, resulting in incorrect feedback loops and unnecessary plan revisions. Although incorporating rule-based reasoning or structured knowledge could potentially address these issues, such modifications could impair generalization. Future work will explore the trade-off between symbolic reasoning and computational overhead, aiming to improve computational efficiency while maintaining robust generalization across diverse scenarios.

7 Ethics Statement

Inspired by cognitive theory, we propose M2PA, a planning agent for open-world tasks. It innovates by integrating LLMs with human-like multi-memory systems, combining the strengths of both to offset the limitations of each. It combines LLMs' world knowledge and language processing capabilities with human-like perception, experience utilization, and lifelong learning, offering a new paradigm for open-world agents. Our experiments were conducted in MineDojo, an open-source simulation framework for Minecraft that provides a wide range of procedurally generated tasks and a comprehensive benchmark suite. MineDojo is built upon MIT-licensed code and APIs, ensuring ethical compliance and research reproducibility. Additionally, we have submitted our code to ensure that researchers and practitioners can easily access and implement our methods.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

John R Anderson, Daniel Bothell, Michael D Byrne, Scott Douglass, Christian Lebiere, and Yulin Qin.

2004. An Integrated Theory of the Mind. *Psychological review*, 111(4):1036.

PKU BAAI. 2023. Plan4MC: Skill Reinforcement Learning and Planning for Open-World Minecraft Tasks. *arXiv preprint arXiv:2303.16563*.

Alan Baddeley. 1992. Working Memory. *Science*, 255(5044):556–559.

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. 2022. Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 24639–24654.

Yarin Benyamin, Argaman Mordoch, Shahaf S Shperberg, and Roni Stern. 2023. Model Learning to Solve Minecraft Tasks. In *PRL Workshop Series Bridging the Gap Between AI Planning and Reinforcement Learning*.

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern Recognition and Machine Learning*, volume 4. Springer.

Allan M Collins and M Ross Quillian. 1969. Retrieval Time from Semantic Nemory. *Journal of verbal learning and verbal behavior*, 8(2):240–247.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. 2022. MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. In *Proceedings of the 36th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 18343–18362.

Maitrey Gramopadhye and Daniel Szafir. 2023. Generating Executable Action Plans with Environmentally-Aware Language Models. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3568–3575. IEEE.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, YK Li, et al. 2024. DeepSeek-Coder: When the Large Language Model Meets Programming–The Rise of Code Intelligence. *arXiv preprint arXiv:2401.14196*.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. 2023. Mastering Diverse Domains through World Models. *arXiv preprint arXiv:2301.04104*.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. 2023. ChatDB: Augmenting LLMs with Databases as Their Symbolic Memory. *arXiv preprint arXiv:2306.03901*.

Iuliia Kotseruba and John K Tsotsos. 2020. 40 Years of Cognitive Architectures: Core Cognitive Abilities and Practical Applications. *Artificial Intelligence Review*, 53(1):17–94.

John E Laird. 2019. *The Soar Cognitive Architecture*. MIT press.

Haochen Li, Jonathan Leung, and Zhiqi Shen. 2024. Towards Goal-oriented Large Language Model Prompting: A Survey. *arXiv preprint arXiv:2401.14043*.

Shaoteng Liu, Haoqi Yuan, Minda Hu, Yanwei Li, Yukang Chen, Shu Liu, Zongqing Lu, and Jiaya Jia. 2024. RL-GPT: Integrating Reinforcement Learning and Code-as-policy. *arXiv preprint arXiv:2402.19299*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-Refine: Iterative Refinement with Self-Feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 46534–46594.

Yutaka Matsuo, Yann LeCun, Maneesh Sahani, Doina Precup, David Silver, Masashi Sugiyama, Eiji Uchibe, and Jun Morimoto. 2022. Deep Learning, Reinforcement Learning, and World Models. *Neural Networks*, 152:267–275.

James L McClelland and Timothy T Rogers. 2003. The Parallel-Distributed Processing Approach to Semantic Cognition. *Nature reviews neuroscience*, 4(4):310–322.

Ali Modarressi, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. 2023. RET-LLM: Towards a General Read-Write Memory for Large Language Models. *arXiv preprint arXiv:2305.14322*.

OpenAI. 2022. New and improved embedding model. https://openai.com/index/new-and-improved-embedding-model/.

OpenAI. 2023. Introducing ChatGPT. https://openai.com/index/chatgpt/.

Vishal Pallagani, Bharath Chandra Muppasani, Kaushik Roy, Francesco Fabiano, Andrea Loreggia, Keerthiram Murugesan, Biplav Srivastava, Francesca Rossi, Lior Horesh, and Amit Sheth. 2024. On the Prospects of Incorporating Large Language Models (LLMs) in Automated Planning and Scheduling (APS). In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, volume 34, pages 432–444.

Yiran Qin, Enshen Zhou, Qichang Liu, Zhenfei Yin, Lu Sheng, Ruimao Zhang, Yu Qiao, and Jing Shao. 2024. MP5: A Multi-modal Open-ended Embodied System in Minecraft via Active Perception. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16307–16316. IEEE.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR.

R Shayna Rosenbaum, Stefan Köhler, Daniel L Schacter, Morris Moscovitch, Robyn Westmacott, Sandra E Black, Fuqiang Gao, and Endel Tulving. 2005. The Case of KC: Contributions of A Memory-Impaired Person to Memory Theory. *Neuropsychologia*, 43(7):989–1021.

Jennifer K Roth and Susan M Courtney. 2007. Neural System for Updating Object Working Memory from Different Sources: Sensory Stimuli or Long-Term Memory. *Neuroimage*, 38(3):617–630.

Stuart J Russell and Peter Norvig. 2016. *Artificial Intelligence: A Modern Approach*. Pearson.

Larry R Squire. 2004. Memory Systems of the Brain: A Brief History and Current Perspective. *Neurobiology of learning and memory*, 82(3):171–177.

Qwen Team. 2024. Qwen2. 5 Technical Report. *arXiv preprint arXiv:2412.15115*.

Endel Tulving. 2002. Episodic Memory: From Mind to Brain. *Annual review of psychology*, 53(1):1–25.

Endel Tulving et al. 1972. Episodic and semantic memory. *Organization of memory*, 1(381-403):1.

Xintong Wang, Jingheng Pan, Liang Ding, Longqin Jiang, Longyue Wang, Xingshan Li, and Chris Biemann. 2025. Cogsteer: Cognition-inspired selective layer intervention for efficiently steering large language models. In *Findings of the Association for Computational Linguistics (ACL)*.

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, Yitao Liang, and Team CraftJarvis. 2023. Describe, Explain, Plan and Select: Interactive Planning with Large Language Models Enables Open-World Multi-Task Agents. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 34153–34189.

Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, et al. 2024. JARVIS-1: Open-World Multi-task Agents with Memory-Augmented Multimodal Language Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Julia Wichlacz, Alvaro Torralba, and Jörg Hoffmann. 2019. Construction-Planning Models in Minecraft. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pages 1–5.

Wikipedia contributors. 2024. Minecraft. https://en.wikipedia.org/wiki/Minecraft.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. 2023. Ghost in the Minecraft: Generally Capable Agents for Open-World Environments via Large Language Models with Text-based Knowledge and Memory. *arXiv preprint arXiv:2305.17144*.

A Implementation Details of M2PA

A.1 Semantic Memory

Semantic memory functionality, primarily implemented through Large Language Models (LLMs), plays an essential role in our system. We integrate OpenAI's GPT-4o and GPT-3.5-turbo via API interfaces for complex text completion tasks. By combining sensory and episodic memory with situation awareness and experiences, we address complex tasks through interactive planning. Our approach decomposes large tasks into manageable sub-tasks using a goal decomposition template, guiding LLMs to create a hierarchical sub-goal tree structure. LLMs then generate executable actions for each sub-goal, forming a skill tree that enables progressive task completion while maintaining focus on overall objectives. This method enhances problem-solving capabilities across varying task complexities.

Experiments reveal that GPT-3.5-turbo, despite inferior inferential capabilities compared to GPT-4o, significantly improves when combined with episodic memory. This finding highlights episodic memory's compensatory role for semantic memory, particularly in enhancing planning capabilities for complex tasks, bringing GPT-3.5-turbo's performance closer to that of GPT-4o.

A.2 Working Memory

The working memory prompt simulates human short-term memory, processing visual buffer information, retrieving and generating episodic memories, and integrating reflector feedback. It maintains and manipulates information over short periods, providing contextual support for M2PA's decision-making and actions.

For sensory memory processing, we adopt Mineclip over CLIP (Radford et al., 2021) due to its superior performance in instruction following strategies. MineClip (Fan et al., 2022) efficiently retrieves relevant episodes from episodic memory through encoded visual buffer information, enabling M2PA to associate current visual inputs with past experiences for contextually relevant decisions.

We design a standardized prompt template to integrate multiple memories and feedback efficiently. Two innovative methods are introduced: (1) Reference for selection, which prompts LLMs autoregressively by incorporating historical experience, reducing sub-goal space and enabling faster,

more accurate sub-goal generation; (2) Evaluation of Sub-goal Achievement based on sub-goal Results, integrating feedback into prompts to enhance LLMs' self-correction ability. This method allows real-time strategy adjustments, improving system adaptability and robustness. Our goal-oriented prompting strategy guides LLMs to follow established human logical thinking patterns, enhancing system transparency and interpretability. It facilitates the efficient transformation of sub-goal and skill trees into experience trees, improving learning efficiency and enabling effective storage of accumulated experiences in episodic memory.

A.3 Sensory Memory

M2PA utilizes RGB images as sensory memory, stored in the visual buffer. These images capture the environment, objects, and entities, approximating human perception. The working memory employs a MineClip (Fan et al., 2022) visual encoder to encode sensory memory for multimodal vector retrieval and storage.

A.4 Episodic Memory

A.4.1 Encoding

In open-world environments like Minecraft, planning is highly context-dependent. Even simple tasks have multiple planning paths, determined by biome, resource distribution, and inventory status. Resource availability varies across biomes, requiring flexible acquisition strategies. To address this contextualized planning problem while avoiding memory explosion and low retrieval efficiency, we propose an innovative experience abstraction and encoding method. We encode six key elements: [goal, scene graph, plan, inventory, biome, and time] into structured experiences stored in episodic memory. This method enables effective interpretation of plan generation reasons and retrieval of relevant past experiences. The encoded elements capture context-dependency and environment-relatedness in planning:

Goal: Defines the specific task. Scene Graph: Provides environmental information. Plan: Record steps to achieve the goal. Inventory: Reflects available resources. Biome: Determines resource distribution. Time: Serves as a marker for memory updating.

This structured encoding mechanism improves planning efficiency, adaptability, and interpretability. It provides a foundation for efficient and reliable planning in complex, dynamic open-world

environments, enabling wiser and more flexible decision-making.

A.4.2 Retrieval

Human episodic memory retrieval relies on situational cues and the recency effect. Environmental context triggers recall, activating highly related memories. The recency effect prioritizes temporally similar situations. We emulate this process by searching for successful past plans, combining task, scene, inventory, and time information. The most recent successful plan serves as the initial plan for the next round. The implementation process involves: (1)Fusing multimodal information through weighted averaging to form a comprehensive query vector. (2)Returning memories meeting a similarity threshold (score below 0.05). (3)Resorting results using the time element, selecting the most recent record. When no similarity scores are below 0.05, indicating no relevant knowledge, LLMs plan based on sensory memory and objective knowledge. This refined search and decision-making mechanism enhances planning decisions across various task scenarios, improving overall task execution efficiency and success rate.

A.4.3 Updating

Human memory is a dynamic reconstruction process, continuously influenced by time and context. This leads to constant updating of episodic memories, contributing to a coherent self-historical cognition.

Memory updating occurs through reconsolidation and integration. Reconsolidation allows new information to merge with retrieved memories before re-storage. Integration consolidates similar memories into a unified representation, reducing redundancy and improving retrieval efficiency. Our updating process design draws on these mechanisms:

(1)Real-time updating: Mimicking reconsolidation, high-similarity memories are merged, with new memories replacing similar old ones. (2)Periodic updating: Inspired by integration, the entire memory library is reconstructed at regular intervals when resources permit. Using the Top-K method (K=3), the three most similar memories are integrated, with the most recent record replacing the others.

(2) Retrieval. Human planning primarily relies on situational cues and the recency effect. Situations provide key cues and trigger points for recall,

with higher similarity memories more easily activated. We design an algorithm simulating the human memory retrieval process. This algorithm integrates multimodal information from various elements (e.g., inventory, scene graph) through a weighted average approach, forming a comprehensive query vector. Values are then selected based on similarity. The process is described as follows:

A.4.4 Storage

Memory storage preserves information over time. Embedding technology, leveraging LLMs' vector embedding capabilities, has become the mainstream approach for memory representation. This method encodes raw text into fixed-dimensional vector space, offering two key advantages:

(1)Data knowledgeization: Vector databases unify multi-modal data (text, images, audio) into vector representations, enabling seamless integration and efficient cross-modal retrieval. This approach captures key features and intrinsic relationships, achieving deep data alignment and integration. (2)Storage optimization: Vector representations, as compressed forms of data, significantly reduce storage requirements and computational overhead, improving overall system efficiency.

These embeddings enable efficient similarity-based retrieval using distance metrics. Vector databases, unlike traditional relational databases, support storage and retrieval of embedded representations, offering flexible solutions for complex data processing. We employ OpenAI's text-embedding-ada-002 and MineClip to transform empirical data into embeddings for vector database storage.

B Environment Setting

B.1 Observation Space

We impose significant restrictions on environmental information to achieve embodied agent-like perception. Unlike omniscient perception methods (e.g., LiDAR rays in GIMT (Zhu et al., 2023)), our agent perceives its surroundings through egocentric RGB images, mimicking human perception. The observation space consists of two components:

Perceptual observations:

Egocentric, Minecraft-style RGB images (sensory memory) 3x3x3 voxels encountered by the agent (object properties)

Status observations:

Relevant ancillary text information (health statistics, inventory details) This dual observation sys-

Index	Action	Argument	Description
1	Find	Object	Navigate the environment to locate a target object
2	Approach	Object	Move towards a target object
3	Mine	Object, tool	Using a tool to break down a block or object
4	Craft/Smelt	Object, platform	Combine raw materials or items at a crafting table or furnace to create a new item
5	Attack	Object, tool	Attack entity
6	Equip	Object	Equip or hold a specific item or tool in the player's hand
7	Dig-down	Y-level, tool	Dig or excavate downwards to a specified y-coordinate using a tool
8	Dig-up	Tool	Dig or excavate upwards using a tool

Table 4: **The definition of the Action Space we use in MineDojo simulator.**

tem enables nuanced, context-aware environmental interaction, aligning with human sensory integration for comprehensive understanding.

B.2 Action Space

The Controller module executes action sequences defined in Table 4, formed through the MineDojo API and refined via environmental interactions. The "find" action, for instance, involves directionless movement, with the agent adjusting its orientation based on MineDojo's field of view information to approach targets.

C Minecraft Task Details

We analyze MineDojo tasks, categorizing them by minimum required sub-goals (Table 5). The actual sub-goal count may vary based on the initial state and biome. We select 50 tasks in ascending order of minimum sub-goals.

To validate M2PA, we choose tasks across five complexity levels: basic, easy, medium, hard, and complex. Tasks are defined by level, name, minimum sub-goals, tools, platforms, initial inventory, and maximum episode steps. Episode time limits range from 12,000 (basic) to 36,000 (complex) steps.

Given Minecraft's open-world nature, we conduct 30 tests per task with randomized initial positions and environment seeds. We report mean success rates per difficulty level. The agent starts in survival mode with an empty inventory. Success requires obtaining the target within the time limit; agent death results in failure. The experimental results of all tasks are shown in Table 6.

D Prompt

M2PA uses GPT-4o and GPT-3.5-turbo APIs as semantic memory cores for interactive planning with LLMs, enabling task decomposition and feedback explanation. Prompts consist of 'SYSTEM' directives and 'USER' inquiries in Table 7.

Task level	Task name	Sub-goal Num.	Tools/Platform	Step Setting
Basic	mine log	1	-	
	mine sand	1	-	
	mine sapling	1	-	
	mine wheat seeds	1	-	
	mine dirt	1	-	
	mine grass	1	-	12K
	craft plank	2	-	
	craft stick	3	-	
	craft button	3	-	
	craft crafting table	3	-	
Easy	craft chest	4	crafting table	
	craft bowl	4	crafting table	
	craft boat	4	crafting table	
	craft wooden slab	5	crafting table	
	craft wooden pressure plate	5	crafting table	
	craft ladder	5	crafting table	12K
	craft barrel	5	crafting table	
	craft wooden axe	5	crafting table	
	craft wooden pickaxe	5	crafting table	
	craft wooden sword	5	crafting table	
Medium	mine cobblestone	6	wooden pickaxe	
	mine coal ore	7	wooden pickaxe	
	craft furnace	7	crafting table	
	craft lever	7	crafting table	
	craft stone pickaxe	7	crafting table	
	craft stone axe	7	crafting table	24K
	craft stone hoe	7	crafting table	
	craft stone shovel	7	crafting table	
	craft stone sword	7	crafting table	
	mine iron ore	8	stone pickaxe	
Hard	smelt glass	9	furnace	
	smelt iron ingot	10	furnace	
	craft iron bars	11	crafting table	
	craft carpentry table	11	crafting table	
	craft iron pickaxe	11	crafting table	
	craft iron door	11	crafting table	36K
	craft iron trapdoor	11	crafting table	
	craft rail	11	crafting table	
	craft cauldron	11	crafting table	
Complex	obtain diamond	12	iron pickaxe	
	mine redstone	12	iron pickaxe	
	craft dropper	13	crafting table	
	craft redstone torch	13	crafting table	
	craft compass	13	crafting table	
	craft clock	13	crafting table	36K
	craft piston	13	crafting table	
	craft diamond pickaxe	13	crafting table	
	craft diamond sword	13	crafting table	
	craft raw gold block	13	crafting table	

Table 5: **The details of all the tasks.**

Task level	Task	Success rate(%)			
		DEPS	Plan4MC	MP5	M2PA(our)
Basic	mine log	93.33	73.33	96.67	100.00
	mine sand	86.67	60.00	100.00	93.33
	mine sapling	93.33	66.67	93.33	100.00
	mine wheat seeds	93.33	60.00	96.67	100.00
	mine dirt	96.67	80.00	96.67	100.00
	mine grass	93.33	80.00	100.00	100.00
	craft plank	90.00	60.00	93.33	100.00
	craft stick	86.67	56.67	100.00	100.00
	craft button	90.00	53.33	93.33	100.00
	craft crafting table	86.67	53.33	100.00	100.00
Easy	craft chest	86.67	53.33	90.00	100.00
	craft bowl	83.33	50.00	90.00	100.00
	craft boat	86.67	43.33	86.67	100.00
	craft wooden slab	76.67	46.67	80.00	100.00
	craft wooden pressure plate	86.67	43.33	90.00	100.00
	craft ladder	80.00	33.33	86.67	100.00
	craft barrel	73.33	50.00	86.67	100.00
	craft wooden axe	76.67	43.33	80.00	96.67
	craft wooden pickaxe	73.33	33.33	80.00	96.67
	craft wooden sword	80.00	46.67	76.67	100.00
Medium	mine cobblestone	76.67	40.00	80.00	93.33
	mine coal ore	73.33	33.33	76.67	86.67
	craft furnace	76.67	36.66	73.33	90.00
	craft lever	66.67	20.00	73.33	86.67
	craft stone pickaxe	60.00	16.00	76.67	83.33
	craft stone axe	40.00	33.33	73.33	86.67
	craft stone hoe	43.33	20.00	60.00	80.00
	craft stone shovel	33.33	16.00	66.67	80.00
	craft stone sword	40.00	20.00	73.33	83.33
	mine iron ore	26.67	20.00	60.00	73.33
Hard	smelt glass	16.67	20.00	50.00	66.67
	smelt iron ingot	26.67	23.33	46.67	60.00
	craft iron bars	23.33	16.67	40.00	56.67
	craft carpentry table	23.33	16.67	50.00	53.33
	craft iron pickaxe	16.00	20.00	43.33	53.33
	craft iron door	20.00	16.67	43.33	50.00
	craft iron trapdoor	13.33	16.00	40.00	43.33
	craft rail	23.33	13.33	43.33	46.67
	craft cauldron	16.00	6.00	33.33	46.67
	craft minecart	13.33	2.00	40.00	43.33
Complex	obtain diamond	6.00	2.00	16.00	33.33
	mine redstone	2.00	0.00	20.00	33.33
	craft dropper	0.00	0.00	13.33	23.33
	craft redstone torch	0.00	0.00	6.00	20.00
	craft compass	0.00	0.00	13.33	23.33
	craft clock	0.00	0.00	6.00	16.00
	craft piston	0.00	0.00	10.00	16.00
	craft diamond pickaxe	0.00	0.00	10.00	13.33
	craft diamond sword	0.00	0.00	0.00	10.00
	craft raw gold block	0.00	0.00	2.00	13.33

Table 6: **Success rates in all the tasks.** Each task is tested for 30 episodes.

SYSTEM:

You are a helpful planner in Minecraft, and good at planning workflows to complete tasks. You need to generate the sequences of goals for a certain task in Minecraft.

I will give you a task, for which you need to conceive a plan, and then create a workflow composed of a sequence of various actions to complete this task.

I will give you the following information:

task information:

- task: The name of the task.
- quantity: The required quantity for the task.
- material: The necessary materials for achieving the task in your inventory.
- tool: The primary tool necessary for this task, for instance a wooden pickaxe. If there are multiple tools, list only the most fundamental one.
- platform: The crafting station or block that is necessary for this task, for instance a crafting table or a furnace.

current environment information: [The image input is a vector encoded by MineClip.]

inventory: a dict representing the inventory, whose keys are the names of the objects and the values are their quantities.

RESPONSE FORMAT

{ "explanation": "explain why the last action failed, set to empty string for the first planning",

"workflow": [{"times": "the number of times the actions will perform", "actions": [{"name": "action name", "args": {"arg name": value}}, ...]}

{"times": "the number of times the actions will perform", "actions": [{"name": "action name", "args": {"arg name": value}}, ...]}]}

Here are some examples for planning workflow:

Example

INPUT:

task information:

- task: iron pickaxe.
- quantity: 1.
- material: {"iron ingot": 3, "stick": 2 }
- tool: None
- platform: crafting table
- tips: 1 iron pickaxe can be crafted with 3 iron ingots and 2 sticks as the material and crafting table as the platform.

current environment information: - []The scene graph encoded by MineClip.]

inventory: {"iron ingot": 4, "stick": 3, "crafting table": 1}

RESPONSE:

{

"explanation": "...",

"workflow": [

{"times": "1", "actions": [{"name": "craft", "args": {"obj": "iron pickaxe", "materials": {"iron ingot": 3, "stick": 2 }, platform: "crafting table"} }]}]

}

USER:

My information is as follows:

- The task is:

{task_information}

- The current environment:

{current_environment_information}

- The current inventory: {inventory}

-Based on experience: {episodic_memory}

-The reflection is: {feedback}

Please plan for the task.

Table 7: **Prompt for M2PA in Minecraft tasks.**