Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Single-Thinking-Process-Based Learning of Complete Issue-Related Materials for Autonomous Agents

Hong Su

suguest@126.com

Chengdu University of Information Technology https://orcid.org/0000-0003-2750-7808

Research Article

Keywords: Large Language Models (LLMs), Single-Thinking-Process-Based Learning, Complete Issue-Related Materials, learning materials

Posted Date: May 6th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9615104/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Single-Thinking-Process-Based Learning of Complete Issue-Related Materials for Autonomous Agents

Hong Su

***Abstract*—Autonomous agents operating in open and long-term environments should be able to learn from their own observations, actions, reasoning processes, external model interactions, and feedback with reduced human guidance. However, many existing agent learning frameworks treat perception, reasoning, memory, action, and feedback as relatively separated processes, which may cause useful learning materials generated around the same issue to be fragmented or ignored. This paper proposes a *Single-Thinking-Process-Based Learning of Complete Issue-Related Materials* model for autonomous agents. The core idea is that all issue-related information is processed through a unified thinking process, including sensory observations, action contexts, LLM prompts, LLM responses, recalled cases, comparison results, judgments, and feedback. Since learning materials are generated through processing, deeper or repeated thinking can produce richer materials for the same issue, while unprocessed data remain raw and do not form structured training materials. The collected materials are organized into named sequences to support recall, comparison, reuse, merging, branching, and learning carrier update. The model also introduces active processing for unmastered issues, allowing the agent to obtain additional observations, reasoning results, or feedback when the current material is insufficient. Controlled verification results show that the proposed model improves recall coverage from about 0.328 to 0.965, improves reuse success rate from 0.919 to 0.991, and increases mastery degree from 0.742 to 0.808 compared with LLM-only processing.**

***Index Terms*—Large Language Models (LLMs), Single-Thinking-Process-Based Learning, Complete Issue-Related Materials, learning materials**

I. Introduction

Autonomous agents are increasingly expected to operate in open, dynamic, and long-term environments [1]. In such environments, an agent should not only execute predefined tasks, but also continuously learn from what it observes, what it does, what it recalls, what it asks, what it receives from external models, and what feedback it obtains after action execution. For example, when a robot hears an unfamiliar sound, fails to adjust a hand movement, or observes a new environmental change, the useful learning material is not only the raw observation or the final judgment. The related task context, action state, recalled past cases, reasoning process, external model interaction, comparison result, and later feedback may all be important for future learning.

H. Su is with the School of Computer Science, Chengdu University of Information Technology, Chengdu, China.
E-mail: suguest@126.com.

However, many existing autonomous agent frameworks treat perception, reasoning, memory, action, and feedback as relatively separated modules. In such frameworks, sensory data may be stored as observations, actions may be stored as execution logs, feedback may be stored as rewards, and large language model (LLM) [2] interactions may be used only for immediate reasoning. Although this modular design is convenient, it may fragment the materials generated around the same issue. As a result, the agent may remember what happened or what decision was made, but fail to preserve the complete context explaining how the issue was processed and why a certain judgment or action was produced.

This limitation becomes more serious when the agent needs to learn continuously with reduced human guidance. In many learning systems, human designers need to specify what data should be stored, which feedback should be used, which events are important, and when the agent should update its knowledge. However, an autonomous agent operating for a long time should be able to discover useful learning materials by itself. If all data generated and processed during issue handling can be collected, compared, organized, and updated through the agent's own thinking process, then the agent can gradually learn from its own observations, actions, reasoning traces, LLM interactions, recalls, comparisons, and feedback without requiring humans to manually select every learning target or annotate every useful case.

Another challenge is that long-term operation may produce a large amount of repeated, slightly different, or genuinely new issue-related data. If every raw event is stored independently, the memory size may grow rapidly and contain much redundant information. If only final labels or results are stored, useful reasoning materials may be lost. If different modules store their own outputs separately, the relationships among observation, recall, comparison, judgment, action, and feedback may be weakened. Therefore, an important problem is how an autonomous agent can collect complete issue-related learning materials, support self-directed learning with less human intervention, and avoid unnecessary storage growth.

To address this problem, this paper proposes a *Single-Thinking-Process-Based Learning of Complete Issue-Related Materials* model for autonomous agents. The main idea is that all issue-related information should first be processed through a unified thinking process. This process handles sensory observations, task contexts, action-related information, LLM prompts and responses, recalled past materials, comparison results, and feedback. Since these materials pass through the

same thinking process, the agent can associate them as parts of the same issue and treat them as complete candidate learning materials.

Different from methods that learn only from raw observations, final decisions, or external rewards, the proposed model learns from the complete context generated during issue processing. For each encountered issue, the agent collects the materials processed by the single thinking process, recalls related past materials, compares the current issue with previous experiences, judges whether it is familiar, slightly different, new, important, or insufficiently mastered, and then organizes the collected materials into a named sequence. These sequences are used to construct and update learning carriers, such as sequence lists, case memories, rule sets, embedding databases, neural models, or hybrid structures.

Meanwhile, sequence-based organization is introduced because autonomous agents usually process information in an ordered manner. An agent may first observe the environment, then interpret the observation, recall related experiences, compare the current situation with past materials, make a judgment, take an action, and finally obtain feedback. Therefore, the learning material should preserve not only the content of each element, but also the processing order through which the agent understood and handled the issue. Moreover, since collecting all processed materials may generate many sequences during long-term operation, each sequence is assigned a name or label to support efficient recall, comparison, merging, branching, and reuse.

The proposed model also includes an active processing mechanism for unmastered issues. If the agent judges that the current issue is not sufficiently understood, it does not simply store the incomplete material and wait passively for future data. Instead, it may actively collect additional observations, monitor the action process more finely, request further LLM reasoning, repeat an action under controlled conditions, or wait for delayed feedback. The newly obtained materials are added to the same issue-related sequence, so that the learning carrier can be improved through continuous enrichment.

The main contributions of this paper are summarized as follows:

- This paper proposes a single-thinking-process-based learning framework in which observation, action context, LLM interaction, recall, comparison, judgment, and feedback are all processed by the same single thinking process, aiming to simulate how humans learn solely through thinking or brain-based information processing.
- This paper introduces the concept of complete issue-related learning materials, where all useful materials generated during the thinking process are collected as candidate learning materials. The model further emphasizes that more thinking and active processing can generate more training materials for the same issue, while unprocessed issues produce little or no structured learning material.
- This paper designs a sequence-based organization mechanism that converts complete issue-related materials into named sequences, supporting efficient recall, comparison, reuse, merging, and branching.

The rest of this paper is organized as follows. Section II reviews related work on autonomous agent learning, memory mechanisms, LLM-based agents, and continual learning. Section III presents the proposed single-thinking-process-based learning model. Section IV reports the verification design and experimental results. Section V concludes the paper and discusses future work.

II. Related Work

This section reviews related work from four aspects: (1) LLM-Based Agents and Embodied Reasoning, (2) Agent Memory and Experience Organization, (3) Lifelong and Self-Improving LLM Agents, and (4) continual Learning, Active Learning, and Reduced Human Guidance. These areas are closely related to the proposed model, but they do not directly address the problem of learning complete issue-related materials through a single thinking process.

A. LLM-Based Agents and Embodied Reasoning

Large language models have increasingly been used as reasoning cores for autonomous agents [3]. ReAct integrates reasoning traces with actions in an interleaved manner, enabling language models to reason about a task while interacting with external environments or tools [4]. This shows that reasoning and acting should not be treated as completely separated processes. Similarly, SayCan grounds language-model outputs in robotic affordances by combining language-model scores with value functions over executable skills [5]. RT-2 further extends this direction by representing robotic actions as tokens and co-training vision-language-action models, allowing web-scale vision-language knowledge to support robotic control [6].

These studies demonstrate that LLMs and vision-language models can support reasoning, action planning, and embodied decision-making. However, their main focus is usually task execution, action feasibility, or generalization from large-scale pretraining. In many cases, the LLM interaction is used to produce an immediate plan, action, or decision, while the materials generated during this process are not explicitly organized as reusable learning materials.

In contrast, this paper focuses on how an autonomous agent can learn from all materials generated during issue processing. The proposed model treats observations, action contexts, LLM prompts, LLM responses, recalled cases, comparison results, judgments, and feedback as complete issue-related learning materials. Therefore, the LLM is not only used as a planner or controller, but also becomes part of the learning material generation process. This allows the agent to learn not only from final actions or results, but also from the complete reasoning and interaction context that leads to them.

B. Agent Memory and Experience Organization

Memory is an important component of autonomous agents. Generative Agents introduce a memory stream, reflection mechanism, and planning process to simulate believable human-like behaviors [7]. In this architecture, agents store observations in natural language, retrieve relevant memories, and

synthesize higher-level reflections. Recent surveys on LLM-agent memory also describe memory as a central mechanism for storing, managing, and retrieving past experiences in agent systems [8]. These studies show that memory is necessary for long-term consistency and adaptive behavior.

However, existing memory mechanisms often store observations, reflections, or interaction records as separate memory items. Although such memories can be retrieved later, the complete relationship among observation, action, recall, LLM interaction, comparison, judgment, and feedback may not always be preserved as a single issue-level learning unit. The proposed model differs by emphasizing the complete collection of issue-related materials and the sequential processing of these materials in the order they are handled by the thinking process. Because all related materials are processed by a single thinking process, they can be organized together before being stored. This enables the agent to recall not only what happened, but also how the issue was understood, what was compared, what judgment was made, and what feedback was later obtained.

C. Lifelong and Self-Improving LLM Agents

Lifelong learning agents aim to continuously acquire new capabilities through long-term interaction with their environments. Voyager is an LLM-powered embodied lifelong learning agent in Minecraft. It explores the environment, acquires executable skills, stores them in a skill library, and reuses them in later tasks [9]. Reflexion proposes verbal reinforcement learning, in which language agents use feedback to generate self-reflections and store them in episodic memory to improve future trials without updating model weights [10]. These works show that LLM-based agents can improve through accumulated experience rather than relying only on offline training.

The proposed method shares the goal of long-term self-improvement, but it focuses on a different problem. Voyager mainly emphasizes autonomous exploration, skill generation, and skill reuse, while Reflexion emphasizes feedback-driven verbal reflection. In contrast, this paper focuses on the source, completeness, and organization of learning materials. It argues that an agent should not learn only from final feedback, generated skills, or reflection texts. Instead, all materials processed during the thinking process, including observations, action contexts, LLM interactions, recalled cases, comparison results, judgments, and feedback, should be treated as candidate learning materials.

Therefore, the proposed model extends self-improvement from feedback- or skill-centered learning to complete issue-related material learning. The learning target is not fully predefined by humans, but is gradually discovered from the agent's own observation, reasoning, comparison, action, and feedback processes. This enables the agent to preserve richer learning contexts and provides a basis for self-directed learning with reduced human guidance.

D. Continual Learning, Active Learning, and Reduced Human Guidance

Continual learning studies how models learn sequentially from changing data while reducing catastrophic forgetting [11, 12]. Existing methods commonly use parameter regularization, replay memory, dynamic architectures, or generative replay to preserve previous knowledge while learning new tasks. These methods are important for maintaining performance across multiple tasks. However, many of them assume that learning samples, labels, or task boundaries are already provided. For autonomous agents operating in open environments, a more basic problem is how the agent discovers what should become a learning sample during continuous interaction.

Active learning and self-directed learning address related problems by allowing a model or agent to request useful information when uncertainty is high. In agent systems, this idea can be extended to active processing. When an issue is not sufficiently mastered, the agent may collect more observations, request further reasoning, repeat actions, monitor execution more finely, or wait for additional feedback. The proposed model incorporates this idea through active processing for unmastered issues.

Different from ordinary active learning, the newly obtained information in the proposed model is not stored as an isolated sample. Instead, it is added to the same issue-related sequence. Therefore, the additional observation, reasoning result, or feedback remains connected with the original observation, recalled cases, comparison process, judgment, and action context. This design supports reduced human guidance because the agent can decide when further information is needed and can enrich its own learning materials without requiring humans to manually define every learning target or annotate every useful case.

III. Single-Thinking-Process-Based Learning Model

The proposed model aims to enable an autonomous agent to learn not only from final decisions or external feedback, but also from the materials generated and processed during its own thinking process. In open environments, an issue may involve sensory observations, action contexts, prompts to large language models, model responses, intermediate interpretations, comparison results, uncertainty states, action attempts, and execution feedback. If these materials are processed by separated modules and stored independently, useful relationships among them may be weakened or lost.

To address this problem, this paper proposes a single-thinking-process-based learning model. The central idea is that all issue-related information is first processed through a unified thinking process. Once an issue is processed, the materials generated during this process are treated as candidate training materials. These materials are then collected, organized into sequence-based training data, and used to update learning carriers. In this way, the agent can learn from the complete processing context of an issue rather than from isolated data points.

A key assumption of the proposed model is that training material is generated through thinking. If an issue is not

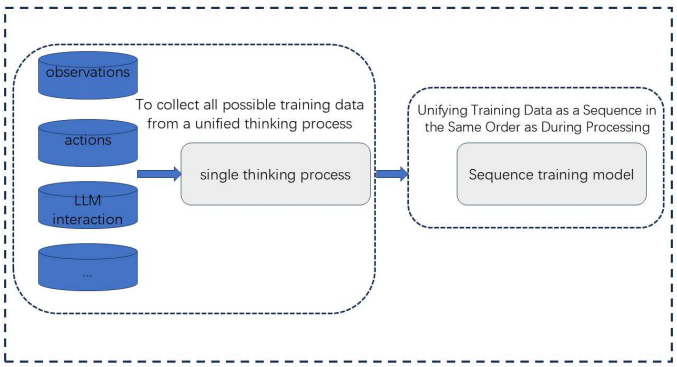


Fig. 1. Overall structure of the proposed single-thinking-process-based learning model. Issue-related information is processed by a single unified thinking process, and organized as sequence-based training data.

processed by the single thinking process, the agent can only obtain raw data or scattered signals, and no structured issue-related training material is formed. In contrast, when an issue is processed more deeply or repeatedly, more learning materials are generated. For example, when a strange object is processed by the thinking process multiple times, the agent may gradually collect its visual features, task context, LLM-assisted explanations, action interactions, uncertainty states, and feedback. As these materials accumulate, the object becomes less strange in later encounters.

The proposed model contains five main components. First, the single thinking process receives and interprets different types of issue-related information, including observations, action guidance, LLM interactions, intermediate reasoning results, and feedback. Second, the complete issue-related material collection mechanism records useful materials generated during the thinking process. Third, the sequence-based organization mechanism converts the processed materials into sequence-based training data. Fourth, the learning carrier construction and update mechanism stores, updates, merges, branches, or trains learning structures based on these sequences. Fifth, the active processing mechanism requests further observation, reasoning, action monitoring, or feedback when an issue is not sufficiently mastered.

Based on these components, the agent forms a closed learning loop. For each encountered issue, it processes related information through the single thinking process, generates learning materials from the processing steps, converts the materials into sequence-based training data, updates its learning carriers when necessary, and actively obtains additional materials when the issue is not sufficiently mastered. The overall structure of the proposed model is illustrated in Fig. 1.

A. Problem Definition

Consider an autonomous agent operating in an open environment. At each time step t, the agent may receive different types of issue-related information, including sensory observations, task states, action contexts, external feedback, and interactions with a large language model. Let

$$\mathcal{I}_t = \{o_t, c_t, a_t, q_t, y_t, f_t\} \tag{1}$$

denote the set of issue-related information available at time t, where o_t is the sensory observation, c_t is the current task or environmental context, a_t is the action-related information, q_t is the query or prompt sent to the LLM, y_t is the response returned by the LLM, and f_t is the feedback obtained from execution or later observation.

In this paper, the *learning carrier* denotes the long-term structure that stores, maintains, trains, or uses the learning results generated from processed materials. It may be a sequence list, a case memory, a rule set, an embedding index, an LSTM module, a Transformer-based module, a policy model, or a hybrid structure. Let $\mathcal{M}_t$ denote the learning carrier at time t.

The agent processes $\mathcal{I}_t$ through a single thinking process and produces a unified thinking result:

$$z_t = T(\mathcal{I}_t, \mathcal{M}_t), \tag{2}$$

where $T(\cdot)$ denotes the single thinking process and z_t is the generated thinking result. The thinking result may include interpretations, intermediate reasoning, LLM-assisted explanations, action suggestions, uncertainty states, or feedback connections.

The objective of the proposed model is not only to produce an immediate action, but also to transform the materials generated during the thinking process into useful training data. The complete issue-related learning material is defined as

$$L_t = \Psi(\mathcal{I}_t, z_t), \tag{3}$$

where $\Psi(\cdot)$ denotes the material collection process and L_t represents the complete issue-related material generated at time t.

Here, z_t denotes the immediate result produced by the single thinking process, while L_t denotes the complete learning material constructed from both the original issue-related information and the thinking result. Therefore, z_t is a component or source of L_t, but L_t contains a broader training context.

The collected material is further converted into sequence-based training data:

$$S_t = \Phi(L_t), \tag{4}$$

where $\Phi(\cdot)$ denotes the sequence construction function. The sequence S_t preserves the processing order of the issue-related materials, such as observation, interpretation, LLM interaction, action, and feedback.

Each sequence is assigned a name or label:

$$n_t = N(S_t), \tag{5}$$

where $N(\cdot)$ denotes the naming function and n_t is the name of the sequence. The name helps organize large numbers of sequence-based training data generated during long-term operation.

Given the current learning carrier $\mathcal{M}_t$, the carrier is updated as

$$\mathcal{M}_{t+1} = U(\mathcal{M}_t, S_t, n_t), \tag{6}$$

where $U(\cdot)$ is the carrier update function. The update may store a new sequence, refine an existing module, merge related sequences, create a new branch, or mark a sequence for further processing.

Thus, the problem studied in this paper can be summarized as follows: how can an autonomous agent use a single thinking process to generate complete issue-related learning materials, convert them into sequence-based training data, and update its learning carriers so that future reasoning and actions can benefit from the full context of previous thinking processes?

B. Overview of the Framework

The proposed framework is built on the assumption that an autonomous agent should not learn only from isolated observations, final actions, or delayed feedback. Instead, it should learn from the complete issue-related materials produced during its own thinking process. When an issue appears, the agent may observe the environment, interpret the observation, generate a prompt to an LLM, receive a response, execute an action, and obtain feedback. All these materials are related to the same issue and should be jointly considered as candidate training materials.

Thus, the framework contains five connected parts: the single thinking process, complete material collection, sequence-based organization, learning carrier construction and update, and active processing for insufficiently mastered issues. The single thinking process acts as the central processor. It receives multimodal observations, action contexts, LLM interactions, intermediate reasoning results, and feedback, and then produces a unified processing result for the current issue.

The complete material collection component records the materials generated and used during the thinking process. These materials include not only external data, such as observations, actions, and feedback, but also internal processing-related materials, such as prompts, LLM responses, interpretations, comparison results, uncertainty states, and explanations. Therefore, the learning material is not a single data item, but a structured set describing how the agent understood and handled the issue.

The sequence-based organization component converts the complete issue-related material into sequence-based training data. This conversion is important because an autonomous agent usually processes information in order. For example, it may first observe an issue, then interpret it, then ask an LLM, then execute an action, and finally obtain feedback. Therefore, the training data should preserve this processing order rather than only store unordered fragments.

The learning carrier construction and update component stores or trains the organized sequences in a form that can support future use. The learning carrier may be a simple sequence list, a case memory, a rule set, an embedding database, an LSTM module, a Transformer-based module, a policy model, or a hybrid structure. If the agent considers that the current issue is not sufficiently mastered, the active processing component can request more observations, finer action monitoring, additional LLM reasoning, repeated execution, or delayed feedback. In this way, the framework forms a closed loop from thinking, material generation, material collection, sequence construction, learning update, and further processing.

C. Single Thinking Process

The single thinking process is the central mechanism that transforms issue-related information into learnable materials. Instead of allowing observations, actions, LLM interactions, intermediate states, and feedback to be handled as separated and weakly connected outputs, the proposed model processes them through one unified thinking channel. This allows the agent to preserve the issue-level relationships among materials generated around the same event.

As defined in Section III-A, z_t denotes the unified result produced by the single thinking process. The output z_t is not limited to a final decision. It may include the interpreted meaning of the observation, the relation between the current issue and the task context, the reason for calling or not calling the LLM, an uncertainty state, the estimated mastery degree, and the suggested action. Therefore, z_t provides a unified context for subsequent material collection and sequence construction.

An important property of the single thinking process is that learning material is generated through processing. If an issue is not processed by the thinking process, the agent can only obtain raw data or scattered signals, and no structured issue-related training material is formed. Let p_t denote the processing depth or number of thinking steps applied to the current issue at time t. The generated structured learning material can be written as

$$L_t(p_t) = \bigcup_{k=1}^{p_t} \Delta L_t^{(k)}, \tag{7}$$

where $\Delta L_t^{(k)}$ denotes the newly generated material at the k-th thinking step. If the issue is not processed, then $p_t = 0$, and thus

$$L_t(0) = \emptyset. \tag{8}$$

In contrast, when an issue is processed more deeply or repeatedly, p_t increases and more related materials are generated, such as interpretations, LLM-assisted explanations, comparison results, action attempts, uncertainty states, and feedback links. Therefore, the amount and richness of learning material depend on how the issue is processed:

$$|L_t(p_t + 1)| \geq |L_t(p_t)|. \tag{9}$$

This means that a strange issue may gradually become less strange as more thinking steps generate and accumulate richer issue-related materials.

For example, when an agent first observes a strange object, it may only record its appearance and an uncertain state. If the object is further processed by the thinking process, the agent may compare its features with the current task context, ask an LLM for possible explanations, interact with the object, observe the action result, and receive feedback. These additional materials are then associated with the same issue. As a result, when the object appears again later, it is no

longer completely strange because the learning carrier already contains a richer issue-related sequence about it.

The single thinking process also provides the source of sequence-based training data. Materials generated during processing are not directly stored as unrelated fragments. Instead, they are collected into complete issue-related materials and further converted into sequences. Therefore, the training data used by the agent originate from the thinking process itself. The more the issue is processed, the richer the corresponding sequence-based training data become.

Different from a modular pipeline in which each component only passes a partial result to the next component, the single thinking process keeps the issue-level relationship among materials. For example, when a robot hears an unfamiliar sound while executing a movement, the sound observation, movement context, LLM-assisted interpretation, action adjustment, and later feedback are processed as related parts of the same issue. This makes it possible to learn not only that a certain sound appeared, but also when it appeared, what action was being performed, how the agent processed it, and whether the processing result was later supported by feedback.

The single thinking process also provides the basis for data reduction. Since all related materials are processed together, the agent can determine whether a newly generated material is useful, redundant, incomplete, uncertain, or important for future learning. Thus, the framework avoids storing all raw data blindly and instead stores processed materials that are useful for future learning and action improvement.

In this paper, the single thinking process is not restricted to a specific implementation. It can be implemented by an LLM, a rule-based reasoning module, a neural-symbolic model, a deep learning model, or a hybrid mechanism. The essential requirement is that the process should provide a unified issue-level processing channel through which complete issue-related materials can be generated, collected, organized, and updated for future learning.

D. Complete Issue-Related Material Collection

After the single thinking process produces the processed result z_t, the agent collects the materials that are related to the current issue. The purpose of this step is to avoid learning only from isolated data, such as a single observation, a final action, or a reward signal. Instead, the agent records the complete context that supports future learning.

The complete issue-related material at time step t is denoted as

$$L_t = \{o_t, c_t, a_t, q_t, y_t, f_t, z_t\}, \tag{10}$$

where o_t is the sensory observation, c_t is the task or environmental context, a_t is the action-related information, q_t is the LLM query, y_t is the LLM response, f_t is the feedback, and z_t is the unified thinking result.

In practice, not every element must appear at every time step. For example, when the agent can handle an issue by itself, the LLM query q_t and response y_t may be empty. When no feedback has been received yet, f_t may be temporarily unavailable and can be added later. Therefore, L_t is a flexible material set that can be completed or updated as more related information becomes available.

The collected material should preserve both external and internal information. External information includes observations, task states, action results, and environmental feedback. Internal information includes LLM prompts, LLM responses, comparison results, uncertainty states, intermediate interpretations, and explanations generated by the thinking process. By storing these materials together, the agent can later use not only what happened, but also how the issue was processed and under what conditions it appeared.

The collection process can be expressed as

$$L_t = C(\mathcal{I}_t, z_t), \tag{11}$$

where $C(\cdot)$ denotes the material collection function. This function selects, links, and records the materials that are useful for describing the current issue. If some materials are redundant or weakly related, they can be compressed or discarded. If some materials are important but incomplete, the agent may mark them for later completion through further observation or feedback.

The proposed model further considers that issue-related materials can grow as the thinking process continues. Suppose the same issue is processed through multiple thinking steps. The collected material after the k-th processing step can be written as

$$L_t^{(k)} = L_t^{(k-1)} \cup \Delta L_t^{(k)}, \tag{12}$$

where $\Delta L_t^{(k)}$ denotes the newly generated material at the k-th thinking step. Such newly generated material may include an additional observation, an LLM-assisted explanation, a comparison result, a correction attempt, an uncertainty state, or feedback obtained after action execution.

If no thinking process is activated for an issue, then no structured additional learning material is produced:

$$\Delta L_t^{(k)} = \emptyset. \tag{13}$$

This means that raw data alone does not necessarily become useful training material. It becomes learnable only after the agent processes, interprets, and connects it with related context. Therefore, a strange object, sound, or event can gradually become familiar only when it is processed and enriched through repeated thinking.

This collection mechanism is central to the proposed model. Since all issue-related materials pass through the same thinking process, the agent can associate them before learning. As a result, learning is based on a complete issue-level material unit rather than scattered fragments. This provides the basis for sequence-based organization and learning carrier update.

E. Sequence-Based Organization of Training Materials

After complete issue-related materials are collected, they are organized into a sequence-based form. This design is motivated by the fact that an autonomous agent usually processes information in an ordered manner. For example, the agent may first observe the environment, then interpret the observation,

generate or receive LLM-assisted information, take an action, and finally obtain feedback. Therefore, the training material should not be represented only as an unordered set of data. Instead, it should preserve the processing order through which the agent understands and handles an issue.

The purpose of sequence-based organization is to convert scattered issue-related materials into structured training data that can be stored, updated, trained, and used in future tasks. In the proposed model, a sequence does not only represent a temporal order of external observations. It also represents the internal processing order of thinking, interpretation, action, feedback, and correction related to an issue. In this way, the agent can learn not only what materials appeared, but also how these materials were processed step by step.

Let S_t denote the sequence-based training data generated from the current issue-related material. The sequence construction process can be represented as

$$S_t = \Phi(L_t), \tag{14}$$

where $\Phi(\cdot)$ is the sequence construction function and L_t is the current complete issue-related material. The generated sequence S_t preserves the main information flow of the issue, such as observation, context, LLM interaction, action, and feedback.

A general sequence can be written as

$$S_t = \langle e_t^1, e_t^2, \ldots, e_t^n \rangle, \tag{15}$$

where each e_t^i is an element related to the current issue. For example, an element may be an observation, an LLM response, a comparison result, an action, or feedback. Therefore, the sequence records not only what happened, but also how the agent processed and understood the issue.

The sequence is also a carrier for the gradual transformation from unfamiliarity to familiarity. When an issue is first encountered, its sequence may be short and uncertain. As the thinking process continues, more elements are appended to the sequence, such as additional observations, LLM-assisted explanations, comparison results, actions, and feedback. Therefore, the sequence becomes richer over time. When the same or a similar issue appears later, the agent can use this enriched sequence and treat the issue as less strange than before.

Each sequence is assigned a name or label to support later organization and use. This is necessary because the proposed model collects all processed materials as candidate training materials, which may produce a large number of sequences during long-term operation. Without proper names or labels, it would be difficult for the agent to efficiently organize related sequences, compare a new issue with existing sequence-based data, or decide whether the current material should be merged, branched, or stored as a new sequence.

The name can be determined by the main feature, the associated task, the main difference from existing materials, or the issue being handled. This process can be represented as

$$n_t = N(S_t), \tag{16}$$

where $N(\cdot)$ is the naming function and n_t is the sequence name. For example, a sound-related sequence may be named according to a vehicle type, a movement-related sequence may be named according to an action failure, and a reasoning-related sequence may be named according to the issue being processed.

The sequence organization is flexible. If the current material is redundant, S_t may be ignored or only used to update frequency information. If it is useful but related to an existing sequence, it may be merged into the existing sequence. If it contains important new information, a new sequence can be created. If the material shares an early part with an existing sequence but later develops differently, the existing sequence can be branched into different paths. This supports both stability and extensibility of the learning carrier.

For example, when the agent observes a strange object, the first sequence may contain only the visual observation and uncertainty state. After more thinking, interaction, and feedback, the sequence can be enriched with object features, possible category explanations, action results, and corrected processing results. Thus, later encounters with the object become easier to understand.

1) Sequence-Trained Learning Modules: The generated sequences can further serve as training data for different learning modules. According to the type of issue and the structure of the sequence, a sequence may be used to train a recurrent model, such as an LSTM, a Transformer-based model, a rule-based sequence processor, a policy model, or another task-specific learning module. Thus, the proposed model does not restrict the learning carrier to a single neural architecture. Instead, many sequence-trained modules can be formed from different types of processed materials.

This process can be written as

$$G_i = \Omega_i(\mathcal{S}_i), \tag{17}$$

where $\mathcal{S}_i = \{S_1, S_2, \ldots, S_m\}$ denotes a set of related sequences, $\Omega_i(\cdot)$ denotes the training or construction process, and G_i denotes the learned module generated from these sequences. For example, G_i may be an LSTM module for temporal action correction, a Transformer module for multimodal reasoning sequences, or a symbolic sequence module for rule-like decision processes.

The purpose of such modules is to transform sequence-based training data into reusable processing ability. A module trained from hand-movement sequences may help correct future movement directions. A module trained from sound-related sequences may help analyze acoustic patterns. A module trained from object-interaction sequences may help identify and handle unfamiliar objects. Therefore, the sequence is not only a storage format, but also a training unit from which specialized learning modules can be constructed.

2) Flexible Composition of Sequence-Trained Modules: The sequence-trained learning modules do not need to work independently. Since different issues may depend on different prerequisite processing steps, the learned modules should be flexibly composed according to the processing relationship among sequences. A newly learned module can be inserted before an existing module, appended after an existing module,

merged with a related module, or treated as a branch of an existing module.

Let G_i and G_j denote two modules trained from different sequence sets. Their relationship can be represented as

$$G_i \rightarrow G_j, \tag{18}$$

which means that the output of G_i provides a prerequisite condition or useful context for G_j. A branch relationship can be represented as

$$G_i \rightarrow \{G_j^{(1)}, G_j^{(2)}, \ldots, G_j^{(m)}\}, \tag{19}$$

where different subsequent modules are selected according to the output or state produced by G_i. In this way, sequence-trained modules can form a flexible processing graph rather than a fixed linear pipeline.

This composition ability is important because some processing tasks cannot be performed reliably without previous processing steps. For example, Bora and Tesla vehicle sounds may be different. If the agent directly judges distance or motion state only from the speed or rhythm of the sound, the result may be unreliable because different vehicle types can produce different acoustic patterns under similar distances or speeds. Therefore, a vehicle-type processing module should be placed before a sound-based distance or motion module:

$$G^{\text{vehicle}} \rightarrow G^{\text{sound}} \rightarrow G^{\text{distance}}. \tag{20}$$

Here, G^{vehicle} first estimates the vehicle type, G^{sound} analyzes the sound characteristics, and G^{distance} judges distance or motion-related information. If the vehicle type is unknown or uncertain, the system may create different branches:

$$G^{\text{sound}} \rightarrow \{G^{\text{tesla}}, G^{\text{bora}}, G^{\text{unknown}}\}. \tag{21}$$

This means that the same sound-processing sequence may lead to different subsequent modules according to the inferred vehicle type. As a result, the agent does not rely on a single fixed sound processing path, but can adjust the processing route according to the issue context.

The flexible composition of sequence-trained modules also supports long-term growth. When a new issue is processed and transformed into a new sequence, the corresponding module can be inserted into an existing processing path if it provides a missing prerequisite step. It can also be appended after an existing module if it refines a later processing stage. If it represents a different development from an existing sequence, it can be stored as a branch. Therefore, the learning carrier grows not only by adding more data, but also by reorganizing the relationships among learned modules.

Through sequence-based organization, the proposed model transforms complete issue-related materials into reusable sequence-based training data. These sequences provide the direct input for learning carrier construction and update. They can also form different trainable modules, such as LSTM-based, Transformer-based, symbolic, or hybrid modules. By allowing these modules to be inserted, appended, merged, or branched, the proposed model supports flexible long-term learning from the materials generated by the single thinking process.

F. Learning Carrier Construction and Update

After the sequence-based training data S_t and its name n_t are generated, the agent needs to place them into a long-term structure that can support future learning and task processing. In this paper, this structure is called the learning carrier. The learning carrier does not describe how a sequence is formed; instead, it describes where the sequence-based training data are stored, maintained, trained, and used.

Let $\mathcal{M}_t$ denote the current learning carrier at time step t. It may contain sequence lists, trained modules, rule sets, embedding indexes, neural models, policy models, or hybrid structures. The carrier update process can be represented as

$$\mathcal{M}_{t+1} = U(\mathcal{M}_t, S_t, n_t), \tag{22}$$

where $U(\cdot)$ is the carrier update function, S_t is the generated sequence-based training data, and n_t is the sequence name.

The update operation depends on the role of S_t in the current carrier. If S_t only repeats existing knowledge, the carrier may remain unchanged or only update frequency and confidence information. If S_t provides useful new elements for an existing carrier, it can be used to refine that carrier. If S_t represents a new issue or a new processing path, a new carrier component can be created. If S_t is incomplete or insufficiently mastered, it can be marked for active processing.

This process can be summarized as

$$\mathcal{M}_{t+1} = \begin{cases} \mathcal{M}_t, & S_t = \text{redundant}, \\ \text{Refine}(\mathcal{M}_t, S_t), & S_t = \text{useful}, \\ \text{Create}(\mathcal{M}_t, S_t, n_t), & S_t = \text{new}, \\ \text{Mark}(\mathcal{M}_t, S_t), & S_t = \text{unmastered}. \end{cases} \tag{23}$$

Compared with the previous sequence organization step, this step focuses on the maintenance of the whole learning system. For example, a sequence may first be organized as training data, and then the carrier update function decides whether it should update an existing LSTM module, create a new Transformer-based module, modify a symbolic rule, update an embedding index, or be stored as an unfinished sequence for later enrichment.

Therefore, the learning carrier connects the current thinking process with future thinking processes. It preserves the usable results of sequence-based training data and maintains the agent's long-term learning state. Through carrier construction and update, the proposed model converts materials generated by the single thinking process into reusable knowledge for future autonomous operation.

G. Active Processing for Unmastered Issues

Although the proposed model collects complete issue-related materials and updates the learning carrier, a single processing result may still be insufficient for reliable learning. In open environments, the agent may encounter issues that are

partially understood, weakly supported by feedback, or different from previous experiences but not yet clear enough to form stable knowledge. Therefore, the agent should not passively wait for future data. Instead, it should actively request further processing when the current issue is not sufficiently mastered.

Let m_t denote the mastery degree of the current sequence S_t. It can be estimated according to the consistency of feedback, the stability of action results, the completeness of sequence elements, or the uncertainty of LLM-assisted processing. The mastery degree can be represented as

$$m_t = H(S_t, f_t), \tag{24}$$

where $H(\cdot)$ is the mastery estimation function, S_t is the current sequence, and f_t is the feedback.

If the mastery degree is lower than a predefined threshold τ, the agent activates further processing:

$$A_t^{\text{active}} = \begin{cases} 1, & m_t < \tau, \\ 0, & m_t \geq \tau. \end{cases} \tag{25}$$

When $A_t^{\text{active}} = 1$, the agent may perform one or more active processing operations. These operations may include collecting additional observations, monitoring the execution process more finely, asking the LLM for further analysis, repeating an action under controlled conditions, or waiting for delayed feedback. The purpose is to enrich the current learning sequence and reduce uncertainty before the material is treated as stable learned knowledge.

Active processing also increases the amount of training material available for the issue. The more the agent observes, compares, reasons, acts, and receives feedback about an unmastered issue, the richer the corresponding issue-related sequence becomes. Therefore, an unfamiliar issue can gradually become familiar through repeated thinking and material accumulation. For example, a strange object may initially be stored with only a visual observation and an uncertain state. After further processing, the sequence may include object features, LLM-assisted explanations, action interaction results, and feedback. These added materials make later recognition and handling more reliable.

The active processing result can be written as

$$L_t^+ = L_t \cup \Delta L_t, \tag{26}$$

where ΔL_t denotes the additional materials obtained through active processing. The corresponding sequence is updated as

$$S_t^+ = \Phi(L_t^+). \tag{27}$$

Then the learning carrier is updated using the enriched sequence:

$$\mathcal{M}_{t+1} = U(\mathcal{M}_t, S_t^+, n_t). \tag{28}$$

This mechanism makes the proposed model different from passive memory accumulation. The agent does not only store what has already happened, but also actively generates new related materials when the current issue is not well understood. As a result, the learning carrier can be improved through additional observation, reasoning, feedback, and correction, which supports more reliable long-term autonomous learning.

Algorithm 1 Single-Thinking-Process-Based Learning Procedure

Require: Current learning carrier $\mathcal{M}_t$, issue-related information $\mathcal{I}_t$
Ensure: Updated learning carrier $\mathcal{M}_{t+1}$
Process $\mathcal{I}_t$ through the single thinking process to obtain z_t.
Generate and collect complete issue-related material L_t based on $\mathcal{I}_t$ and z_t.
Construct sequence-based training data S_t from L_t.
Assign a name or label n_t to S_t.
Estimate the mastery degree m_t of S_t.
if $m_t < \tau$ **then**
    Actively collect additional materials ΔL_t.
    Enrich the current material: $L_t^+ \leftarrow L_t \cup \Delta L_t$.
    Reconstruct the sequence: $S_t^+ \leftarrow \Phi(L_t^+)$.
    Update the carrier: $\mathcal{M}_{t+1} \leftarrow U(\mathcal{M}_t, S_t^+, n_t)$.
else
    Update the carrier: $\mathcal{M}_{t+1} \leftarrow U(\mathcal{M}_t, S_t, n_t)$.
end if
return $\mathcal{M}_{t+1}$

H. Algorithmic Procedure

The overall procedure of the proposed model is summarized in Algorithm 1. The algorithm shows how an autonomous agent processes each encountered issue, generates complete issue-related learning materials, converts them into sequence-based training data, and updates the learning carrier. When the current issue is not sufficiently mastered, the agent actively obtains additional materials before updating the carrier.

Algorithm 1 emphasizes that learning is embedded in the thinking process rather than performed only after task completion. The agent preserves the complete issue-related context, including observation, LLM interaction, action, and feedback. Therefore, the updated learning carrier can support future processing, sequence enrichment, and long-term autonomous learning.

IV. Verification

This section verifies the effectiveness of the proposed single-thinking-process-based learning model. The core purpose is to examine whether an autonomous agent can benefit from collecting complete issue-related materials through a unified thinking process. Different from task-oriented evaluations that only measure final task success, the verification in this paper focuses on whether the proposed model can preserve richer learning materials, improve recall and reuse, actively enrich unmastered issues, and reduce unnecessary storage.

The verification is designed around four questions:

- Can the proposed model collect more complete issue-related learning materials than fragmented or result-only learning methods?

- Can sequence-based organization and recall-based comparison improve recall, difference judgment, and reuse of past experiences?
- Can active processing improve learning when an issue is not sufficiently mastered?
- Can the proposed model reduce unnecessary storage while preserving useful learning materials?

To answer these questions, four controlled simulation verifications are conducted. Each verification corresponds to one important component of the proposed framework. The first verification evaluates whether the single thinking process can collect complete issue-related materials. The second verification evaluates whether named sequence organization improves recall coverage, judgment accuracy, and reuse success. The third verification evaluates whether active processing can improve the mastery degree of insufficiently understood issues. The fourth verification evaluates whether the proposed method can reduce storage size while retaining useful information for future tasks.

The compared methods include observation-only storage, result-only storage, fragmented-module storage, unordered memory, unnamed sequence memory, passive processing, random additional observation, LLM-only processing, store-all memory, fixed-threshold compression, and final-label memory. These baselines represent different simplified learning strategies that either store incomplete materials, ignore sequence structure, lack active enrichment, or reduce data without preserving complete issue-related context.

All verification programs are implemented as independent Python scripts. Each script runs multiple test rounds and writes the results into data files. To ensure robustness, each completed round is marked with a completion flag, so that the program can resume from the last incomplete round after interruption. The reported results are obtained from the generated summary files.

A. Verification Design

The verification adopts a controlled simulation setting. In each round, the agent receives a stream of issue-related events. Each event may contain sensory observation, task context, action information, LLM prompt, LLM response, recalled material, comparison result, judgment, and feedback. These elements simulate the materials generated during the thinking process of an autonomous agent.

An event can be represented as

$$E_t = \{o_t, c_t, a_t, q_t, y_t, r_t, d_t, j_t, f_t\}, \tag{29}$$

where o_t denotes the observation, c_t denotes the task or environmental context, a_t denotes the action-related information, q_t denotes the LLM prompt, y_t denotes the LLM response, r_t denotes the recalled material, d_t denotes the comparison result, j_t denotes the judgment, and f_t denotes the feedback.

The proposed method treats these elements as complete issue-related materials. In contrast, the baselines only store part of the event, such as observation only, final result only, unordered records, or compressed labels. Therefore, the verification can compare whether preserving the complete thinking context leads to better learning and reuse.

The verification scenarios include several types of issues, such as vehicle sound differences, hand movement direction deviation, object appearance changes, obstacle avoidance cases, and sensor abnormality patterns. These scenarios are not intended to reproduce a specific physical robot environment. Instead, they provide controlled issue streams for testing whether the proposed model can organize, recall, update, and reuse complete issue-related materials.

The main evaluation metrics are summarized as follows:

- **Material completeness**: the proportion of required issue-related elements successfully stored.
- **Context preservation**: whether the stored material preserves the relation among observation, action, recall, judgment, and feedback.
- **Recall precision**: the proportion of recalled sequences that are truly relevant.
- **Recall coverage**: the proportion of relevant sequences that can be recalled.
- **Reuse success rate**: the proportion of later issues that benefit from recalled materials.
- **Mastery degree**: the estimated understanding level of an issue after processing.
- **Action success rate**: the proportion of issues successfully handled after processing.
- **Storage size**: the number of stored records or sequences.
- **Compression ratio**: the reduction of stored materials compared with the store-all baseline.
- **Useful information retention**: the proportion of important materials preserved after reduction.
- **Future task performance**: the performance of the learning carrier on later related issues.

These metrics jointly evaluate whether the proposed method can learn from complete issue-related materials, reuse past experiences, improve unmastered issues, and maintain storage efficiency.

B. Verification I: Complete Issue-Related Material Collection

The first verification evaluates whether the proposed model can collect more complete issue-related learning materials than methods that store only isolated observations, final results, or separated fragments. This verification directly corresponds to the main motivation of this paper: if all issue-related information is processed by a single thinking process, the agent can preserve the complete materials generated around an issue and use them for later learning.

In this verification, each simulated event contains several types of materials generated during issue processing, including observation, task context, action information, LLM prompt, LLM response, recalled material, comparison result, judgment, and feedback. These elements represent the materials that may appear when an autonomous agent observes an issue, reasons about it, recalls past experience, interacts with an LLM, makes a judgment, and receives feedback.

Four methods are compared:

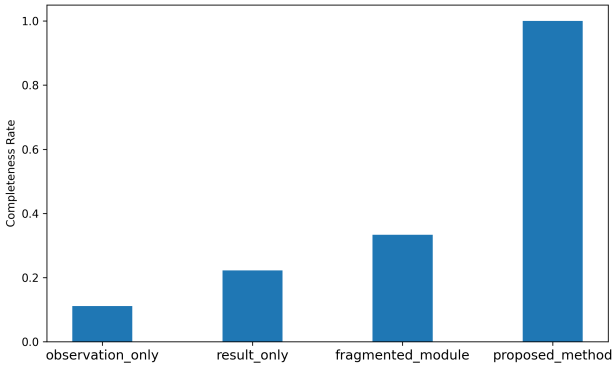


Fig. 2. Material completeness comparison among different storage methods.

- **Observation-only method**: stores only the sensory observation.
- **Result-only method**: stores only the final judgment and feedback.
- **Fragmented-module method**: stores several separated fragments, such as observation, action, and judgment, but does not preserve the complete issue-level context.
- **Proposed method**: stores the complete issue-related materials processed by the single thinking process.

Two metrics are used. *Material completeness* measures the proportion of required issue-related elements that are successfully stored. *Context preservation* measures whether the stored material preserves the relationship among observation, action or context, recall or comparison, judgment, and feedback. These two metrics evaluate not only how much information is stored, but also whether the stored information remains connected as one issue-level learning unit.

Fig. 2 shows the comparison of material completeness among different methods. The proposed method achieves the highest score, indicating that it can preserve all required issue-related elements. In contrast, the observation-only method stores only the sensory input, the result-only method stores only the final result, and the fragmented-module method stores only partial information. Therefore, these baselines lose many materials generated during the thinking process.

The detailed numerical results are shown in Table I. The proposed method achieves 1.000 on both material completeness and context preservation. The observation-only method obtains only 0.111 material completeness and 0.200 context preservation because it records only one part of the issue. The result-only method improves slightly by storing judgment and feedback, but its material completeness remains 0.222. The fragmented-module method records more elements than the first two baselines, but its completeness and context preservation are still limited because the complete issue-related context is not preserved.

These results show that the proposed model has a clear structural advantage in collecting complete issue-related materials. Since the single thinking process handles observation, action context, LLM interaction, recall, comparison, judgment, and feedback together, the generated learning material contains richer contextual information than the baselines. Therefore, the agent can later recall not only what happened or what result was obtained, but also how the issue was processed and why the judgment was made.

TABLE I
RESULTS OF COMPLETE ISSUE-RELATED MATERIAL COLLECTION.

Method	Completeness	Context
Observation-only	0.111	0.200
Result-only	0.222	0.400
Fragmented-module	0.333	0.600
Proposed method	**1.000**	**1.000**

This verification confirms the necessity of learning from complete issue-related materials. If only observations or final results are stored, many useful materials generated during the thinking process are lost. If materials are stored as separated fragments, their issue-level relationships may be weakened. In contrast, the proposed method preserves these relationships and provides a more complete basis for future recall, comparison, and learning.

C. Verification II: Sequence-Based Recall, Difference Judgment, and Reuse

The second verification evaluates whether sequence-based organization and recall-based comparison can improve the recall, difference judgment, and reuse of past experiences. This verification corresponds to the proposed sequence-based organization mechanism. Since the proposed model collects all processed materials as candidate learning materials, many learning sequences may be generated during long-term operation. Therefore, each sequence is assigned a name or label to support efficient recall and comparison.

In this verification, the agent first receives a set of issue-related events and stores them as learning materials. Later, the agent encounters new query events and needs to recall related past materials. The simulated events include several issue types, such as vehicle sounds, hand movement direction deviations, object appearance changes, obstacle avoidance cases, and sensor abnormality patterns. For each query event, the agent should recall useful related sequences and judge whether the current issue is familiar, slightly different, or new.

Three methods are compared:

- **Unordered memory**: stores materials as unordered records and recalls them according to simple similarity.
- **Unnamed sequence memory**: stores complete sequences but does not assign names or labels to them.
- **Proposed named sequence memory**: stores complete issue-related materials as named sequences and recalls them according to sequence name, issue type, and content similarity.

Five metrics are used. *Memory size* measures the number of stored records or sequences. *Recall precision* measures the proportion of recalled sequences that are truly relevant. *Recall coverage* measures the proportion of all relevant sequences that are successfully recalled. *Reuse success rate* measures whether recalled sequences help later issue processing. *Judgment accuracy* measures the accuracy of judging whether the current issue is familiar, slightly different, or new.

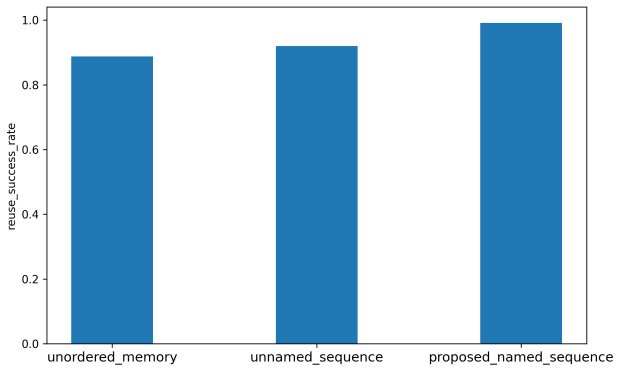


Fig. 3. Reuse success rate of recalled learning materials.

TABLE II
RESULTS OF SEQUENCE-BASED RECALL, DIFFERENCE JUDGMENT, AND REUSE.

Method	Size	Prec.	Cov.	Reuse	Acc.
Unordered memory	80.00	**1.000**	0.328	0.888	0.954
Unnamed sequence	80.00	0.995	0.326	0.919	0.965
Proposed method	**21.73**	0.809	**0.965**	**0.991**	**0.972**

Fig. 3 shows the reuse success rate of different memory methods. The proposed named sequence memory achieves the highest reuse success rate, which indicates that the recalled sequences are more useful for later issue processing. This result supports the motivation of assigning names to sequences: when many sequences are collected from the thinking process, names or labels help the agent retrieve and reuse related materials more effectively.

The detailed numerical results are shown in Table II. The proposed method stores much fewer memory entries than unordered memory and unnamed sequence memory. Its average memory size is only 21.73, while the two baselines both store 80.00 entries. This shows that named sequence organization can merge related materials and avoid storing every event as an independent record.

The proposed method achieves the highest recall coverage, reuse success rate, and judgment accuracy. Its recall coverage reaches 0.965, which is much higher than the unordered memory method and the unnamed sequence method. This indicates that named sequence organization helps the agent retrieve a broader set of related materials. The reuse success rate reaches 0.991, showing that the recalled sequences are useful for later issue processing. The judgment accuracy also reaches 0.972, which indicates that recall-based comparison over named sequences can support more reliable familiar, slightly different, and new judgments.

It should be noted that the recall precision of the proposed method is lower than that of the two baselines. This does not mean that the proposed method is ineffective. The reason is that the proposed method recalls a broader set of potentially useful sequences, which improves coverage and reuse but may also introduce some weakly related materials. In contrast, unordered memory and unnamed sequence memory obtain high precision because they retrieve fewer relevant cases overall, but their recall coverage remains low.

Therefore, the advantage of the proposed method lies in the balance among compact storage, broad recall, and effective reuse. The result shows that named sequence organization does not merely store materials in order. It also provides an efficient indexing and comparison mechanism for future reasoning. By organizing complete issue-related materials into named sequences, the agent can recall more useful past experiences with a much smaller memory size and can better reuse them in later issue processing.

D. Verification III: Active Processing for Unmastered Issues

The third verification evaluates whether active processing can improve learning when an issue is not sufficiently mastered. This verification corresponds to the active processing mechanism in the proposed model. In open environments, an autonomous agent may encounter issues that cannot be reliably understood from the currently available materials. In such cases, simply storing the incomplete material may not be enough. The agent should actively collect additional observations, request further reasoning, or obtain more feedback to enrich the current issue-related sequence.

This verification also examines an important property of the proposed model: more processing can generate more learning material. If an issue is not processed, no structured issue-related training material is produced. If an issue is processed repeatedly or more deeply, additional materials can be generated and attached to the same sequence. Therefore, an initially strange or uncertain issue may gradually become more familiar as the corresponding sequence is enriched.

In this verification, the agent first builds an initial learning carrier from previously encountered events. Then it receives a set of test events, some of which are unfamiliar, partially understood, or weakly supported by feedback. For each test event, the agent estimates a mastery degree. If the mastery degree is lower than a predefined threshold, the proposed method activates further processing and adds the newly obtained materials to the current issue-related sequence.

Four methods are compared:

- **Passive method**: stores the current material without further processing.
- **Random observation method**: collects additional materials randomly.
- **LLM-only method**: requests one-shot LLM reasoning but does not continuously enrich the current sequence.
- **Proposed active method**: actively collects additional materials according to the mastery judgment and updates the same issue-related sequence.

Two main metrics are used. *Mastery degree* measures the agent's estimated understanding level after processing. *Action success rate* measures the proportion of issues successfully handled after processing. These metrics evaluate whether additional processing can improve the agent's understanding and handling of uncertain issues. In addition, the proposed method records the usefulness of additionally collected materials, which reflects whether active processing produces helpful training materials for the current issue.

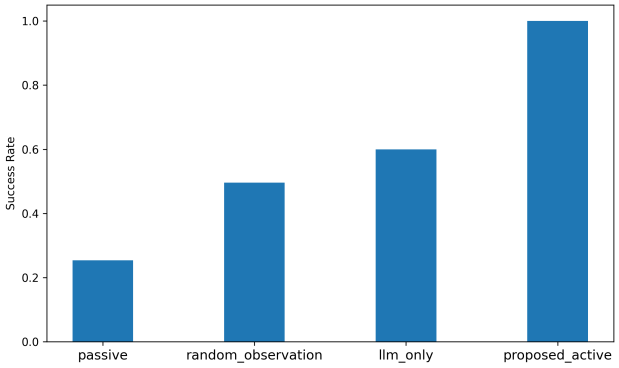


Fig. 4. Action success rate after different processing strategies.

TABLE III
RESULTS OF ACTIVE PROCESSING FOR UNMASTERED ISSUES.

Method	Mastery	Success
Passive method	0.611	0.254
Random observation	0.697	0.496
LLM-only method	0.742	0.600
Proposed active method	**0.808**	**1.000**

Fig. 4 shows the action success rate under different processing strategies. The proposed active method achieves the highest success rate, indicating that actively enriching the current issue-related sequence is more effective than passive storage, random observation, or one-shot LLM reasoning.

The detailed numerical results are shown in Table III. The passive method obtains the lowest mastery degree and action success rate because it does not enrich incomplete materials. The random observation method improves the results to some extent, but the collected materials are not always useful. The LLM-only method further improves mastery and success rate, showing that external reasoning can help analyze uncertain issues. However, because it does not continuously connect the new reasoning result with the complete issue-related sequence, its improvement remains limited.

The proposed active method achieves the highest mastery degree and action success rate. Its mastery degree reaches 0.808, which is higher than the passive method, random observation method, and LLM-only method. Its action success rate reaches 1.000 in the controlled simulation setting. This result indicates that actively enriching the current issue-related sequence is more effective than passively storing incomplete materials or using one-shot reasoning alone.

The key reason is that the proposed method does not treat additional observations or reasoning results as isolated new records. Instead, when an issue is judged as insufficiently mastered, the newly obtained materials are added to the same issue-related sequence. Therefore, the agent can gradually improve the completeness and reliability of the sequence. This makes the learning carrier more useful for later recall and reuse.

This verification supports the necessity of active processing in the proposed framework. If an agent only stores the current incomplete material, the learning carrier may contain weak or unreliable knowledge. If additional materials are collected randomly, the cost may increase without sufficient improvement. If the LLM is used only once, the reasoning result may help the current judgment but may not be fully integrated into the long-term learning carrier. In contrast, the proposed method uses mastery judgment to decide when further processing is needed and uses sequence enrichment to preserve the additional materials as part of the complete issue-related learning process.

More importantly, this result supports the idea that learning material is generated by processing. When the agent performs more thinking, comparison, reasoning, action monitoring, and feedback collection for an uncertain issue, more training materials are obtained. As these materials are accumulated in the same sequence, the agent's mastery degree increases, and a strange issue can gradually become less strange in later encounters.

It should be noted that this verification is conducted in a controlled simulation setting. Therefore, the result should be interpreted as evidence that active enrichment can improve unmastered issues under the designed conditions, rather than as a claim that the method always solves all uncertain issues in real environments.

E. Verification IV: Data Reduction and Learning Efficiency

The fourth verification evaluates whether the proposed model can reduce unnecessary storage while preserving useful learning materials. This verification corresponds to the data reduction ability of the proposed framework. Since the model treats all processed issue-related information as candidate learning material, a large number of materials may be generated during long-term operation. Therefore, the agent should not store every event as an independent record. Instead, it should merge familiar materials, preserve important variations, create new sequences only when necessary, and maintain reusable materials for future tasks.

In this verification, the agent receives a long stream of simulated issue events. Many events are repeated or only slightly different, while some events are genuinely new or important. The purpose is to test whether each method can reduce storage size without losing useful information. Later test events are used to evaluate whether the reduced learning carrier can still support future issue processing.

Four methods are compared:

- **Store-all method**: stores every incoming event as an independent record.
- **Fixed-threshold compression method**: compresses records according to a fixed similarity or category threshold.
- **Final-label method**: stores only final labels or judgments.
- **Proposed method**: selectively stores complete issue-related materials through named sequence construction, merging, and updating.

Four metrics are used. *Storage size* measures the number of stored records or sequences. *Compression ratio* measures the reduction of storage compared with the store-all method. *Useful information retention* measures the proportion of important materials preserved after reduction. *Future task performance*

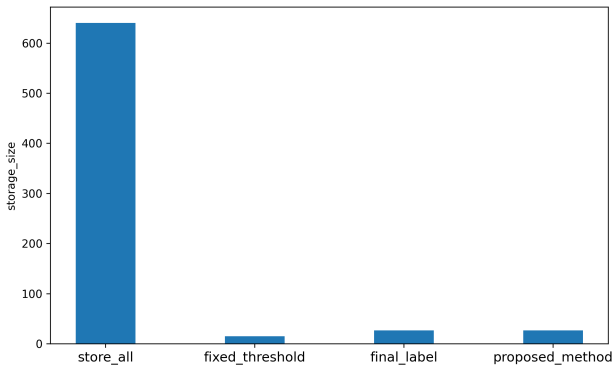


Fig. 5. Storage size comparison among different learning carriers.

TABLE IV
RESULTS OF DATA REDUCTION AND LEARNING EFFICIENCY.

Method	Size	Comp.	Ret.	Future Perf.
Store-all	640.00	0.000	**1.000**	0.998
Fixed-threshold	**15.00**	**0.977**	**1.000**	0.807
Final-label	26.80	0.958	0.350	0.309
Proposed method	26.80	0.958	**1.000**	**1.000**

measures whether the reduced learning carrier can support later issue processing.

Fig. 5 shows the storage size of different methods. The store-all method keeps all incoming events and therefore has the largest storage size. The proposed method greatly reduces the number of stored items by merging related materials into named sequences. Although its storage size is larger than that of the fixed-threshold compression method, it preserves richer issue-related context and supports better future reuse.

The detailed numerical results are shown in Table IV. The store-all method preserves all useful information, but its storage size is the largest. It stores 640.00 records and has no compression. This method can support future tasks well, but it is inefficient for long-term operation because storage grows directly with the number of incoming events.

The proposed method achieves a strong balance between storage reduction and useful information preservation. Its storage size is reduced from 640.00 to 26.80, and the compression ratio reaches 0.958. At the same time, its useful information retention remains 1.000, showing that important issue-related materials are preserved during reduction. Its future task performance also reaches 1.000, indicating that the reduced learning carrier can still support later issue processing effectively.

Compared with the store-all method, the proposed method achieves similar future task performance with much smaller storage size. This shows that the advantage of the proposed method is not simply higher task performance, but storage-efficient learning. Compared with the final-label method, the proposed method uses a similar storage size, but preserves much more useful information and achieves much better future task performance. This demonstrates that storing only final labels loses important issue-related context. Compared with the fixed-threshold compression method, the proposed method stores slightly more sequences, but achieves higher future task performance. This indicates that simple compression may preserve some useful information but may not organize it in a way that supports future reuse.

These results show that the proposed model can reduce unnecessary storage while preserving useful learning materials. By using recall-based judgment and named sequence organization, repeated familiar materials can be merged, slight variations can be stored as variations, and genuinely new or important issues can be preserved. Therefore, the learning carrier does not grow blindly with raw data, but maintains compact and reusable issue-related knowledge.

This verification supports the data reduction ability of the proposed framework. The proposed method does not store all raw events, nor does it reduce data only by final labels or fixed thresholds. Instead, it reduces data according to the complete issue-related context produced by the single thinking process. As a result, it maintains useful information for future recall and task processing while greatly reducing storage requirements.

V. CONCLUSION

This paper proposed a *Single-Thinking-Process-Based Learning of Complete Issue-Related Materials* model for autonomous agents. The main idea is that an agent should not learn only from isolated observations, final decisions, or external feedback. Instead, all materials processed during issue handling, including observations, action contexts, LLM interactions, recalled cases, comparison results, judgments, and feedback, should be treated as candidate learning materials.

The proposed model uses a unified thinking process to collect complete issue-related materials, organizes them into named sequences, and updates learning carriers for future recall and reuse. It also introduces active processing for unmastered issues, allowing the agent to collect additional observations, reasoning results, or feedback when the current material is insufficient. In this way, the agent can gradually discover learning targets from its own interaction process with reduced human guidance.

Controlled verification results show that the proposed model improves material completeness, recall coverage, reuse success, active improvement of unmastered issues, and storage-efficient learning. These results support the effectiveness of learning from complete issue-related materials rather than from fragmented or result-only records.

Future work will further apply the proposed model to more realistic embodied agents and robotic systems. In addition, more advanced sequence naming, recall, difference judgment, and learning carrier update methods can be explored to improve scalability and reliability in complex long-term environments.

REFERENCES

[1] D. B. Acharya, K. Kuppan, and B. Divya, "Agentic ai: Autonomous intelligence for complex goals—a comprehensive survey," *IEEe Access*, vol. 13, pp. 18 912–18 936, 2025.

[2] L. Qin, Q. Chen, Y. Zhou, Z. Chen, Y. Li, L. Liao, M. Li, W. Che, and P. S. Yu, "A survey of multilingual large language models," *Patterns*, vol. 6, no. 1, 2025.

[3] J. Jiang, K. Zhou, W. X. Zhao, Y. Song, C. Zhu, H. Zhu, and J.-R. Wen, "Kg-agent: An efficient autonomous agent framework for complex reasoning over knowledge graph," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 9505–9523.
[4] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022.
[5] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on robot learning*. PMLR, 2023, pp. 287–318.
[6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023," *URL https://arxiv.org/abs/2307.15818*, vol. 1, p. 2, 2024.
[7] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proceedings of the 36th annual acm symposium on user interface software and technology*, 2023, pp. 1–22.
[8] P. Du, "Memory for autonomous llm agents: Mechanisms, evaluation, and emerging frontiers," *arXiv preprint arXiv:2603.07670*, 2026.
[9] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," *arXiv preprint arXiv:2305.16291*, 2023.
[10] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," *Advances in neural information processing systems*, vol. 36, pp. 8634–8652, 2023.
[11] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.
[12] Z. Chen and B. Liu, "Continual learning and catastrophic forgetting," in *Lifelong Machine Learning*. Springer, 2022, pp. 55–75.

PLACE PHOTO HERE

Hong Su received the MS and PhD degrees, in 2006 and 2022, respectively, from Sichuan University, Chengdu, China. He is currently a researcher of Chengdu University of Information Technology Chengdu, China. His research interests include blockchain, large language model and human simulation computing.