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.

Toward Zero-Human-Intervention Autonomous Robot Learning: A Continuous Result-Driven Self-Reward and Correction Framework

Hong Su

suguest@126.com

Chengdu University of Information Technology

Research Article

Keywords: Autonomous robots, Self-reward learning, Delayed outcome attribution, Zero Human Intervention

Posted Date: April 14th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9398282/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.

Toward Zero-Human-Intervention Autonomous Robot Learning: A Continuous Result-Driven Self-Reward and Correction Framework

Hong Su

***Abstract*—Autonomous robots operating in complex real-world environments require the ability to continuously improve their behavior without human-provided reward annotation or online intervention. However, robot actions often produce delayed and multi-factor consequences, making it difficult to correctly associate later outcomes with earlier actions and to perform reliable autonomous self-reward and self-correction. In this paper, we propose a continuous result-driven self-reward and correction framework for autonomous robots. The framework enables robots to collect full-process behavioral, environmental, and internal reasoning records, perform delayed outcome discovery and temporal backtracking, generate unified internal–external self-reward signals, and revise earlier reward judgments when later evidence reveals them to be incomplete or incorrect. It also supports self-intervention for improving causal verification and policy reliability. Experimental results show that, compared with conventional baselines, the proposed framework improves delayed-outcome attribution accuracy from 15.93% to 38.36%, increases safety score from 66.1 to 98.5, and reduces persistent wrong reward rate of 48.77%.**

***Index Terms*—Autonomous robots, Self-reward learning, Delayed outcome attribution, Zero Human Intervention**

I. Introduction

Autonomous robots are expected to operate for extended periods in complex and evolving real-world environments, where predefined rules and manually engineered feedback are often insufficient to support sustained adaptation [1]. To achieve long-term autonomy, a robot should not only execute actions, but also continuously learn from the consequences of its own behavior. In particular, it should be able to determine whether previous actions should be strengthened, weakened, or corrected according to the results they eventually produce, without relying on human-provided reward annotation during long-term operation.

However, fully autonomous self-improvement remains difficult in practice. A central challenge is that the result of a robot action is often not immediately observable. In many real environments, an action may produce its true consequence only after several later transitions, and the final observed outcome may depend on a combination of multiple earlier actions, environmental conditions, and internal reasoning processes. As a result, immediate reward assignment can be misleading, since it may reinforce the most recent action rather than the true historical cause. Another difficulty is that many important aspects of autonomous behavior are not reflected solely by external outcomes. A robot may appear to complete a task successfully while its internal reasoning process is unsafe, inefficient, or inconsistent with long-term objectives. Conversely, useful internal reasoning, such as detecting hidden risk or discovering a corrective strategy, may contribute significantly to later success even before a visible external reward appears. These properties make conventional fixed and externally oriented reward mechanisms insufficient for long-horizon autonomous robot learning.

Meanwhile, existing robotic learning approaches still have several limitations. Many reinforcement-learning-based methods assume that reward signals are either externally defined or can be directly obtained from observable task outcomes [2]. Such methods work well in controlled environments with relatively clear feedback, but they are less suitable for open settings in which consequences are delayed, partially observable, and causally entangled. Similarly, although some works incorporate self-reflection, planning, or LLM-based reasoning into robotics, they often focus on task decomposition, action generation, or execution guidance rather than on the broader problem of how a robot should autonomously obtain, analyze, and revise reward from its own long-term experience. As a result, the problem of continuous self-reward and self-correction without human intervention remains insufficiently addressed.

To address this problem, this paper proposes a *continuous result-driven self-reward and correction framework* for autonomous robots, with the goal of enabling robots to autonomously collect feedback from the environment and from their own internal processes, without requiring human involvement in reward annotation during operation. The main idea is that an autonomous robot should continuously collect full-process information generated during operation, including environmental observations, executed actions, internal reasoning traces, program execution records, and related contextual data, and then use these records to discover delayed outcomes, analyze causal relations, generate self-reward, and revise past judgments. Instead of defining reward only from final external task results, the proposed framework combines *external reward* and *internal reward*. External reward is derived from observable consequences such as success, failure, safety, or efficiency, whereas internal reward is derived from the robot's own cognitive activities, such as identifying risk, finding

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

a valid solution, or maintaining consistency with expected goals. Moreover, the framework does not treat reward as permanently fixed once assigned. When later evidence shows that an earlier reward judgment was incomplete or incorrect, the robot can revise that judgment and reduce the influence of previous mistakes. To further improve causal reliability, the framework also supports self-intervention, allowing the robot to perform additional reasoning or controlled probing actions when passive observation is insufficient to confirm the true cause of an observed result.

The proposed framework is motivated by a simple but important view of autonomous intelligence: a robot should not only act in the world, but also continuously observe the long-term consequences of its own behavior, reason about why those consequences occurred, and correct its previous understanding when new evidence appears. Under this view, autonomous learning is not a one-time mapping from action to reward, but a continuous closed-loop process involving behavior execution, full-process data collection, delayed-result discovery, multi-factor attribution, internal–external reward generation, reward revision, and policy correction. This design moves autonomous robot learning beyond fixed immediate reward assumptions and toward a more realistic form of self-improvement in open environments.

The main contributions of this paper are summarized as follows:

- We propose a continuous result-driven self-reward and correction framework for autonomous robots, enabling robots to autonomously obtain, analyze, and revise reward from full-process experience without human reward annotation during operation.
- We introduce a delayed outcome analysis mechanism with temporal backtracking and multi-factor attribution, allowing the robot to associate later results with the historical actions, reasoning processes, and contextual factors that actually produced them.
- We design an internal–external self-reward mechanism and a revisable reward mechanism, enabling the robot not only to learn from observable task outcomes, but also to evaluate internal cognitive quality and correct earlier reward judgments when later evidence becomes available.

The remainder of this paper is organized as follows. Section II reviews related work. Section III presents the proposed continuous result-driven self-reward and correction framework. Section IV reports the verification results. Section V concludes the paper and discusses future directions.

II. Related Work

A. Delayed Reward and Temporal Credit Assignment

A long-standing challenge in autonomous learning is temporal credit assignment, i.e., determining which earlier actions are responsible for a later outcome when rewards are delayed. Conventional reinforcement learning methods often address this problem through temporal-difference learning, eligibility traces, or backward reward propagation mechanisms [3, 4]. Such methods have achieved strong performance in many benchmark settings, but delayed and sparse reward scenarios remain difficult, especially when the true consequence of an action emerges only after many later transitions or depends on multiple intervening factors [4, 5].

Recent studies have further emphasized that temporal credit assignment remains a fundamental bottleneck in deep reinforcement learning [4, 5]. When the temporal gap between action and consequence becomes large, learning can become unstable, because recent but non-causal actions may be mistakenly reinforced while the truly responsible historical actions are not sufficiently credited [4]. This challenge becomes even more severe in robotic environments, where delayed consequences may depend not only on physical execution but also on environmental dynamics, internal decisions, and interacting contextual factors. Therefore, although existing temporal credit assignment methods provide important foundations, they are not fully sufficient for autonomous robots that must reason over long-horizon, partially observable, and causally entangled outcomes.

In contrast to methods that mainly propagate externally observed reward backward along trajectories, our work focuses on a broader delayed-result setting for autonomous robots. The proposed framework does not only consider temporal distance, but also performs result-driven historical backtracking over full-process traces, including actions, observations, internal reasoning records, and execution context. This allows delayed outcomes to be analyzed in a more causally structured way rather than through reward propagation alone.

B. Intrinsic Motivation and Internal Reward

Another relevant line of research concerns intrinsic motivation and internally generated reward. Intrinsic reward has been widely studied as a mechanism for encouraging exploration, curiosity, novelty seeking, uncertainty reduction, or empowerment when external reward is sparse or incomplete [6, 7]. These approaches enrich agent learning by providing additional signals beyond explicit task feedback, and they have played an important role in improving autonomy in reinforcement learning [6, 7].

However, most intrinsic motivation methods are designed primarily for exploration or representation learning. Their internal reward is often based on novelty, prediction error, information gain, or controllability, rather than on the semantic quality of the agent's own reasoning process [6, 7]. For autonomous robots operating in long-horizon real environments, useful internal signals may arise not only from curiosity or novelty, but also from whether the robot's internal thinking detects hidden risk, discovers a corrective strategy, or remains aligned with expected long-term goals. In such cases, internal reward should reflect the quality and usefulness of internal cognitive activities rather than only the statistical rarity of observations.

Our work is related to intrinsic motivation in that it also introduces internally generated learning signals, but differs in both purpose and mechanism. Instead of rewarding exploration alone, we define internal reward from the robot's own reasoning-related activities and their consistency with long-term objectives. Moreover, internal reward in our framework

is integrated with delayed external outcomes and revisable judgment, enabling the robot to evaluate both what happened in the environment and whether its internal cognitive process contributed positively or negatively to that outcome.

C. Large Language Models for Robotics and Autonomous Reasoning

Large language models have recently attracted significant interest in robotics because of their capabilities in language understanding, task decomposition, semantic reasoning, and sequential decision support. Representative studies such as SayCan use language models together with grounded affordance estimation to select feasible robot skills [8], while PaLM-E extends embodied reasoning by integrating multimodal robotic and perceptual inputs into a large language model [9]. Other works such as Voyager demonstrate that LLM-based agents can continuously explore, accumulate reusable skills, and perform iterative self-improvement in open-ended environments [10]. Recent surveys further show that language-based reasoning and grounding are becoming increasingly important for robot planning, communication, and embodied intelligence [11, 12].

Despite these advances, most existing LLM-for-robotics studies focus on task planning, action generation, language grounding, or skill composition [8, 9, 10]. Although some of them incorporate feedback and self-reflection, they usually do not address the more specific problem studied in this paper: how a robot should autonomously obtain, analyze, revise, and reuse reward signals from its own long-term experience. In particular, current LLM-based robotic frameworks typically emphasize *what action should be taken next*, whereas our focus is on *how later outcomes should be traced back, interpreted, rewarded, and corrected* after actions have already been executed.

Therefore, our work complements the existing LLM-for-robotics literature from a different perspective. Rather than using LLMs mainly as planners or policy generators, we use them as reasoning components for delayed outcome backtracking, internal reward evaluation, and reward revision. This shifts the role of LLMs from pure action guidance toward continuous self-reward and self-correction, which is more directly aligned with long-term autonomous robot learning.

D. Summary of Differences

In summary, existing studies on temporal credit assignment address delayed reward propagation, intrinsic motivation methods provide internally generated signals, and LLM-based robotic systems enhance planning and embodied reasoning [4, 6, 8, 9, 10]. However, these lines of work do not directly provide a unified framework in which an autonomous robot continuously collects full-process traces, discovers delayed outcomes, attributes them to historical actions and reasoning processes, generates both internal and external reward, and revises earlier reward judgments when later evidence becomes available. The proposed framework is intended to fill this gap by combining delayed-result analysis, internal–external self-reward, and revisable correction into a continuous closed-loop learning process for autonomous robots.

III. Model

A. Overview of the Proposed Model

To enable truly autonomous learning in open environments, a robot should not rely only on manually designed rewards, externally annotated feedback, or immediate observable outcomes. Instead, it should be able to continuously collect the consequences of its own behavior, analyze these consequences over time, and use the resulting judgments to strengthen, weaken, or even correct previous actions. Motivated by this requirement, we propose a *continuous result-driven self-reward and correction model* for autonomous robots.

The core idea of the proposed model is that robot learning should be based on a full closed loop rather than a single-step action–reward response. After executing an action, the robot does not simply wait for an externally provided reward. Instead, it continuously records multi-source information generated before, during, and after its behavior, including environmental observations, sensor readings, internal state changes, program execution traces, action logs, and interaction records with the large language model (LLM). These continuously accumulated records provide the basis for identifying both immediate and delayed consequences of previous actions.

A central challenge is that the effect of a robot action may not appear immediately. Some actions may influence later states only after a long delay, and some outcomes may be jointly caused by multiple actions, internal decisions, and environmental factors. Therefore, the proposed model introduces a delayed-result analysis mechanism. When an important result, abnormal event, or unexpected success is observed, the system retrospectively analyzes historical records and uses LLM-assisted temporal reasoning to infer which earlier actions, contextual conditions, and internal decision processes may have contributed to the current outcome. In this way, the model is not limited to short-term feedback, but can assign credit or blame across long temporal horizons.

In addition, the proposed model does not define reward only from external task outcomes. Instead, it combines *external reward* and *internal reward*. External reward is derived from observable consequences in the environment, such as task completion, safety, efficiency, or failure. Internal reward is derived from the robot's own reasoning and cognitive processes, such as discovering a hidden risk, identifying an inconsistency, finding a valid solution after extended reasoning, or determining that the actual result is close to the internally expected target. This design allows the robot to evaluate not only "what happened in the environment" but also "whether its internal thinking helped or harmed the final objective."

Another important feature of the model is that reward judgment is not fixed once generated. In real environments, an early evaluation may later prove to be incomplete or incorrect because more evidence becomes available. To address this issue, the proposed framework incorporates a revisable reward mechanism, in which previously assigned rewards can be re-evaluated and corrected when later observations provide stronger evidence. Thus, the robot is able not only to reinforce successful actions and suppress harmful ones, but also to revise former mistaken judgments and gradually override

the influence of earlier wrong conclusions. This property is important for long-term autonomy, where initial interpretations are often uncertain.

When retrospective reasoning alone is insufficient to determine the true cause of a result, the model further allows the robot to perform self-intervention. Such intervention may occur at the cognitive level, through deeper reasoning over historical traces, or at the behavioral level, through actively executing probing actions to test whether a hypothesized factor truly causes the observed outcome. This intervention-based verification improves the reliability of action–result attribution, especially in environments where multiple coupled factors produce overlapping effects.

Overall, the proposed model forms a continuous autonomous learning loop: *action execution*, *full-process data collection*, *delayed-result discovery*, *multi-factor attribution*, *internal–external reward generation*, *reward revision*, *self-intervention*, and *policy correction*. Through this loop, the robot can continuously learn from its own experience without requiring human-provided reward labels. The model therefore provides a foundation for autonomous robots to achieve long-term self-improvement, self-correction, and adaptive behavior evolution in complex real-world environments.

B. Full-Process Autonomous Data Collection

A fundamental prerequisite for autonomous self-reward and self-correction is that the robot must be able to obtain the consequences of its own behavior without relying on external manual annotation. However, in real environments, the consequence of an action is often not represented by a single observation. Instead, it is distributed across perception, control, software execution, environmental change, and internal reasoning processes. Therefore, the proposed model adopts a full-process autonomous data collection mechanism that continuously records multi-source information related to the robot's behavior and its potential consequences.

The first principle of this mechanism is *comprehensiveness*. Rather than recording only final task outcomes or a small number of sensor values, the robot collects data from multiple channels throughout operation. These channels include external sensory observations, internal state variables, action execution records, control signals, program traces, system logs, LLM interaction records, LLM outputs, and other context information that may later help explain why a result occurred. This design is necessary because delayed results and hidden causal relations often cannot be recovered from sparse records. Only when the robot preserves sufficiently rich traces can it later analyze whether an earlier action, an internal reasoning error, a software execution condition, or an environmental factor contributed to a later success or failure.

The second principle is *continuity*. Data collection should not be limited to a single moment before or after an action. Instead, the robot continuously records information across the full temporal process, including pre-action context, action execution, intermediate transitions, and post-action consequences. Such continuous recording is particularly important in autonomous systems because some effects emerge gradually and may become observable only after a sequence of subsequent events. Without temporal continuity, the system may observe a result but lose the evidence needed to relate that result to earlier decisions. Thus, the proposed mechanism maintains persistent traces over time so that later delayed-result analysis can reconstruct the historical evolution of events.

The third principle is *internal–external unification*. Traditional robotic learning often focuses mainly on external observations, such as task completion, collision, or sensor-detected state changes. However, for the proposed model, the robot must also evaluate the role of its own internal processes. Therefore, the collected data include not only external observations but also internal thinking-related records. These internal records may include prompts sent to the LLM, returned reasoning outputs, intermediate self-checking messages, internally predicted targets, confidence estimates, and reasoning-derived hypotheses. Such information is important because a final outcome may depend not only on physical execution but also on whether the robot's internal reasoning correctly interpreted the environment or generated an appropriate plan.

Formally, at time step t, the robot collects a unified record

$$D_t = \{O_t^{ext}, O_t^{int}, A_t, C_t, L_t, M_t\},$$

where O_t^{ext} denotes external observations from sensors and the environment, O_t^{int} denotes internal observations and reasoning-related states, A_t denotes executed actions, C_t denotes control and program execution context, L_t denotes system logs and trace information, and M_t denotes memory entries or historical summaries retained for later analysis. Over time, these records form a trajectory

$$\mathcal{D}_{1:T} = \{D_1, D_2, \ldots, D_T\},$$

which serves as the evidence base for delayed-result discovery, temporal backtracking, reward generation, and later correction.

To make the collected information usable for subsequent reasoning, the model further organizes these records into temporally indexed and semantically linked traces. In particular, action records are linked to nearby internal reasoning states, environmental observations are aligned with execution timestamps, and important events are marked for later retrieval. This trace-oriented organization helps the system efficiently identify candidate historical segments when a later outcome must be explained. It also reduces the risk that important causal evidence is lost in long autonomous runs.

Overall, the full-process autonomous data collection module provides the informational foundation of the entire framework. Without comprehensive, continuous, and internally enriched records, the robot would be unable to discover delayed consequences, attribute outcomes to earlier actions, or revise previous judgments. By persistently collecting and organizing full-process data, the proposed model enables later modules to perform autonomous result analysis and self-reward assignment in a principled manner.

C. Delayed Outcome Discovery and Temporal Backtracking

A key difficulty in autonomous self-reward learning is that the consequence of a robot action is often delayed. Unlike simplified settings in which each action is immediately followed

by a clear reward signal, real-world robotic behavior may produce effects only after many later transitions. Moreover, an observed success or failure at the current moment may have been caused not only by the most recent action, but also by a sequence of earlier actions, internal decisions, and environmental interactions. Therefore, the proposed model includes a delayed outcome discovery and temporal backtracking module to identify historical behaviors that may be responsible for later results.

The first task of this module is to detect delayed outcomes. Given the continuously collected trajectory data, the system monitors the evolution of task states, environmental changes, internal reasoning states, and system events. When an important event is detected, such as a success, failure, abnormal change, safety issue, or unexpected improvement, the system does not directly assign its effect to the current action only. Instead, it treats the observed event as a potential delayed consequence and initiates a retrospective analysis over a historical time range. This allows the robot to search for actions and conditions that may have contributed to the current result, even if they occurred much earlier.

To support this process, the model constructs a candidate temporal window for backtracking. Suppose that an outcome of interest is observed at time t. Rather than examining the entire history without restriction, the system first extracts a candidate window

$$W_t = \{D_{t-k}, D_{t-k+1}, \ldots, D_t\},$$

where k is determined according to the type of event, task dynamics, and the expected temporal latency of possible causes. For rapidly responding events, k may be small; for slow or accumulated effects, k may be larger. This windowed design improves computational efficiency and focuses the analysis on the most relevant historical segment.

Within the candidate window, the system performs temporal backtracking to infer possible action–result relations. The proposed framework uses LLM-assisted reasoning to analyze temporally ordered records and identify plausible causal chains. Specifically, the LLM is provided with structured summaries of actions, state changes, internal reasoning traces, and major events within the candidate window, and is asked to infer which earlier behaviors or internal decisions may explain the currently observed outcome. This process is not limited to a single-step relation. Instead, it may produce a chain of dependencies such as

$$(a_{t-k}, s_{t-k}) \rightarrow (a_{t-k+1}, s_{t-k+1}) \rightarrow \cdots \rightarrow r_t,$$

where r_t denotes the delayed result observed at time t. In this way, the model can assign credit or blame over long temporal horizons rather than only to the most recent action.

An important advantage of temporal backtracking is that it can incorporate both external and internal evidence. For example, an unsafe result may not be caused solely by a physical motion, but also by an earlier reasoning error, an incorrect prediction, or a misleading intermediate conclusion generated during planning. Similarly, an eventual success may arise because the internal reasoning process discovered a useful strategy long before the final visible result appeared. By tracing both behavioral and cognitive history, the model can identify richer and more realistic explanations of delayed outcomes.

Because delayed outcome analysis is inherently uncertain, the framework does not force a binary conclusion at the first step. Instead, it generates a set of candidate historical causes together with their explanatory strengths or confidence levels. Let

$$\mathcal{H}_t = \{(h_1, c_1), (h_2, c_2), \ldots, (h_n, c_n)\}$$

denote the set of candidate causes for the observed outcome at time t, where each h_i is a historical action, state, reasoning event, or composite segment, and c_i is its associated confidence. These candidates are then forwarded to the subsequent attribution and reward modules for deeper evaluation. This design is important because some delayed effects may involve multiple interacting causes rather than a single dominant factor.

Overall, the delayed outcome discovery and temporal backtracking module enables the robot to move beyond immediate reward assumptions and to reason over historical trajectories in a principled manner. It provides the bridge between current observations and past behaviors, making it possible for the system to autonomously discover long-range action–result dependencies. This capability is essential for robots operating in complex environments, where true consequences often emerge gradually and where meaningful self-improvement depends on correctly understanding the temporal origin of success and failure.

D. Multi-Factor Result Attribution

After delayed outcomes and candidate historical causes have been identified, the robot must further determine how the observed result should be attributed. This step is necessary because real-world outcomes are rarely produced by a single isolated action. Instead, they often emerge from the combined influence of multiple actions, environmental conditions, internal reasoning processes, software execution states, and temporal interactions among these factors. Therefore, the proposed model includes a multi-factor result attribution module to analyze how different factors jointly contribute to a later success, failure, or abnormal event.

The first challenge of attribution is that the relation between behavior and result is typically many-to-many. On the one hand, a single action may generate multiple consequences, including both beneficial and harmful ones. For example, a fast motion may improve task efficiency while simultaneously increasing safety risk. On the other hand, a single observed result may be jointly caused by multiple earlier factors. A navigation failure, for instance, may arise from an inaccurate sensor reading, an incorrect reasoning conclusion, a poor action choice, and a delayed environmental change. Thus, the attribution problem cannot be simplified into a one-action–one-result mapping. Instead, the system must evaluate coupled dependencies among multiple candidate factors.

To formalize this process, suppose an important result r_t is observed at time t, and the delayed backtracking module produces a candidate set

$$\mathcal{H}_t = \{h_1, h_2, \ldots, h_n\},$$

where each h_i may denote an action, an internal reasoning event, a contextual state, a software execution condition, or a short temporal segment. The attribution module estimates the contribution of each candidate factor, or of combinations of factors, to the observed result. This can be represented as

$$r_t \leftarrow f(h_1, h_2, \ldots, h_n),$$

where $f(\cdot)$ is not assumed to be a simple linear mapping, but a structured attribution function that considers temporal order, interaction, and contextual dependence.

To support meaningful attribution, the proposed model organizes candidate factors into several categories:

- **Action factors**, including executed robot motions, control commands, and strategy choices;
- **Environmental factors**, including external state changes, disturbances, and sensed scene conditions;
- **Internal reasoning factors**, including LLM outputs, intermediate plans, inferred hypotheses, and self-evaluation results;
- **Execution factors**, including software states, program traces, controller status, and logging events;
- **Temporal interaction factors**, including accumulated effects, ordering relations, and dependencies across multiple time steps.

By jointly analyzing these categories, the system can avoid attributing outcomes only to immediately visible actions while ignoring hidden upstream causes.

Because different candidate factors may overlap in effect, the module assigns not only contribution scores but also attribution confidence. Let

$$A_t = \{(h_i, \alpha_i, c_i)\}_{i=1}^{n},$$

where α_i denotes the estimated contribution strength of factor h_i to result r_t, and c_i denotes the confidence of this attribution. A high α_i with high c_i indicates that the factor is likely to be an important cause, whereas a low-confidence attribution suggests that the relation remains uncertain and may require later verification. This uncertainty-aware design is important because delayed and multi-factor effects are often only partially observable.

In addition to individual attribution, the model also considers composite attribution. That is, some results may not be explained by any single factor alone, but only by the joint effect of several factors. For example, a system error may occur only when a specific action is executed under a particular environmental condition after an earlier reasoning mistake. To capture such cases, the attribution module can generate compound candidates such as

$$(h_i, h_j), \quad (h_i, h_j, h_k),$$

and estimate whether these combinations better explain the observed result than isolated factors. This makes the attribution process more suitable for complex robotic environments, where causal relations are often intertwined rather than independent.

The output of the attribution module is not yet the final reward itself. Instead, it provides a structured explanation of which historical factors are likely to deserve reinforcement, suppression, correction, or further verification. These attribution results are then used by the reward generation and intervention modules. Factors with strong positive attribution may later receive reinforcing reward, factors with negative attribution may receive penalty or suppression, and factors with high uncertainty may trigger deeper reasoning or active intervention.

Overall, the multi-factor result attribution module transforms delayed historical evidence into structured causal responsibility estimates. It enables the robot to move beyond simplistic immediate-feedback learning and to reason about how multiple behavioral, environmental, and internal factors jointly shape later outcomes. This capability is essential for autonomous self-improvement, because correct reward assignment depends not only on observing that a result occurred, but also on understanding why it occurred and which earlier factors truly deserve credit or blame.

E. Internal–External Self-Reward Generation

After delayed outcomes have been discovered and their possible causes have been attributed, the robot must convert these analyses into learning signals that can guide future behavior. In conventional reinforcement learning, such signals are usually defined as externally given rewards associated with task success or failure. However, for a truly autonomous robot operating without human intervention, relying only on explicit external feedback is insufficient. Many important behaviors do not produce immediate observable rewards, and some valuable internal cognitive processes may contribute significantly to eventual success even before any external result becomes visible. Therefore, the proposed model introduces an internal–external self-reward generation mechanism that jointly evaluates environmental outcomes and internal thinking-related processes.

The first component is *external reward*, which is derived from observable consequences in the environment. These consequences may include task completion, task failure, safety preservation, collision occurrence, efficiency improvement, resource consumption, or other measurable changes in the physical world. Let R_t^{ext} denote the external reward associated with the result observed at time t. Its value may be positive when the outcome supports the robot's objective, negative when the outcome harms the objective, and near zero when the observed consequence is weak or neutral. Unlike manually assigned reward labels, however, R_t^{ext} is inferred by the robot itself from the collected trajectory data and the attributed relation between historical factors and current outcomes.

The second component is *internal reward*, which is one of the main characteristics of the proposed framework. Internal reward is generated from the robot's own cognitive and reasoning processes rather than only from directly visible environmental changes. This is important because autonomous

behavior often depends on whether internal thinking helped the robot move toward its objective, avoided hidden danger, identified an inconsistency, or discovered a useful solution. Let R_t^{int} denote the internal reward at time t. It may be assigned based on several types of internal evidence.

First, internal reasoning may be evaluated according to whether the resulting state is close to the robot's internally expected target. For example, if safety is treated as an expected objective during autonomous driving, then reaching the destination without collision or dangerous events indicates that the internal planning and monitoring process remained aligned with the expected goal. Second, internal reasoning may receive positive reward when it discovers an effective solution to a difficult or previously unresolved problem. If the robot spends a long time searching for an explanation, plan, or corrective strategy and eventually finds a valid one, the successful reasoning activity itself should be rewarded. Third, internal reasoning may also receive reward when it detects hidden risks, inconsistencies, or errors before they fully manifest as external failures. Similarly, misleading or incorrect reasoning that later contributes to undesirable outcomes should receive negative internal reward. Thus, internal reward reflects whether the robot's own thinking process improved or harmed future decision quality.

Based on these two components, the overall self-reward is defined as a fusion of external and internal rewards:

$$R_t = \alpha R_t^{ext} + \beta R_t^{int},$$

where α and β are nonnegative weighting coefficients that control the relative importance of environmental outcomes and internal cognitive contributions. This formulation allows the robot to adapt reward generation to different tasks. In tasks dominated by physical execution, α may be larger. In tasks requiring heavy reasoning, diagnosis, or long-horizon planning, β may also play a major role. More generally, the fusion mechanism ensures that the robot does not learn solely from visible success or failure, but also from whether its internal reasoning process was useful, safe, and goal-consistent.

An important property of the proposed reward mechanism is that reward assignment is not restricted to the current time step only. Because many results are delayed, the reward observed at time t may need to be redistributed to earlier actions, internal decisions, or reasoning events that contributed to the current outcome. Suppose the attribution module identifies a set of historical factors

$$A_t = \{(h_i, \alpha_i, c_i)\}_{i=1}^{n},$$

where h_i denotes a candidate historical factor, α_i its contribution strength, and c_i the confidence of attribution. Then the reward associated with r_t can be retrospectively assigned to these factors, for example through

$$R(h_i \mid r_t) = g(R_t, \alpha_i, c_i),$$

where $g(\cdot)$ denotes a redistribution rule that allocates stronger reward or penalty to factors with larger contribution and higher confidence. In this way, the robot can strengthen useful earlier actions, suppress harmful earlier decisions, and learn across long temporal horizons.

The reward generated by this module is therefore not merely a scalar evaluation of what happened at the current moment. Instead, it is a structured learning signal grounded in both external reality and internal cognition, and capable of being propagated backward over historical trajectories. This design is especially important for autonomous robots because meaningful learning often depends on hidden cognitive quality, delayed consequences, and partial observability rather than on immediate surface-level outcomes alone.

Overall, the internal–external self-reward generation module converts attributed results into autonomous learning signals that are both richer and more realistic than conventional externally defined rewards. By jointly considering environmental consequences and internal reasoning activities, the proposed model enables the robot to evaluate not only whether an outcome was good or bad, but also whether its own thinking process contributed positively or negatively to that outcome. This provides a more principled basis for long-term autonomous adaptation, self-improvement, and self-correction.

F. Revisable Learned Reward Mechanism

Although the internal–external self-reward generation module provides autonomous learning signals, the reward assigned at one moment should not be regarded as permanently correct. In real environments, early judgments are often made under incomplete information. A behavior that initially appears beneficial may later be found to have caused a hidden problem, while an action that seems harmful in the short term may eventually support a better long-term result. Therefore, the proposed model introduces a revisable learned reward mechanism, which allows reward judgments to be continuously updated, corrected, and improved as more evidence becomes available.

The central idea of this mechanism is that reward itself should be treated as a learnable and correctable object rather than a fixed label. Traditional reward design often assumes that once a reward is observed or assigned, it can be directly used to update policy parameters. However, for long-horizon autonomous robots, this assumption is too rigid. When delayed effects, hidden causal chains, and multi-factor interactions are present, an early reward judgment may later prove incomplete, biased, or wrong. Thus, the proposed framework stores historical reward assignments together with their supporting evidence, so that these judgments can be revisited during later analysis.

To formalize this idea, let

$$\hat{R}_t^{(0)}$$

denote the initial reward assigned to a historical factor or action at time t, and let

$$\hat{R}_t^{(m)}$$

denote its revised estimate after incorporating later evidence up to revision step m. Rather than assuming

$$\hat{R}_t^{(m)} = \hat{R}_t^{(0)},$$

the framework allows

$$\hat{R}_t^{(m+1)} = \phi\left(\hat{R}_t^{(m)}, E_t^{(m+1)}\right),$$

where $E_t^{(m+1)}$ denotes newly collected evidence relevant to the original judgment, and $\phi(\cdot)$ denotes a revision function that updates the previous reward estimate. In this way, reward is continuously refined as the robot observes more outcomes, performs more analyses, and acquires stronger causal evidence.

To support efficient revision, the model maintains for each important historical judgment a reward memory item containing at least the following elements:

- the historical factor or action segment being evaluated;
- the originally assigned reward value;
- the evidence and attribution basis used for the assignment;
- the confidence level of that judgment;
- later evidence that supports, weakens, or overturns the previous conclusion.

This design allows the system not only to remember what reward was previously assigned, but also to remember *why* that judgment was made. As a result, later correction is based on traceable reasoning rather than arbitrary replacement.

In addition to explicit revision, the framework introduces a learned reward evaluation component. This component can be implemented as a neural model or another trainable evaluator that takes as input historical trajectories, attribution summaries, internal reasoning traces, and later observed consequences, and outputs an updated reward estimate. Formally, for a historical factor h_t, the learned reward estimator can be expressed as

$$\tilde{R}(h_t) = \psi(\mathcal{D}_{1:T}, A_{t:T}, Z_{t:T}),$$

where $\mathcal{D}_{1:T}$ denotes the collected trajectory, $A_{t:T}$ denotes attribution information related to h_t, and $Z_{t:T}$ denotes later reasoning and verification records. The role of this learned evaluator is not to replace explicit analysis, but to help generalize reward revision patterns from previous experience. If the system repeatedly observes that certain early reward judgments are often corrected under similar later evidence, the learned evaluator can gradually improve future reward estimation.

An important property of the revisable reward mechanism is its ability to override the influence of former incorrect judgments. Suppose an earlier action was mistakenly rewarded because its harmful delayed effects had not yet appeared. Once later evidence reveals that this action actually contributed to failure or danger, the system should not continue reinforcing it simply because it received a past positive label. Instead, the revised reward mechanism reduces or reverses the effect of the earlier incorrect reward and replaces it with a better-informed estimate. This process is similar to human self-correction, where a person may later realize that a former conclusion was wrong and repeatedly reinforce the corrected understanding until the earlier mistaken belief loses influence.

To reflect this idea in learning, the policy update should depend not only on the initial reward but also on the revised reward history. For a historical factor h_t, its effective learning signal may be defined as

$$R_t^{eff} = \sum_{m=0}^{M} \omega_m \hat{R}_t^{(m)},$$

where ω_m denotes the weight assigned to the m-th reward estimate, typically giving greater weight to later and more reliable revisions. Through this design, the robot can gradually suppress the impact of outdated judgments and emphasize better-supported ones.

Overall, the revisable learned reward mechanism enables the robot to treat reward as an evolving assessment rather than a one-time label. This is essential for real autonomous systems, because meaningful self-improvement requires not only learning from past feedback, but also correcting past feedback when it is later shown to be incomplete or wrong. By combining traceable reward memory, evidence-based revision, and learned reward evaluation, the proposed model supports a more flexible and reliable form of long-term autonomous learning.

G. Self-Intervention for Causal Verification

Although delayed backtracking and multi-factor attribution can identify plausible causes of an observed result, retrospective analysis alone is not always sufficient to establish reliable causal responsibility. In complex environments, multiple factors may co-occur, some important variables may be partially unobserved, and several candidate explanations may remain consistent with the available historical records. As a result, the robot may still be uncertain whether a hypothesized factor truly caused the observed outcome or merely correlated with it. To address this limitation, the proposed model introduces a self-intervention mechanism that allows the robot to actively verify candidate causal relations through additional reasoning and behavior.

The basic idea is that when passive observation is insufficient, the robot should not stop at uncertain attribution. Instead, it should generate and execute interventions designed to test whether a candidate factor is genuinely responsible for a result. Such intervention can occur at two levels: cognitive intervention and behavioral intervention. Cognitive intervention performs deeper internal analysis over the available traces, while behavioral intervention modifies the external process by executing probing actions or changing relevant conditions. Together, these two forms of intervention improve the reliability of causal verification and reduce the risk of learning from spurious correlations.

The first form is *cognitive intervention*. In this case, the robot does not immediately act in the environment, but instead performs deeper analysis on the historical data and current hypotheses. For example, it may re-examine the temporal order of events, compare similar historical episodes, test alternative explanations through internal simulation, or ask the LLM to reason under counterfactual assumptions such as whether the same result would likely have occurred if a particular earlier action had not been taken. Let

$$Q_t = \{q_1, q_2, \ldots, q_m\}$$

denote a set of candidate explanatory questions generated for an uncertain result at time t. The cognitive intervention process then evaluates these questions over the stored trajectory and reasoning records to refine the confidence of candidate causes. This mechanism is useful when direct physical intervention is expensive, unsafe, or temporarily impossible.

The second form is *behavioral intervention*. When internal analysis remains insufficient, the robot may actively execute actions intended to perturb the process and observe whether the suspected result changes accordingly. Suppose that a candidate factor h_i is believed to contribute to result r_t. The robot can perform an intervention

$$do(h_i = \tilde{h}_i),$$

where $\tilde{h}_i$ denotes a modified value, alternative action, or controlled removal of the original factor. The robot then observes the subsequent outcome under this modified condition. If the result changes in a manner consistent with the causal hypothesis, the confidence in that hypothesis increases; otherwise, the attribution may be weakened or rejected. In practice, such interventions may include repeating a task with one factor changed, delaying an action, replacing a reasoning-derived plan, disabling a suspected software module, or deliberately probing a specific environmental variable.

To make intervention safe and purposeful, the proposed model does not intervene arbitrarily. Instead, interventions are generated only when the expected benefit of causal clarification is sufficiently high and when safety constraints allow the probing process. Formally, for a candidate intervention u, the system evaluates an intervention utility

$$J(u) = \lambda_1 G(u) - \lambda_2 C(u) - \lambda_3 S(u),$$

where $G(u)$ denotes the expected gain in causal certainty, $C(u)$ denotes execution cost, and $S(u)$ denotes safety risk. Only interventions with sufficiently favorable utility are selected. This design is important because autonomous robots should not compromise task success or safety merely to collect explanatory evidence.

The output of intervention is not only a revised binary judgment of whether a hypothesis is true or false. Instead, the system uses intervention results to update attribution strengths, confidence values, and related reward judgments. Suppose the attribution module has produced

$$A_t = \{(h_i, \alpha_i, c_i)\}_{i=1}^{n}.$$

After intervention, the confidence of a candidate factor may be revised as

$$c_i' = \eta(c_i, v_i),$$

where v_i denotes the verification evidence obtained through cognitive or behavioral intervention, and $\eta(\cdot)$ is a confidence update function. Factors whose causal role is confirmed may receive stronger subsequent reward or penalty assignment, while factors whose influence is not supported may be downgraded or removed from consideration. Thus, intervention directly improves later learning quality.

An important advantage of self-intervention is that it transforms the robot from a passive observer into an active causal learner. Instead of merely waiting for naturally occurring evidence, the robot can deliberately create informative situations that help distinguish true causes from coincidental associations. This is especially valuable in multi-factor environments, where purely observational learning may repeatedly misattribute results due to overlapping effects, hidden dependencies, or sparse feedback.

Overall, the self-intervention module provides an essential verification capability for autonomous self-reward learning. It complements retrospective analysis by enabling the robot to actively test uncertain hypotheses through deeper reasoning and controlled behavioral probing. By incorporating intervention-based causal verification, the proposed model improves the reliability of action–result attribution and establishes a more solid foundation for reward correction, policy update, and long-term autonomous self-improvement.

H. Continuous Re-Analysis and Self-Correction Loop

In long-term autonomous operation, not all meaningful consequences can be identified at the moment they first occur. Some results emerge only after extended interaction, some earlier judgments remain uncertain for a long time, and some initially accepted explanations may later be contradicted by new evidence. Therefore, autonomous learning should not treat analysis as a one-time process performed only when an event is first detected. Instead, the robot should continuously revisit previously observed behaviors, previously assigned rewards, and previously accepted explanations. To achieve this, the proposed model incorporates a continuous re-analysis and self-correction loop that persistently re-examines historical records and updates past judgments when better evidence becomes available.

The first function of this loop is persistent background analysis. While the robot continues executing tasks, the system monitors incoming data streams and compares newly observed events with previously stored historical traces, unresolved attribution hypotheses, and prior reward assignments. When new evidence is related to an earlier action or earlier judgment, the system reopens the corresponding historical segment for further examination. This means that learning is not limited to the moment of immediate feedback; rather, historical experience remains analytically active and may be reinterpreted later. Such a mechanism is particularly important in environments where delayed outcomes are common, because the significance of an earlier behavior may become visible only after many subsequent steps.

Let

$$\mathcal{M} = \{m_1, m_2, \ldots, m_K\}$$

denote the set of stored memory items, where each item may correspond to an important action segment, an attributed cause, a reward judgment, or an unresolved causal hypothesis. At a later time t', newly collected evidence $E_{t'}$ is matched against these memory items. If a relevant relation is detected, the system performs a re-analysis update

$$m_k' = \rho(m_k, E_{t'}),$$

where $\rho(\cdot)$ denotes a revision operator that updates the stored item using the new evidence. In this way, prior judgments remain revisable throughout the robot's continued operation.

The second function of the loop is self-correction of past decisions. This self-correction may occur at several levels. First, the robot may correct past action evaluations, for example by deciding that a previously rewarded action should actually be penalized once its delayed harmful effects become known. Second, the robot may correct past reasoning conclusions, such as recognizing that an earlier LLM-generated explanation was incomplete or wrong. Third, the robot may correct its attribution structure, for example by shifting credit or blame from one candidate factor to another when later evidence provides clearer causal distinction. Thus, self-correction is not limited to a scalar reward update; it includes correction of the broader explanatory structure that supports later learning.

To support this process, the framework distinguishes between *provisional* and *stabilized* judgments. A provisional judgment is one that has been assigned under limited evidence and remains open to later revision. A stabilized judgment is one whose supporting evidence has become sufficiently strong and consistent over time. This distinction is useful because not all decisions need to be treated equally. Some early conclusions may be used temporarily for behavior guidance while still remaining revisable, whereas others may gradually become reliable enough to serve as stronger learning anchors. Formally, for a historical judgment z_i, its stability may be represented by a score

$$s_i \in [0, 1],$$

where larger values indicate stronger evidential support and lower expected revision probability. The re-analysis loop updates both the content of the judgment and its stability as more evidence accumulates.

Another important role of continuous re-analysis is to detect missed or hidden consequences. A result that appeared irrelevant at one time may later reveal itself to be important when viewed together with subsequent events. Similarly, a weak anomaly that was initially ignored may later be recognized as an early sign of a more serious issue. By continuously revisiting prior traces, the robot can uncover consequences that were not sufficiently visible during the initial analysis stage. This capability is essential because autonomous systems in open environments often operate under partial observability, and meaningful interpretation may require integrating evidence across distant times.

The self-correction loop also supports gradual suppression of outdated knowledge. If a previously accepted explanation is repeatedly contradicted by later evidence, its influence on subsequent learning should diminish. Conversely, if a revised explanation is repeatedly supported, its influence should strengthen. This can be expressed through an evolving belief weight

$$w_i^{(t+1)} = \chi\left(w_i^{(t)}, e_i^{(t+1)}\right),$$

where $w_i^{(t)}$ denotes the current weight of a stored judgment or explanation, and $e_i^{(t+1)}$ denotes new supporting or contradicting evidence. Through this process, the robot gradually shifts its learning basis away from outdated conclusions and toward better-supported ones.

Overall, the continuous re-analysis and self-correction loop ensures that autonomous learning remains dynamic rather than static. It allows the robot to treat experience as an evolving source of knowledge whose meaning may improve over time. By persistently revisiting old actions, old rewards, and old explanations in light of new evidence, the proposed model enables deeper long-term adaptation and reduces the risk of locking future behavior onto early but mistaken judgments. This loop is therefore a central mechanism for robust self-improvement in complex real-world environments.

I. Policy and Behavior Update

After reward generation, reward revision, causal verification, and continuous re-analysis, the robot must ultimately transform the accumulated knowledge into behavioral change. Without this final step, the previous modules would only provide retrospective explanation rather than actual autonomous improvement. Therefore, the proposed framework includes a policy and behavior update module that uses the corrected reward signals and validated attribution results to strengthen beneficial behaviors, suppress harmful ones, and adapt future decision-making.

The first role of this module is action strengthening and weakening. For historical factors or actions that are repeatedly associated with positive long-term outcomes, the robot should increase their likelihood of being selected in similar future contexts. Conversely, for behaviors that are linked to unsafe, inefficient, or failed outcomes, the robot should reduce their future influence. Unlike conventional immediate-feedback updates, however, the present framework relies on corrected and possibly revised reward signals. This means that policy adjustment is based not only on what appeared beneficial at one moment, but on what remained beneficial after delayed analysis, attribution, and self-correction. As a result, the robot learns from more reliable experience rather than from raw short-term impressions alone.

Let h_t denote a historical action-related factor at time t, and let R_t^{eff} denote its effective revised learning signal. Then, in abstract form, the policy update seeks to increase the preference for factors with larger positive R_t^{eff} and decrease the preference for factors with negative R_t^{eff}. If $\pi_\theta(a \mid s)$ denotes the robot policy parameterized by θ, the update may be conceptually expressed as

$$\theta \leftarrow \theta + \gamma \nabla_\theta \mathcal{U}(\theta; R_t^{eff}),$$

where γ is the learning rate and $\mathcal{U}(\cdot)$ is a policy optimization objective driven by the effective revised reward. The exact form of $\mathcal{U}$ is left open, since the proposed framework is intended to be compatible with different underlying learning backbones rather than restricted to a single training algorithm.

The second role of the module is correction-oriented behavior adaptation. This is another important difference from standard reinforcement learning. If later evidence shows that a previously preferred action was actually based on mistaken attribution or incomplete reward evaluation, the system should

not merely stop reinforcing it; it should explicitly adapt away from it. Similarly, if a previously suppressed behavior is later found to have contributed positively under long-horizon analysis, the system should recover or strengthen that behavior again. Thus, policy adaptation under the proposed model is inherently bidirectional: it supports both reinforcement of correct historical conclusions and correction of previously mistaken behavioral tendencies. This property is particularly important for autonomous robots in nonstationary environments, where earlier assumptions may lose validity over time.

In addition to parameter-level policy adjustment, the framework also supports behavior-level adaptation. That is, the robot may modify not only the probability of selecting low-level actions, but also higher-level behavioral structures such as planning preferences, safety thresholds, diagnostic routines, exploration strategies, or intervention policies. For example, if internal reasoning repeatedly reveals that certain classes of plans tend to create delayed safety risks, the robot may adapt by increasing caution in those contexts. If repeated intervention confirms that some type of self-check is valuable before action execution, the robot may incorporate that self-check into future behavior more systematically. Therefore, the update process applies not only to primitive actions but also to broader decision procedures that shape long-term autonomy.

To express this more generally, let

$$\mathcal{B} = \{b_1, b_2, \ldots, b_m\}$$

denote the set of updateable behavior components, which may include low-level actions, reasoning patterns, planning routines, safety constraints, and intervention strategies. The system updates these components according to their effective learning signals:

$$b_i \leftarrow \Gamma(b_i, R_i^{eff}, \xi_i),$$

where R_i^{eff} is the revised effective reward associated with component b_i, and ξ_i denotes contextual information or constraints relevant to updating that component. This generalized formulation reflects the fact that autonomous improvement in robots often requires coordinated adaptation across both physical behavior and internal decision mechanisms.

Another important aspect of this module is stability control. Because the proposed framework allows reward revision and repeated self-correction, the update mechanism should avoid overreacting to a single uncertain revision. Instead, behavioral changes should reflect both reward magnitude and evidential reliability. Hence, updates may be weighted by confidence, stability, or revision consistency, so that strongly supported conclusions exert greater policy influence than provisional or weakly supported ones. This reduces the risk of oscillatory learning caused by premature or uncertain corrections.

Overall, the policy and behavior update module closes the learning loop of the proposed framework. It converts autonomous data collection, delayed analysis, attribution, reward generation, reward revision, and intervention-based verification into concrete future behavioral improvement. Through this module, the robot can progressively strengthen beneficial actions, suppress harmful ones, correct earlier mistaken tendencies, and refine both its external behavior and internal decision procedures. As a result, the entire framework supports long-term autonomous adaptation in a way that is more robust, self-corrective, and causally informed than conventional immediate-reward learning.

IV. Verification

A. Experimental Setup

To verify the effectiveness of the proposed framework, we design three experiments corresponding to its three core capabilities: delayed outcome handling, internal reward effectiveness, and revisable reward correction. All experiments are conducted in a simulated autonomous robot environment, where the robot continuously interacts with the environment, observes delayed consequences, and updates its behavior without human-provided reward labels during execution.

The environment is designed to exhibit three common properties of real autonomous systems. First, some action consequences are delayed and only become visible after several later transitions. Second, some task outcomes depend not only on external execution but also on internal reasoning quality. Third, some early reward judgments are made under incomplete evidence and may need later correction. During operation, the robot records external observations, executed actions, reasoning-related states, and trace information, forming the full-process trajectory used for delayed-result analysis, self-reward generation, and reward revision.

We compare the proposed framework with the following baselines:

- **Immediate External Reward (IER):** reward is assigned only according to immediately observable external outcomes.
- **Delayed External Reward (DER):** reward is assigned after delayed observation, but without internal reward or reward revision.
- **Fixed Internal–External Reward (FIER):** both external and internal reward are considered, but once assigned, reward is never revised.
- **Proposed Method:** the full framework with delayed outcome backtracking, internal–external self-reward generation, and revisable reward correction.

For delayed-outcome handling and revisable reward correction, the proposed method is instantiated with LLM-assisted reasoning. In the delayed-outcome experiment, the LLM receives a structured historical trajectory window and predicts the most likely historical action responsible for the delayed failure. In the revisable-reward experiment, the LLM receives the initial reward judgment together with later evidence and decides whether the stored reward sign should be retained or revised. For internal reward verification, the comparison focuses on the behavioral effect of introducing internal reward into the learning process.

We use attribution accuracy and wrong attribution rate for delayed-outcome verification; success rate, safety score, and average total reward for internal-reward verification; and reward correction accuracy, persistent wrong reward rate, and final policy quality for revisable-reward verification.

TABLE I
RESULTS OF DELAYED OUTCOME VERIFICATION UNDER LLM-ASSISTED BACKTRACKING

Method	Attribution Accuracy (%)	Wrong Attribution Rate (%)
IER	0.0	100.0
DER	15.9	84.1
Proposed Method	38.4	61.6

B. Verification of Delayed Outcome Handling

This experiment tests whether the proposed framework can correctly associate a delayed outcome with the historical action that actually caused it. The simulator constructs delayed-result episodes in which some risky actions do not produce immediate failure but instead trigger latent effects that only become observable several steps later. For each delayed event, the simulator records the true responsible historical action, enabling direct quantitative evaluation.

In the current LLM-instantiated setting, the experiment is conducted over 120 episodes and produces 477 delayed-result events, averaging about 3.98 delayed events per episode. IER attributes the outcome only to the most recent action, DER attributes the outcome within a short delayed window, and the proposed method performs LLM-assisted temporal backtracking over stored full-process traces.

Table I reports the overall results. The proposed method achieves an attribution accuracy of 38.36%, clearly outperforming DER (15.93%) and IER (0.00%). Its wrong attribution rate is correspondingly reduced to 61.64%, compared with 84.07% for DER and 100.00% for IER. These results show that immediate reward assignment is fundamentally unsuitable for delayed-outcome settings, while simple delayed assignment without structured backtracking remains limited.

Figure 1 shows the attribution-accuracy curve. The proposed method maintains a clear advantage over both baselines across evaluation checkpoints, indicating that explicit trajectory-based backtracking remains useful even when attribution is delegated to an LLM instead of a hand-crafted scoring rule. A closer inspection by delay length further shows that DER performs slightly better only for very short delays, whereas the proposed method becomes superior for delays 3–8, where true delayed causal reasoning is required.

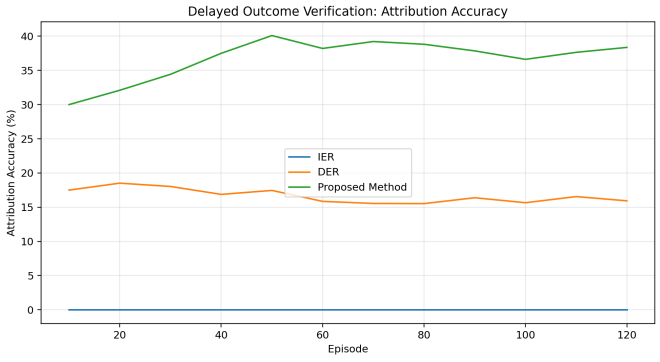


Fig. 1. Attribution accuracy in delayed outcome verification.

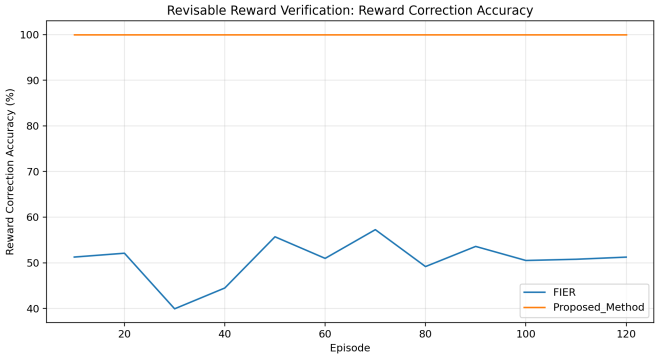


Fig. 2. Reward correction accuracy in revisable reward verification.

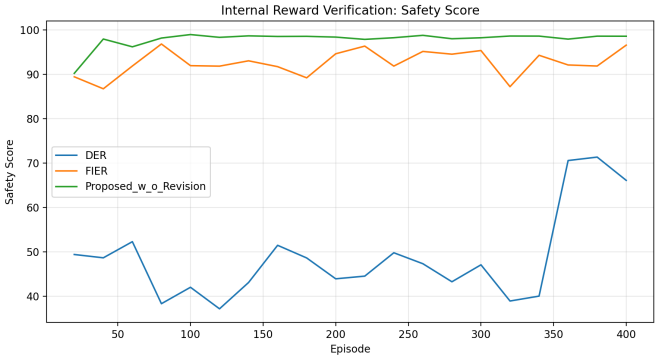


Fig. 3. Safety score in internal reward verification.

C. Verification of Internal Reward Effectiveness

The second experiment examines whether internal reward provides useful learning signals beyond external outcomes alone. The task is designed to be partially observable and risk-sensitive: a robot may complete the task quickly while still relying on unsafe or poor-quality internal decisions, whereas internal reasoning may identify hidden risk or better strategies before external consequences fully appear.

Three methods are compared: DER, FIER, and the proposed method without reward revision. The evaluation uses success rate, safety score, and average total reward. As shown in Table II, DER achieves the highest success rate at 100.0%, but its safety score is only 66.1, and its average total reward is 24.5. Once internal reward is introduced, both FIER and the proposed method become much safer and obtain higher overall reward quality. FIER reaches a safety score of 96.5 and an average total reward of 33.2, while the proposed method without revision further improves these values to 98.5 and 36.2, respectively.

These results indicate that external-only reward tends to encourage aggressive task completion while ignoring hidden safety costs. This explains why DER obtains perfect task completion but much poorer safety-related performance. By contrast, internal reward makes the learning signal more sensitive to reasoning quality and hidden risk. The gain in safety score from 66.1 for DER to 96.5 for FIER and 98.5 for the proposed method confirms that internal reward helps the robot value beneficial cognitive activities such as early risk detection

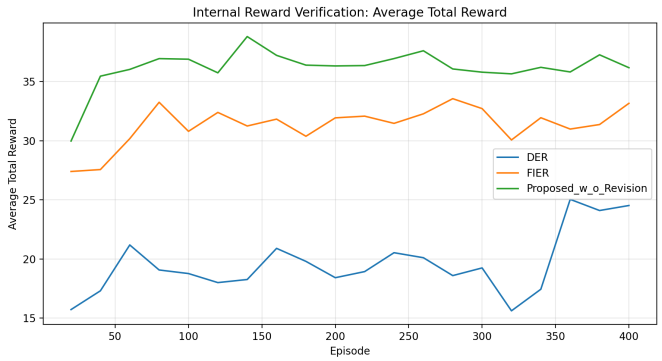


Fig. 4. Average total reward in internal reward verification.

TABLE II
RESULTS OF INTERNAL REWARD VERIFICATION

Method	Success Rate (%)	Avg. Reward	Safety Score
DER	100.0	24.5	66.1
FIER	93.3	33.2	96.5
Proposed w/o Rev.	83.3	36.2	98.5

and goal-consistent planning.

Figure 3 and Figure 4 further illustrate the effect of internal reward. The safety-score curve shows a large and stable advantage for methods with internal reward, and the total-reward curve shows that the proposed method achieves the highest overall behavior quality despite a lower pure completion rate than DER. This means that success rate alone is insufficient for evaluating long-horizon autonomous learning in risk-sensitive settings.

D. Verification of Revisable Reward Mechanism

The third experiment evaluates whether the proposed revisable reward mechanism can correct earlier reward judgments when later evidence reveals that the original evaluation was incomplete or wrong. The environment intentionally includes misleading short-term outcomes: some actions appear beneficial initially but later cause hidden failure, while others appear unfavorable at first but later prove beneficial.

In the current LLM-instantiated setting, FIER keeps the initial reward sign fixed, whereas the proposed method provides the LLM with the initial reward judgment and later evidence and asks it to determine whether the reward sign should be revised. Table III reports the results. The proposed method achieves 100.0% reward correction accuracy, compared with 51.23% for FIER. At the same time, its persistent wrong reward rate is reduced from 48.77% to 0.0%, and final policy quality improves from 24.43 to 27.69.

These results show that fixed reward is strongly affected by early misjudgment. Because FIER never revises its initial reward assignment, nearly half of its stored reward judgments remain inconsistent with the true long-term direction. In contrast, the proposed framework uses later evidence to update earlier reward judgments through LLM-assisted revision, fully eliminating persistent wrong reward in the current controlled setting. Although the increase in final policy quality is smaller

TABLE III
RESULTS OF REVISABLE REWARD VERIFICATION UNDER LLM-ASSISTED REWARD REVISION

Method	Corr. Acc. (%)	Wrong Rate (%)	Policy Quality
FIER	51.2	48.8	24.4
Proposed	100.0	0.0	27.7

in magnitude than the correction-related gains, it still demonstrates that reward revision yields a measurable behavioral benefit rather than only a bookkeeping improvement.

Figure 2 shows the reward-correction-accuracy curve. The gap between FIER and the proposed method remains large throughout the evaluation, showing that once delayed evidence is available, revisable reward can reliably overwrite misleading early judgments and align learning with long-term task value.

E. Discussion

The three experiments jointly support the main design objectives of the proposed framework. First, delayed-outcome handling shows that autonomous robots should not rely only on immediate reward assignment: the proposed method improves attribution accuracy from 15.93% for DER and 0.00% for IER to 38.36%. Second, internal reward shows that externally observable outcomes alone are insufficient: compared with DER, the proposed method improves safety score from 66.1 to 98.5 and average total reward from 24.5 to 36.2. Third, revisable reward shows that reward should not be treated as a fixed one-time label: compared with FIER, the proposed method improves reward correction accuracy from 51.23% to 100.0% and reduces persistent wrong reward rate from 48.77% to 0.0%.

Taken together, these results indicate that autonomous robot learning benefits from treating reward generation as a continuous process rather than a static event. The proposed framework continuously collects evidence, analyzes delayed consequences, incorporates internal cognitive quality, and revises earlier judgments when needed. This makes the learning process more consistent with open environments, where causality is often delayed, partially observable, and subject to reinterpretation.

The current verification still has limitations. All experiments are conducted in simulation, which cannot fully represent real robotic deployment. The quality of LLM-instantiated reasoning also depends on prompt design and the effective historical context window, as reflected by the degradation of delayed-outcome attribution when the causal distance becomes too large. Moreover, the internal reward design remains partly task-dependent, and revisable reward introduces additional computational overhead because it requires storing historical traces and repeatedly re-analyzing earlier judgments. Future work should therefore examine more efficient implementations and more realistic physical-robot evaluations.

V. CONCLUSION

This paper presented a continuous result-driven self-reward and correction framework for autonomous robots operating without human-provided reward annotation during long-term

execution. The framework enables a robot to collect full-process experience, discover delayed outcomes, trace them back to historical actions and reasoning processes, generate internal–external self-reward, and revise earlier reward judgments when later evidence becomes available. In this way, autonomous learning is treated as a continuous closed-loop process rather than as a fixed immediate action–reward mapping.

The verification results showed that the proposed framework improves delayed-outcome attribution, strengthens safety-aware learning through internal reward, and effectively corrects misleading early reward judgments through revisable reward correction. These findings suggest that continuous self-reward and self-correction provide a promising direction for long-horizon autonomous robot learning in complex environments.

Future work will extend the framework to more realistic physical robotic systems, more open-ended environments, and more efficient large-model-assisted reasoning under longer historical contexts.

REFERENCES

[1] K. Shaheen, M. A. Hanif, O. Hasan, and M. Shafique, "Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks," *Journal of Intelligent & Robotic Systems*, vol. 105, no. 1, p. 9, 2022.

[2] Z. Li, G. Jin, R. Yu, Z. Chen, N. Li, W. Han, L. Xiong, B. Leng, J. Hu, I. Kolmanovsky *et al.*, "A survey of reinforcement learning-based motion planning for autonomous driving: Lessons learned from a driving task perspective," *arXiv preprint arXiv:2503.23650*, 2025.

[3] M. A. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, learning, and optimization*, vol. 12, no. 3, p. 729, 2012.

[4] E. Pignatelli, J. Ferret, M. Geist, T. Mesnard, H. van Hasselt, O. Pietquin, and L. Toni, "A survey of temporal credit assignment in deep reinforcement learning," *arXiv preprint arXiv:2312.01072*, 2023.

[5] D. Arumugam and T. L. Griffiths, "On temporal credit assignment and data-efficient reinforcement learning," in *Finding the Frame Workshop at RLC 2025*.

[6] A. G. Barto, "Intrinsic motivation and reinforcement learning," in *Intrinsically motivated learning in natural and artificial systems*. Springer, 2012, pp. 17–47.

[7] A. Aubret, L. Matignon, and S. Hassas, "An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey," *Entropy*, vol. 25, no. 2, p. 327, 2023.

[8] 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.

[9] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[10] 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.

[11] V. Cohen, J. X. Liu, R. Mooney, S. Tellex, and D. Watkins, "A survey of robotic language grounding: Tradeoffs between symbols and embeddings," *arXiv preprint arXiv:2405.13245*, 2024.

[12] W. Hunt, S. D. Ramchurn, and M. D. Soorati, "A survey of language-based communication in robotics," *arXiv preprint arXiv:2406.04086*, 2024.



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.