DUAL-SCALE WORLD MODELS FOR LLM AGENTS TOWARDS HARD-EXPLORATION PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-based agents have seen promising advances, yet they are still limited in "hard-exploration" tasks requiring *learning new knowledge through exploration*. We present GLoW, a novel approach leveraging dual-scale textual world models, maintaining a trajectory frontier of high-value discoveries at the global scale, while learning from local trial-and-error in exploration through a Multi-path Advantage Reflection mechanism which infers advantage-based progress signals to guide exploration. To evaluate our framework for hard-exploration, we tackle the Jericho benchmark suite of text-based games, where GLoW achieves a new state-of-the-art performance for LLM-based approaches. Compared to state-of-the-art RL-based methods, our approach achieves comparable performance while requiring 100-800× fewer environment interactions.[1]

1 INTRODUCTION

While LLM agents (Yao et al., 2023; Sumers et al., 2024; Wang et al., 2024) excel at leveraging vast pre-trained knowledge in tasks such as robotic planning, software engineering, and web automation (Ahn et al., 2022; Yang et al., 2024; 2025), they are reportedly limited in *hard-exploration problems* (Sutton & Barto, 2018; Ecoffet et al., 2019). Hard exploration problems are typically characterized by large state–action spaces, deceptive local optima, and sparse rewards. These factors often trap naive exploration in local optima, such that exploration fails to reach deeper states with rewards. For LLM agents, such problems pose two central challenges: (1) Global learning, for maintaining long-term knowledge of valuable discoveries during exploration, (2) Local trial-and-error, for quickly refining exploration policies from sparse environmental feedback. Current LLM agent approaches such as ReAct (Yao et al., 2023) or Reflexion (Shinn et al., 2023) support local trial-and-error, but lack mechanisms for long-term knowledge accumulation. Consequently, LLM agents fall short on hard-exploration tasks that humans can often solve effectively (Cui et al., 2025; Phan et al., 2025).

In this work, we introduce **G**lobal-**Lo**cal **W**orld Models (GLoW), a framework enabling effective exploration in hard-exploration problems through dual-scale *textual world models* for global and local learning. Rather than predicting transition dynamics, these world models encode structured knowledge from exploration trajectories to guide the LLM agent. Our approach builds on the Go-Explore (Ecoffet et al., 2019) algorithm, which achieves breakthroughs on hard-exploration problems by enhancing the exploration capabilities of RL and LLM-based agents (Lu et al., 2025). The key idea of Go-Explore is to store discovered states into a *state archive*. Then, based on this archive, Go-Explore decomposes hard-exploration into alternating between: (1) a *selection* phase, choosing a *promising* state from the archive to return to, and (2) an *exploration* phase, to continue discovering new states from the selected state. In its original implementation, Go-Explore used hand-crafted heuristics for selection, and random action sampling for exploration, while later work, such as IGE (Lu et al., 2025) improved selection to leverage LLM inference.

In this work, our core insight is that both selection and exploration require structured learning from past exploration experiences, but at different scales: we first enrich beyond an archive of isolated states, by additionally maintaining a trajectory frontier, which keeps the full temporal context of how high value states were reached and why progress stalled, into a **global world model** for richer structured learning. This allows an LLM-based analysis across the frontier to infer high-value regions

[1] Code will be open sourced after blind review

as well as bottleneck states with high future potential, enabling principled state selection in GLoW, beyond heuristic or LLM-internalized notions of interestingness. At the local scale, to guide exploration actions from the state, we draw insights that advantage-based rewards better capture progress signals than Q-values (Kazemnejad et al., 2025; Setlur et al., 2025): Our Multi-path Advantage Reflection mechanism explores multiple trajectories from the same starting state and leverages LLM reasoning to infer *intermediate advantages at key state-action pairs*. Through these advantage signals, the **local world model** enables controlled exploration under sparse environmental feedback.

To evaluate the capability of LLM agents in hard-exploration problems, we study the Jericho benchmark suite of text-based games (Hausknecht et al., 2019), where the SOTA methods have been RL-based solutions (Hausknecht et al., 2019; Ammanabrolu & Hausknecht, 2020; Guo et al., 2020) with ε-greedy or softmax exploration or MCTS-based exploration (Jang et al., 2021; Shi et al., 2025). However, they suffer from poor sample efficiency, relying on extensive trial-and-error which requires **hundreds of thousands** of environment interactions. Meanwhile, existing LLM agents have been insufficient to address the challenge of learning from exploration in Jericho games, showing limited performance compared to humans (Cui et al., 2025; Phan et al., 2025).

Through extensive experiments, we show that GLoW improves the performance of LLM-based agents while achieving orders of magnitude improvement in sample efficiency compared to RL baselines. Our contributions are summarized as follows:

- We propose GLoW, a novel LLM agent framework for hard-exploration problems through global-local world models.
- We conduct comprehensive comparisons with existing agent approaches (RL, MCTS, LLM) and ablation studies to validate components of our method.
- We achieve a new state of the art for LLM-based approaches on Jericho, achieving comparable performance with RL-based SOTA, while reducing environment interactions required by 100-800×.

2 BACKGROUND

Jericho Benchmark The Jericho benchmark (Hausknecht et al., 2019) remains an unsolved hard-exploration problem, where the text-based game environments provide two fundamental challenges (Ammanabrolu & Riedl, 2021): (1) partial observability, requiring agents to construct models of the world from local textual descriptions, and (2) combinatorial state-action spaces. For example in Zork1, the game vocabulary has 697 words and up to five-word commands, resulting in $O(697^5) = 1.64 \times 10^{14}$ possible actions per step, though only a tiny fraction are grammatically coherent and contextually relevant. As a result, RL approaches, with simple exploration strategies, incur hundreds of thousands interactions to offset sample inefficiencies in exploration. This makes Jericho an ideal testbed for evaluating whether agents learn by exploring, rather than brute-force discovery.

Methods for Hard-Exploration Problems Go-Explore (Ecoffet et al., 2019) achieved breakthroughs in hard-exploration problems by maintaining an archive of discovered states as global knowledge to (1) *select* promising states and (2) *explore* from the state. Algorithm 1 illustrates this framework, contrasting the original Go-Explore (in gray) with our proposed approach GLoW (in blue). The original algorithm uses novelty-based heuristics for state selection and random actions for exploration. XTX (Tuyls et al., 2022) improves upon these with imitation learning for selection and DQN with curiosity rewards for exploration, while IGE (Lu et al., 2025) leverages LLM inference for both phases. Our approach introduces two key innovations: (1) a trajectory frontier $\mathcal{F}$ with LLM-based value decomposition for principled state selection, and (2) Multi-path Advantage Reflection (MAR) for learning from local exploration. Appendix B provides a detailed comparison across Go-Explore variants. Beyond the Go-Explore family, MCTS-based methods like MC-LAVE (Jang et al., 2021) and MC-DML (Shi et al., 2025) leverage tree search with language-driven exploration and LLM priors respectively, though requiring 400,000+ interactions.

3 METHOD

In this section, we describe the dual-scale learning paradigm of GLoW's textual world models.

Algorithm 1 Go-Explore with GLoW

procedure GO-EXPLORE(s_0, n_{iter}, n)
  $\mathcal{A} \leftarrow \{(s_0, 0)\}$ ▷ Archive of (state, score)
  $\mathcal{F} \leftarrow \emptyset$ ▷ Trajectory Frontier
  for $i = 1$ to n_{iter} **do**
    Go-Explore: $s_{next} \leftarrow select(\mathcal{A}) \propto \frac{1}{\text{visits}(s)^{\alpha}}$ ▷ Novelty-based heuristics
    GLoW: $W_{\text{global}} \leftarrow g_{\text{LLM}}(\mathcal{F})$ ▷ Principled value decomposition (Sec. 3.1)
      $s_{next} \leftarrow \text{align}_{\text{LLM}}(\mathcal{A}, W_{\text{global}})$
    Go-Explore: $\tau \leftarrow explore(s_{next}) \propto$ Random ▷ No learning
    GLoW:
    for $j = 1$ to n **do** ▷ LLM agent with advantage-driven exploration (Sec. 3.2)
      $\tau_j \leftarrow \pi_{explore}(s_{next}, W_{\text{local}}, \{\tau_1, \ldots, \tau_{j-1}\}, \mathcal{F})$
      $W_{\text{local}} \leftarrow \text{MAR}(\{\tau_1, \ldots, \tau_j\}, \mathcal{F})$
    end for
    Update $\mathcal{A}$, $\mathcal{F}$
  end for
end procedure

3.1 GLOBAL WORLD MODEL FOR STATE SELECTION

The global world model extracts value signals from accumulated exploration trajectories. Unlike traditional state-based archives, we maintain trajectories in a value-ranked frontier. The global world model additionally maintains LLM-generated trajectory analysis.

Value-Ranked Trajectory Frontier As the source of value information, the global world model maintains a trajectory frontier $\mathcal{F} = \{\tau_1, \tau_2, ..., \tau_k\}$, containing the k highest-value trajectories discovered during exploration, ranked by a value function $v : \mathcal{T} \rightarrow \mathbb{R}$. Each trajectory $\tau_i = (s_0^i, a_1^i, r_1^i, s_1^i, ..., a_T^i, r_T^i, s_T^i)$ represents a complete episode generated by the exploration policy π_{explore} defined by the LLM agent, where $s_t \in \mathcal{S}$ are states, $a_t \in \mathcal{A}$ are actions, and $r_t \in \mathbb{R}$ are rewards. For the trajectory value function v, we use the maximum cumulative reward achieved during the episode, $v(\tau_i) = \max_{t \in [1,T]} \sum_{j=1}^{t} r_j^i$. This is an effective choice for Jericho's sparse reward structure, where agents can encounter negative rewards or terminal failures. In contrast to state-only representations, which lose the context of action and observation sequences, preserving complete trajectories enables accurate credit assignment and value estimation in sparse-reward environments where success depends on precise action sequences. For instance, in Zork1, progressing past the troll requires first acquiring both the lantern and the sword before descending into the cellar, but only entering the cellar yields a reward. Analyzing complete trajectories, which may each capture different portions of these sequential dependencies, enables inferring across them that both items are necessary despite the sparse feedback.

The frontier evolves progressively through iterative exploration. When exploration from selected states (detailed in Section 3.2) produces trajectory τ_{new} with value $v(\tau_{\text{new}})$, the frontier is updated:

$$\mathcal{F}_{t+1} = \text{top-}k(\mathcal{F}_t \cup \{\tau_{\text{new}}\}, v) \quad (1)$$

This sliding window mechanism ensures the frontier maintains diverse high-value strategies, while allowing newly discovered superior trajectories to replace outdated ones. For any state s_i, we can derive the achieved value $v(s_i) = \max_{\tau \in \mathcal{F}, s_i \in \tau} v(\tau)$, representing the maximum value reached from state s_i across all frontier trajectories. By tracking complete trajectories, the frontier serves as both an estimator of achieved values and a repository of successful action sequences.

Motivation: Decomposing value for *select* and *explore* Inspired by UCB's value decomposition which balances exploitation with exploration bonus as:

$$\bar{V}(s) + c\sqrt{\frac{\log(N)}{n_s}}$$

where $\bar{V}(s)$ is the empirical mean value and the second term is the exploration bonus based on visit count n_s, we annotate two types of values v and v', corresponding to each term, by analyzing patterns across all frontier trajectories $\mathcal{F}$, to extract a set of critical global states with value annotations:

$$W_{\text{global}} = g_{\text{LLM}}(\mathcal{F}) = \{(s_1, v_1, v_1'), (s_2, v_2, v_2'), \ldots, (s_k, v_k, v_k')\} \quad (2)$$

Here, each (s_i, v_i, v'_i) represents a critical global state, key semantic landmarks such as exploration frontiers, bottlenecks, and milestones, identified from frontier analysis by a prompted LLM g_{LLM}, where v_i denotes the achieved value from s_i, while v'_i reflects LLM's estimate of future value potential. Importantly, this potential value v'_i cannot be derived from trajectory scores alone, requiring LLM's reasoning about why trajectories fail and what progress could be achieved by resolving current bottlenecks. For instance, a state where multiple trajectories fail might have *low achieved value*, but have *high potential value* when: (1) multiple high-value trajectories converge but fail to progress further, suggesting unexplored regions beyond, (2) partial solution patterns indicate missing components, or (3) environmental hints suggest valuable areas remain undiscovered. This implements a *semantic* form of optimism under uncertainty (Auer, 2003; Brafman & Tennenholtz, 2003) where UCB uses statistical bonuses while we derive optimistic values from LLM analysis of bottlenecks. See Appendix E.1 for a full example of W_{global} generated for Zork1.

Balancing Exploitation and Exploration in State Selection We maintain a state archive $\mathcal{A} = \{(s_i, \text{score}(s_i))\}$ containing discovered states with their achieved scores. Given W_{global}, we select the next exploration state s_{next} by balancing achieved and potential values via LLM as shown in Fig. 1-(a). We leverage $\text{align}_{\text{LLM}}$, an LLM-based state selection operation which evaluates how

---

Require: Frontier $\mathcal{F}$, State archive $\mathcal{A}$
Ensure: Selected state s_{next}

1: $W_{global} \leftarrow g_{LLM}(\mathcal{F})$ where
   $W_{global} = \{(s_1, v_1, v'_1), \dots, (s_k, v_k, v'_k)\}$
   v_i: achieved, v'_i: potential
2: **for** each state $s \in \mathcal{A}$ **do**
3: $\quad score[s] \leftarrow align_{LLM}(s, W_{global})$
4: **end for**
5: $s_{next} \leftarrow \arg\max_{s \in \mathcal{A}} score[s]$
6: **return** s_{next}

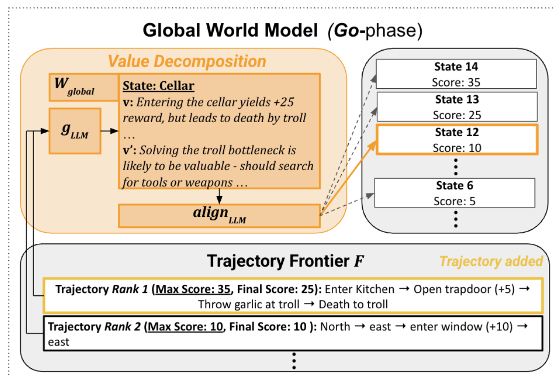

Figure 1: (a) Select procedure in GLoW, (b) Illustration of selection with Global World Model

well each archived state s aligns with the high-value patterns identified in W_{global} using a prompted LLM (see Appendix D.2 for the full prompt). Since W_{global} contains both achieved and potential values for key frontier states, this alignment naturally balances exploitation (favoring states similar to proven high-reward regions), with exploration (prioritizing states near identified bottlenecks with high potential). Fig. 1-(b) illustrates selection in GLoW with the Global World Model where a new trajectory (highlighted in gold) has been added to the frontier. Once a state is chosen, we replay the stored sequence of actions to return to the state[2], which becomes the starting point of the next exploration phase, described in the following section.

3.2 Local World Model for Exploration

In addition to the selection of states which align with exploration goals with high potential value, exploration can be enhanced by learning which actions are likely to lead to further progress, which is the objective of the local world model.

Motivation: From Q-values to Advantages for Exploration Existing LLM learning methods like self-reflection can be viewed as estimating state-action values (Q-values) from single trajectories. However, Q-value estimation from sparse rewards is notoriously high-variance (Sutton et al., 1999; Schulman et al., 2017), and we observe the same challenge in LLM-based learning: inferences from entire trajectories with sparse feedback are prone to incorrect causal attribution.

Drawing from RL theory, advantage functions $A(s, a) = Q(s, a) - V(s)$ reduce variance by comparing actions to a baseline rather than estimating absolute values. Recent work on process reward models (PRMs) further demonstrates that advantage-based rewards are more suited for exploration,

---

[2] Note that this assumes a deterministic environment. We discuss this limitation and possible stochastic extensions in Appendix F.

by better capturing progress signals than Q-values, which tend to exploit known strategies (Setlur et al., 2025; Kazemnejad et al., 2025).

Multi-path Advantage Reflection (MAR) Inspired by TRPO (Schulman et al., 2015), which computes robust advantage in sparse-reward setting over multiple rollouts from the same state, we propose Multi-path Advantage Reflection to compare multiple trajectories from the same starting state, to produce pseudo-dense advantage signals from sparse environmental feedback. This effectively densifies the reward signal by inferring intermediate advantages at key state-action pairs, providing rich guidance for exploration where environmental rewards are insufficient.

Given a state s selected by the global world model, we perform iterative exploration by sampling n trajectories sequentially: after each trajectory τ_i, we perform MAR to extract learnings that inform the next trajectory τ_{i+1}, in the form of world representation W_{local}. This creates a sequence $\mathcal{T}_s = \{\tau_1, \tau_2, ..., \tau_n\}$ where each trajectory benefits from insights gained from previous attempts.

Require: Selected state s_{next}, Frontier $\mathcal{F}$, Exploration count n
Ensure: Trajectory set $\mathcal{T}_s$

$\mathcal{T}_s \leftarrow \emptyset$
$W_{local} \leftarrow \emptyset$
for $i = 1$ to n **do**
  $\tau_i \leftarrow \pi_{explore}(s_{next}, W_{local}, \mathcal{T}_s, \mathcal{F})$
  $\mathcal{T}_s \leftarrow \mathcal{T}_s \cup \{\tau_i\}$
  $W_{local} \leftarrow MAR(\mathcal{T}_s, \mathcal{F})$
  where $W_{local} = \{(s_1^*, A_{s_1^*}), ..., (s_k^*, A_{s_k^*})\}$
end for
return $\mathcal{T}_s$

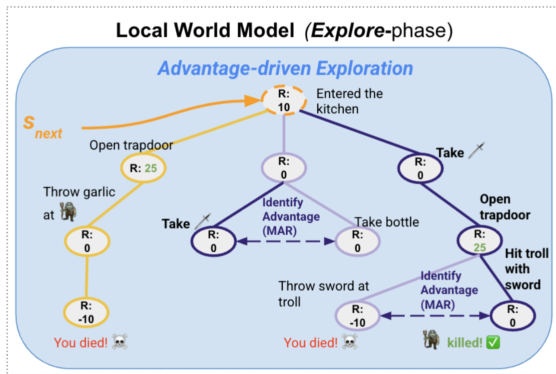

Figure 2: (a) Explore procedure in GLoW, (b) Illustration of exploration with Local World Model

Semantic Advantage Representation Concretely, MAR is an LLM operation taking the local exploration trajectories $\mathcal{T}_s$ and frontier trajectories $\mathcal{F}$ as inputs, and generating the structured textual output $W_{\text{local}} = \{(s_1^*, A_{s_1^*}), ..., (s_k^*, A_{s_k^*})\}$, where $s_1^*, ..., s_k^*$ are critical states (typically 2-4) and each $A_{s_i^*}$ encodes *semantic advantages*. MAR features two design principles for enhancing the accuracy of semantic advantage inference: 1) Multi-trajectory comparison enables LLM reasoning to aggregate over divergent outcomes revealing good/bad actions, or consistent patterns confirming reliable strategies, while focusing analysis on critical states where these signals are most informative. 2) The frontier trajectories (representing the best outcomes achieved so far) provide a stable reference point, grounding the LLM's evaluation of whether new trajectories constitute meaningful progress. This implements a functional role analogous to a value baseline through context-based reasoning rather than numerical subtraction.[3]

Unlike scalar advantages $A(s, a)$, these semantic advantages capture progress signals not expressed by sparse rewards, while serving an analogous functional role by guiding exploration policy through W_{local}. We provide a full example of W_{local} generated for Zork1 in Appendix E.2.

Exploration Policy The local world model enhances the exploration phase by guiding a policy defined by an LLM agent, as:

$$\pi_{\text{explore}}(a|s_t, h_t) = \text{Agent}_{\text{LLM}}(s_t, h_t, W_{\text{local}}, T_s, \mathcal{F}) \quad (3)$$

where h_t is the current trajectory history, T_s contains previous trajectories in the same exploration phase, and the policy leverages both learned advantages from W_{local} and successful strategies from frontier $\mathcal{F}$. Fig. 2 illustrates exploration in GLoW with the local world model. Consider a trajectory (gold) that reached the cellar but failed at the troll bottleneck without the sword. After analysis by the global world model (Fig. 1), which identifies high v' at the cellar state, this state becomes s_{next} (orange root, Fig. 2). The local world model drives multiple exploration attempts (purple paths),

[3] In Appendix A, we provide theoretical motivation showing how multi-trajectory comparison with a stable reference can reduce variance in numerical advantage estimation. MAR applies these principles through LLM reasoning rather than explicit numerical computation.

where MAR identifies advantages for "taking sword" despite no immediate reward. This advantage learning guides successful exploration through the troll bottleneck (rightmost path). Finally, to address Jericho's exponential action space, we implement a hybrid approach combining free generation with soft constraints given by valid actions from the Jericho environment (further details are provided in Appendix D.4).

4 RESULTS

We evaluate GLoW on the Jericho benchmark suite. We next present baselines (Sec. 4.1), experimental setup (Sec. 4.2), main results demonstrating the effectiveness of GLoW (Sec. 4.3), and ablation studies (Sec. 4.4) isolating each module contribution. Lastly, we provide detailed analysis of exploration dynamics in (Sec. 4.5).

4.1 BASELINES

We perform comprehensive comparison against baselines spanning RL-based, MCTS-based, and LLM-based approaches. Furthermore, we compare with specialized methods for hard-exploration problems in each type of baseline. All methods assume access to valid actions from Jericho.

RL-Based Methods: DRRN (He et al., 2016) is a value-based RL approach for choice-based games, learning Q-values for valid actions using GRU encoders and decoders trained via TD loss. **KG-A2C** (Ammanabrolu & Hausknecht, 2020) is a on-policy RL agent that adapts Advantage Actor Critic (A2C) (Mnih et al., 2016), augmented by a dynamic knowledge graph as a state representation that is learned during exploration. Similar to DRRN, **RC-DQN** (Guo et al., 2020) is a DQN-based agent (Mnih et al., 2015), but leverages object-centric neural reading comprehension architectures (Seo et al., 2017) for computing Q-values from observations. **eXploit-Then-eXplore (XTX)** (Tuyls et al., 2022) is the current state-of-the-art method in Jericho, implementing Go-Explore with imitation learning on promising trajectories for state selection, and DQN with intrinsic curiosity reward for exploration. RL-based methods rely on million-scale interaction data to learn, leveraging parallel environments for training, with the exception of RC-DQN which leverages 100,000 interactions.

MCTS-Based Methods: Monte Carlo Tree Search is widely adopted for large sequential decision-making problems (Browne et al., 2012; Silver et al., 2016), which explores effectively by combining random sampling and tree search. **MC-LAVE** (Jang et al., 2021) combines MCTS with language-driven exploration, concentrating search effort on promising actions identified based on value estimates from semantically similar past actions. **MC-DML** (Shi et al., 2025) enhances MCTS by incorporating LLMs as action priors in the PUCT algorithm (Silver et al., 2016), which balances exploration and exploitation during tree search. The LLM is equipped with a cross-trial memory mechanism, allowing it learn from past experiences such as death in Zork1. Both methods require around 400,000 environment interactions to build comprehensive search trees.

LLM-Based Methods: ReAct (Yao et al., 2023) is the widely adopted standard LLM agent approach interleaving reasoning and acting. **Reflexion** (Shinn et al., 2023) is a multi-episode approach building on ReAct, incorporating self-reflection on each episode to guide future episodes. **In-context Reinforcement Learning** (ICRL) (Song et al., 2025) is another multi-episode approach leveraging in-context reinforcement learning, using cumulative history of past trajectories and rewards as context for future episodes. **Intelligent Go-Explore (IGE)** (Lu et al., 2025) implements Go-Explore with LLMs, leveraging LLM-based state selection from a state archive, combined with ReAct-based exploration. As LLM-based baseline methods were not originally applied on Jericho, we re-implement them for Jericho using the action generation approach with valid action soft-constraint described in Sec. 3.2. All LLM-based approaches use 1,000 interactions to balance performance and API cost. We provide details of LLM API usage and cost in Appendix C.1.

4.2 EXPERIMENTAL SETUP

Implementation Details Each method is evaluated over 3 random seeds, reporting mean and standard deviation of maximum achieved scores. ReAct performs 20 independent 50-step episodes. Reflexion performs 20 trials of 50-step episodes, incorporating sliding-window memories from up to 10 previous attempts. Likewise, ICRL includes a sliding window of 10 previous trajectories as in-context examples. IGE and GLoW adaptively alternate between state selection and 50-step exploration episodes within the total 1,000 step budget. We found 50 steps to be sufficient for baseline agents, as they typically plateau early on puzzles or repetitive action loops before reaching this limit. We use temperature 0.5 for all methods except IGE, which uses 0.3 following Lu et al. (2025). For GLoW hyperparameters, n=3 exploration trajectories and k=5 trajectory frontier size is used.

Evaluation We evaluate on 10 games from the Jericho benchmark (Hausknecht et al., 2019), spanning different difficulty levels. Following the benchmark's categorization, we test on *possible games* (Pentari, Detective, Temple, Ztuu) featuring moderate puzzles and frequent rewards, *difficult games* (Zork1, Zork3, Deephome, Ludicorp) requiring more complex inventory management, puzzle-solving and navigation, and *extreme games* (Enchanter) involving non-standard actions and spell mechanics. We use the standard Jericho interface providing textual observations and access to valid actions at each step. Unlike some prior work, we do not augment observations with explicit "look" or "inventory" commands, instead allowing agents to learn these through play.

Games	*RL-based*				*MCTS-based*		*LLM-based*				
	DRRN	**KG-A2C**	**RC-DQN**	**XTX**	**MC-LAVE**	**MC-DML**	**ReAct**	**Reflexion**	**ICRL**	**IGE**	**GLoW (Ours)**
Steps	1,000,000	1,600,000	100,000	800,000	∼400,000	∼400,000	1000	1000	1000	1000	1000
Enchanter	20	12.1	20	<u>52.0</u>	–	20±0.0	46.7±9.4	48.3±9.4	43.3±8.5	50.0±7.1	**61.7**±20.1
Zork1	32.6	40.2±0.4	38.8	**103.4**±10.9	45.2	48.66±1.89	48.3±4.7	48.0±5.0	51.7±4.7	44.3±0.5	<u>73.0±4.5</u>
Zork3	0.5	0.0	2.83	<u>4.2±0.1</u>	–	3±0.0	3.0±0.0	2.7±0.5	3.0±0.0	3.7±0.9	**4.3**±0.9
Deephome	1	20±2.1	1	**77.7**±2.1	35	67±1.41	11.0±4.2	22.0±1.6	24.0±5.7	71.3±4.9	<u>75.0±8.7</u>
Ludicorp	13.8	19.8±1.0	17	**78.8**	22.8	19.67±1.7	19.7±0.9	21.7±1.2	32.0±7.1	28.3±11.3	<u>73.7±11.0</u>
Balances	10	10	10	**24**	10	10±0.0	10±0.0	10±0.0	11.7±2.4	10.0±0.0	<u>16.7±2.4</u>
Pentari	27.2	44±0.9	43.8	49.6	<u>68</u>	**70**±0.0	30.0±0.0	30.0±0.0	26.7±4.7	30.0±0.0	30.0±0.0
Detective	197.8	<u>338±3.4</u>	291.3	312.2	330	**346.67**±9.43	113.3±4.7	166.7±20.5	233.3±47.8	316.7±4.7	310.0±8.2
Temple	7.4	8	8	–	8±0.0	8±0.0	8.7±0.9	8.7±0.9	8±0.0	**13.7**±0.9	<u>13.0±0.0</u>
Ztuu	21.6	5±0.0	–	–	7	<u>23.67±1.9</u>	18.7±2.4	18.3±2.6	16.7±4.1	15.0±9.1	**29.3**±4.0

Table 1: Comparison of RL-based, MCTS-based, and LLM-based methods on Jericho benchmark games. We report mean ± standard deviation over 3 runs following prior works (Tuyls et al. (2022); Shi et al. (2025)). **Bold** indicates best overall performance, and <u>underline</u> indicates second-best. Steps shows total environment interactions. The color of game name indicates original game difficulty categories from Hausknecht et al. (2019): extreme, difficult, and possible. GLoW achieves state-of-the-art among LLM-based approaches in 7/10 games, and is overall best among all compared approaches in 3/10, second-best in 5/10.

4.3 Main Results

We report our main results in Table 1. GLoW achieves a new state-of-the-art performance among LLM approaches across 7 out of 10 games. On Zork1, a canonical game of the Jericho suite, our method reaches a score of 73.0, a significant improvement over the next best LLM method (ICRL at 51.7), and surpassing all compared approaches (with the exception of XTX), including RL and MCTS baselines that use orders of magnitude more interactions. We observe the same strong improvements over the closest LLM method in Ludicorp (73.7 vs. 32.0 for ICRL), Enchanter (61.7 vs. 50.0 for IGE), Ztuu (29.3 vs. 18.7 for ReAct), and Balances (16.7 vs. 11.7 for ICRL).

Notably, our implementation of baselines with hybrid action generation approach shows surprisingly strong performance, whereas prior works reported near-zero scores for LLM agents on Jericho (Shi et al., 2025; Cui et al., 2025; Phan et al., 2025). Our implementation enables ReAct, Reflexion and ICRL to reach 48.3, 48.0, 51.7 on Zork1, respectively, and similarly on par with RL baselines such as KG-A2C and RC-DQN across the board. While this reveals the sample efficiency of LLM agents, these baselines still fall far short of more advanced exploration methods such as XTX and MC-DML, demonstrating the necessity of effective exploration for LLM agents.

Next we compare GLoW against advanced exploration approaches. First, comparing with IGE which is the most directly comparable to ours as an LLM-based Go-Explore method, GLoW substantially outperforms with better performance on 8 out of 10 games. GLoW also achieves competitive performance with state-of-the-art RL and MCTS methods, XTX and MC-DML. We nearly match the overall state-of-the-art XTX, which uses 800× more interactions, on both Deephome (75.0

vs. 77.7) and Ludicorp (73.7 vs. 78.8), and notably surpass it on Enchanter (61.7 vs. 52.0). It also outperforms MC-DML, which employs extensive MCTS-based exploration around 400× more interactions, on most games including Zork1 (73.0 vs. 48.66), Deephome (75.0 vs. 67.0), and Ludicorp (73.7 vs. 19.67). These results demonstrate that our dual-scale approach combining global world models for value-based state selection, with advantage learning for exploration, enables significant performance gains in LLM agents, competitive with sample-intensive RL approaches.

4.4 Ablation Study

To validate the contribution of each component of GLoW, we perform systematic ablations and report the results in Table 2.

Effectiveness of Local World Model We first analyze the efficacy of our local world model by ablating MAR. We replace MAR by Reflexion, which performs the same multi-path exploration but does not leverage our proposed advantage learning, instead performing single-trajectory reflection on the latest trajectory. The results show that the performance drops significantly across most games, demonstrating that MAR's advantage-based formulation more effectively leverages multi-trajectory information than Reflexion, improving exploration under sparse rewards.

Effectiveness of Global World Model Next, we analyze the effectiveness of the global world model, which consists of the frontier of high-value trajectories, and the LLM-based value analysis and alignment state selection. We first ablate the LLM-based value analysis W_{global}, leveraging the raw frontier trajectories for state selection. The negative performance impact shows that, using LLM to reason across the frontier trajectories to infer potential value is indeed effective. Next, we ablate the trajectory frontier $\mathcal{F}$ altogether, such that it is not used for state selection or leveraged by the exploration policy. This causes further decrease in performance, confirming the contribution of the trajectory frontier in both phases.

Synergy of LWM and GWM Finally, we ablate all the above components together. The resultant model is similar to IGE, with multi-path Reflexion for exploration. The results show that simply adding multi-path reflection does not lead to a clear improvement over IGE, indicating that the overall performance of GLoW comes from the complementary synergy of its components.

Ablation Variants	**Zork1**	**Zork3**	**Enchanter**	**Deephome**	**Ludicorp**	**Balances**
GLoW (Full)	**73.0**±4.5	**4.3**±0.9	**61.7**±20.1	**75.0**±8.7	**73.7**±11.0	**16.7**±2.4
✗ [Local WM] Multi-path Advantage Reflection (MAR)	70.0±13.6	4.3±0.5	51.7±9.4	56.7±21.7	54.7±22.4	11.7±2.4
✗ [Global WM] State selection with W_{global}	62.0±15.6	4.3±0.9	60.0±10.8	61.3±26.0	63.3±14.7	13.3±2.4
✗ [Global WM] Trajectory frontier $\mathcal{F}$	61.7±1.9	4.0±0.8	53.3±10.3	57.7±23.3	63.3±12.3	11.7±2.4
✗ All above	51.3±5.2	4.3±0.9	51.7±9.4	56.0±21.2	22.0±0.8	10.0±0.0
Standard IGE	44.3±0.5	3.7±0.9	50.0±7.1	71.3±4.9	28.3±11.3	10.0±0.0

Table 2: Ablation study on GLoW components. We evaluate the contribution of: (1) Local world model through MAR, (2) Global world model for state selection, (3) trajectory frontier $\mathcal{F}$.

4.5 Analysis

Controlling global vs. local focus with n exploration parameter We study the tradeoff between local learning depth and global exploration coverage by varying n, the number of explorations per selected state. Larger n enables MAR to learn from more trajectories, while smaller n increases state selection frequency, helping escape local minima. With budget B=1000 and steps s=50, minimum state selections is $m = \lfloor B/(s \cdot n) \rfloor - 1$. With n=1, MAR is turned off. With n>1, MAR analyzes n-1 local trajectories plus the global frontier trajectories.

Table 3 shows that extreme values of n generally yield suboptimal performance. When n=1, effectively disabling MAR, performance drops significantly on certain games like Ludicorp (34.0 vs 73.7 with n=3). Conversely, Deephome shows consistent improvement with increasing n, suggesting it particularly benefits from deeper local exploration. The results demonstrate that moderate increases in n improve performance across several games, consistent with our theoretical motivation (Appendix A) that MAR should benefit from multi-trajectory comparisons. However, setting n=5 begins to degrade performance, as excessive commitment to individual exploration phases reduces minimum state selection frequency to just 3, increasing susceptibility to local optima. These

Table 3: Controlling the focus on global (less explorations per state but more frequent state selection) vs local learning (more explorations per state). The results demonstrate n=3 exploration from promising states strikes a good balance between the two.

Explorations per State	Max. Steps per Exploration Phase	Min. State Selection	Zork1	Zork3	Enchanter	Deephome	Ludicorp	Balances
1 (no MAR)	50 × 1	19	59.0±5.7	3.7±0.9	58.3±9.4	59.7±22.6	34.0±15.6	13.3±4.7
2 (MAR w/ 1)	50 × 2	9	67.3±8.7	3.7±1.2	55.0±7.1	43.3±26.6	66.0±3.7	11.7±2.4
3 (MAR w/ 2)	50 × 3	5	**73.0±4.5**	**4.3±0.9**	61.7±20.1	75.0±8.7	**73.7±11.0**	**16.7±2.4**
4 (MAR w/ 3)	50 × 4	4	63.0±6.5	**4.3±0.9**	**66.7±10.3**	73.7±4.5	62.0±12.4	16.7±2.4
5 (MAR w/ 4)	50 × 5	3	59.3±13.8	4.0±0.8	46.7±6.2	**76.3±6.8**	53.3±7.0	15.0±0.0

findings indicate that balancing global and local learning is crucial. We select n=3 as our default parameter, as it achieves the best overall performance by providing sufficient trajectories for robust advantage estimation while maintaining adequate state selection frequency to escape local minima.

5 Related Works

Go-Explore-based Methods Go-Explore (Ecoffet et al., 2019) enables effective exploration in sparse-reward environments by decomposing exploration into state selection and exploration IGE (Lu et al., 2025) adapts Go-Explore for LLMs, using LLM-based "promisingness" for state selection and ReAct for exploration. However, IGE's limited exploration and ill-defined selection criteria limit its effectiveness in complex environments like Jericho. Our work addresses these limitations through principled value decomposition for selection, and multi-path advantage learning for exploration.

Agents for Text-based Games RL approaches to Jericho include DRRN (He et al., 2016), KG-A2C (Ammanabrolu & Hausknecht, 2020), and RC-DQN Guo et al. (2020), and the aforementioned XTX, where all are sample-intensive, relying on hundreds of thousands of interactions. MCTS-based methods like MC-LAVE Jang et al. (2021) and MC-DML Shi et al. (2025) leverage tree search but still rely on a similar scale of interactions. We show that LLM agents can achieve comparable performance to RL methods, while requiring orders of magnitude fewer interactions through structured exploration and learning mechanisms.

Learning in LLM Agents Recent works have studied how LLMs can learn from experience. Reflexion (Shinn et al., 2023) enables learning through self-reflection on failed attempts, while in-context reinforcement learning (ICRL) (Song et al., 2025) leverages previous trajectories' history as context. However, these approaches struggle with sparse rewards due to noisy learning signals. Our MAR mechanism addresses this challenge through multi-path advantage-based learning, providing more robust learning signals.

World Models for LLM Agents While traditional world models in model-based RL focus on transition dynamics (Ha & Schmidhuber, 2018; Hafner et al., 2024), recent works adopt an expanded paradigm of world models as mechanisms for implicit representations of task-relevant knowledge Ding et al. (2025); Li et al. (2024). Li et al. (2024) formalize this notion through state abstraction theory Abel (2022), showing that effective LLM agents build goal-oriented abstractions without recovering full dynamics. GLoW 's dual-scale textual world models align with this view, where the global world model extracts value decompositions across global discoveries, while the local world model captures semantic advantage signals for exploration.

6 Conclusion

We introduce GLoW, a dual-scale world model framework to tackle hard-exploration problems. GLoW leverages a global world model that enables principled decomposition of state values, and a local world model that integrates trajectories from the same state as controlled exploration feedbacks. Our approach achieves state-of-the-art performance among LLM methods on the challenging Jericho benchmark, while matching RL-based methods that require 800× more environment interactions. By learning global value patterns across discoveries, and local progress signals from multi-path exploration, GLoW overcomes a key limitation of LLM agents in hard-exploration tasks, demonstrating a sample efficient yet high performance results.

Reproducibility Statement

To ensure reproducibility of our results, we provide comprehensive implementation details in the paper. Algorithm 2 provides the complete pseudocode for GLoW, and hyperparameters are detailed in Section 4.2 (n=3 exploration trajectories, temperature=0.5, k=5 frontier size, 1000 environment steps). All prompts used for the global world model (Appendix D.1), LLM-based state selection (Appendix D.2), MAR (Appendix D.3), and exploration policy (Appendix D.4) are provided in full. Experiments use `GPT-4.1-mini-2025-04-14` as the LLM backbone, reporting results averaged over 3 random seeds with standard deviations. We implement all LLM baselines using the same action generation approach (Section 3.2) for fair comparison. The Jericho benchmark is publicly available, and we use the standard evaluation protocol from Hausknecht et al. (2019). Code implementation will be publicly released upon publication.

References

David Abel. A theory of abstraction in reinforcement learning, 2022. URL https://arxiv.org/abs/2203.00397.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022. URL https://arxiv.org/abs/2204.01691.

Prithviraj Ammanabrolu and Matthew Hausknecht. Graph constrained reinforcement learning for natural language action spaces. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1x6w0EtwH.

Prithviraj Ammanabrolu and Mark Riedl. Modeling worlds in text. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=7FHnnENUG0.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3(null):397–422, March 2003. ISSN 1532-4435.

Ronen I. Brafman and Moshe Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *J. Mach. Learn. Res.*, 3(null):213–231, March 2003. ISSN 1532-4435. doi: 10.1162/153244303765208377. URL https://doi.org/10.1162/153244303765208377.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012. doi: 10.1109/TCIAIG.2012.2186810.

Christopher Cui, Xingdi Yuan, Ziang Xiao, Prithviraj Ammanabrolu, and Marc-Alexandre Côté. Tales: Text-adventure learning environment suite. *arXiv preprint arXiv:2504.14128*, 2025. URL https://arxiv.org/abs/2504.14128.

Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, Fengli Xu, and Yong Li. Understanding world or predicting future? a comprehensive survey of world models. *ACM Comput. Surv.*, 58(3), September 2025. ISSN 0360-0300. doi: 10.1145/3746449. URL https://doi.org/10.1145/3746449.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *CoRR*, abs/1901.10995, 2019. URL http://arxiv.org/abs/1901.10995.

Xiaoxiao Guo, Mo Yu, Yupeng Gao, Chuang Gan, Murray Campbell, and Shiyu Chang. Interactive fiction game playing as multi-paragraph reading comprehension with reinforcement learning. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7755–7765, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.624. URL https://aclanthology.org/2020.emnlp-main.624/.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/2de5d16682c3c35007e4e92982f1a2ba-Paper.pdf.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. Interactive fiction games: A colossal adventure. In *AAAI 2020*, October 2019. URL https://www.microsoft.com/en-us/research/publication/interactive-fiction-games-a-colossal-adventure/. ArXiv.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. Deep reinforcement learning with a natural language action space. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1621–1630, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1153. URL https://aclanthology.org/P16-1153/.

Youngsoo Jang, Seokin Seo, Jongmin Lee, and Kee-Eung Kim. Monte-carlo planning and learning with language action value estimates. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=7_G8JySGecm.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Myx2kJFzAn.

Zichao Li, Yanshuai Cao, and Jackie CK Cheung. Do LLMs build world representations? probing through the lens of state abstraction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=lzfzjYuWgY.

Cong Lu, Shengran Hu, and Jeff Clune. Intelligent go-explore: Standing on the shoulders of giant foundation models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=apErWGzCAA.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 00280836. URL http://dx.doi.org/10.1038/nature14236.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/mniha16.html.

Long Phan, Mantas Mazeika, Andy Zou, and Dan Hendrycks. Textquests: How good are llms at text-based video games?, 2025. URL https://arxiv.org/abs/2507.23701.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1889–1897, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HJ0UKP9ge.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for LLM reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=A6Y7AqlzLW.

Zijing Shi, Meng Fang, and Ling Chen. Monte carlo planning with large language model for text-based game agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=r1KcapkzCt.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 8634–8652. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1b44b878bb782e6954cd888628510e90-Paper-Conference.pdf.

David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016. URL http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html.

Kefan Song, Amir Moeini, Peng Wang, Lei Gong, Rohan Chandra, Yanjun Qi, and Shangtong Zhang. Reward is enough: Llms are in-context reinforcement learners, 2025. URL https://arxiv.org/abs/2506.06303.

Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. Cognitive architectures for language agents. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=1i6ZCvflQJ. Survey Certification.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf.

Chen Feng Tsai, Xiaochen Zhou, Sierra S. Liu, Jing Li, Mo Yu, and Hongyuan Mei. Can large language models play text games well? current state-of-the-art and open questions, 2025. URL https://arxiv.org/abs/2304.02868.

Jens Tuyls, Shunyu Yao, Sham M. Kakade, and Karthik R Narasimhan. Multi-stage episodic control for strategic exploration in text games. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Ek7PSN7Y77z.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), March 2024. ISSN 2095-2236. doi: 10.1007/s11704-024-40231-1. URL http://dx.doi.org/10.1007/s11704-024-40231-1.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=mXpq6ut8J3.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. Agentoccam: A simple yet strong baseline for LLM-based web agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=oWdzUpOlkX.

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

A THEORETICAL MOTIVATION FOR MULTI-PATH ADVANTAGE REFLECTION

A.1 VARIANCE REDUCTION THROUGH MULTI-TRAJECTORY COMPARISON

The design of MAR draws motivation from classical results on variance reduction in advantage estimation. We present this theoretical background that inspired our approach, noting that MAR implements these principles through LLM reasoning rather than explicit numerical computation.

Proposition 1. *Consider n trajectories $\{\tau_1, ..., \tau_n\}$ starting from state s. For any state s^* visited by $m \leq n$ of these trajectories, an advantage estimate computed by averaging over m trajectory outcomes has variance reduced by factor $1/m$ compared to a single-trajectory estimate, assuming bounded variance across trajectories:*

$$\text{Var}[\hat{A}_{\text{multi}}(s^*)] \leq \frac{\text{Var}[\hat{A}_{\text{single}}(s^*)]}{m}$$

Proof. For a trajectory j passing through state s^* and taking action a_j, let $R_j(s^*, a_j)$ denote the random variable representing the sum of future rewards from s^* onward. This provides an unbiased estimate of the true $Q(s^*, a_j)$.

The single-trajectory advantage estimate for action a is:

$$\hat{A}_{\text{single}}(s^*, a) = R_j(s^*, a) - \hat{V}(s^*)$$

where $\hat{V}(s^*)$ is an estimate of the state value. This estimate has high variance because it relies on a single sample: $\text{Var}[\hat{A}_{\text{single}}(s^*, a)] = \text{Var}[R_j(s^*, a)]$ when $\hat{V}(s^*)$ is held constant.

Now consider a multi-trajectory approach. From the m trajectories passing through s^*, let m_a denote the number of trajectories taking action a. Averaging outcomes yields an improved Q-value estimate:

$$\hat{Q}_{\text{multi}}(s^*, a) = \frac{1}{m_a} \sum_{j:a_j=a} R_j(s^*, a)$$

Using basic properties of variance for independent random variables with equal variance σ_a^2:

$$\text{Var}[\hat{Q}_{\text{multi}}(s^*, a)] = \text{Var}\left[\frac{1}{m_a} \sum_{j:a_j=a} R_j\right] = \frac{1}{m_a^2} \cdot m_a \cdot \sigma_a^2 = \frac{\sigma_a^2}{m_a}$$

This shows variance reduction by factor m_a for the Q-estimate. For the baseline, incorporating a stable reference (such as outcomes from previously successful trajectories) rather than a fluctuating estimate further reduces variance. Under the assumption that this stable baseline has low variance relative to the Q-estimate, the variance of the advantage estimate is dominated by the Q-component:

$$\text{Var}[\hat{A}_{\text{multi}}(s^*, a)] \approx \text{Var}[\hat{Q}_{\text{multi}}(s^*, a)] = \frac{\sigma_a^2}{m_a} \leq \frac{\sigma_a^2}{1} = \text{Var}[\hat{A}_{\text{single}}(s^*, a)]$$

For any action with $m_a \geq 1$ samples, we achieve variance reduction by a factor of m_a. □

A.2 Connection to MAR

The proposition above motivates why multi-trajectory comparison with a stable baseline can reduce variance in advantage estimation. While not directly approximating the numerical quantity, MAR operationalizes these principles through LLM reasoning rather than explicit numerical computation:

Multi-trajectory aggregation. Rather than computing the average $\frac{1}{m_a}\sum_j R_j$, MAR prompts the LLM to reason across multiple trajectories from the same starting state, identifying consistent patterns and divergent outcomes. This achieves a benefit analogous to variance reduction through averaging, as the LLM can implicitly weigh evidence from multiple outcomes when inferring which actions led to better progress.

Stable baseline via frontier. The frontier $\mathcal{F}$ serves an analogous role to the stable $\hat{V}(s^*)$ in the proposition. Like target networks in DQN (Mnih et al., 2015) that update periodically to provide stable targets, $\mathcal{F}$ updates only when superior trajectories are discovered, providing a consistent reference point for the LLM's evaluation of whether new trajectories constitute meaningful progress.

B Algorithms

We provide the detailed overview of Go-Explore-based algorithms in Alg. 3, and the full algorithm of GLoW in Alg. 2.

C Contamination Check

Table 4: Data contamination analysis: LLM accuracy (%) on navigation questions without seeing gameplay.

Game	# Questions	Accuracy (%)
Zork1	230	10.9
Zork3	194	8.2
Enchanter	239	9.2
Detective	66	9.1
Balances	54	1.9
Library	26	15.4
Pentari	70	1.4
Deephome	288	17.0
Temple	92	12.0
Ludicorp	320	19.7
Ztuu	71	9.9

To assess whether large language models have prior knowledge of Jericho games, we conducted a data contamination analysis following the methodology of Tsai et al. (2025). We evaluate contamination by testing whether models can navigate between locations without being shown any gameplay. Specifically, we: (1) collect a walkthrough trajectory by executing up to 300 steps from each game's built-in Jericho walkthrough actions, (2) build a graph of locations and transitions from this walkthrough, (3) generate navigation questions asking for paths between observed locations, and (4) query the model with these questions without providing any context. Navigation questions take the form: "In [GAME], what steps would you take to go to [LOCATION B] from [LOCATION

Algorithm 2 GLoW: **G**lobal-**Lo**cal **W**orld Models

procedure GLOW(s_0, n_{iter}, n, k)
  $\mathcal{F} \leftarrow \emptyset$ ▷ Initialize frontier
  $\mathcal{A} \leftarrow \{(s_0, 0)\}$ ▷ Initialize state archive
  for $i = 1$ to n_{iter} **do**
    $s_{\text{next}} \leftarrow$ SELECTSTATE($\mathcal{F}$, $\mathcal{A}$)
    $\mathcal{T} \leftarrow$ EXPLORE(s_{next}, $\mathcal{F}$, n)
    UPDATEARCHIVE($\mathcal{T}$, $\mathcal{F}$, $\mathcal{A}$, k)
  end for
  return $\arg\max_{\tau \in \mathcal{F}} v(\tau)$
end procedure

procedure SELECTSTATE($\mathcal{F}$, $\mathcal{A}$)
  $W_{\text{global}} \leftarrow g_{\text{LLM}}(\mathcal{F})$
  $s_{\text{next}} \leftarrow \arg\max_{s \in \mathcal{A}} \text{align}_{\text{LLM}}(s, W_{\text{global}})$ ▷ Select state based on decomposed value
  return s_{next}
end procedure

procedure EXPLORE(s, $\mathcal{F}$, n)
  $\mathcal{T} \leftarrow \emptyset$ ▷ Initialize trajectory set for current exploration phase
  $W_{\text{local}} \leftarrow \emptyset$
  for $j = 1$ to n **do**
    $\tau_j \leftarrow \pi_{\text{explore}}(s, W_{\text{local}}, \mathcal{T}, \mathcal{F})$ ▷ Rollout full trajectory from s
    $\mathcal{T} \leftarrow \mathcal{T} \cup \{\tau_j\}$
    $W_{\text{local}} \leftarrow$ MAR($\mathcal{T}$, $\mathcal{F}$)
  end for
  return $\mathcal{T}$
end procedure

procedure MAR($\mathcal{T}$, $\mathcal{F}$)
  $W_{\text{local}} \leftarrow f_{\text{LLM}}(\mathcal{T}, \mathcal{F})$ ▷ Extract semantic advantages at key states
  return W_{local}
end procedure

procedure UPDATEARCHIVES($\mathcal{T}$, $\mathcal{F}$, $\mathcal{A}$, k)
  for $\tau \in \mathcal{T}$ **do**
    $\mathcal{F} \leftarrow \text{top-}k(\mathcal{F} \cup \{\tau\}, v)$ ▷ Update the trajectory frontier
    for $s' \in \tau$ **do**
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{(s', score(s'))\}$ ▷ Add states to state archive
    end for
  end for
end procedure

A]?" We evaluate responses using strict pattern matching with word boundaries, requiring the exact sequence of navigation commands to appear consecutively in the model's response.

Table 4 shows results of contamination checks for `GPT-4.1-mini` across 11 Jericho games. We observe minimal contamination, with all games showing below 20% accuracy. Most games (8 out of 11) show less than 10% accuracy, consistent with random guessing or generic text adventure knowledge. The slightly higher accuracies for Ludicorp (19.7%), Deephome (17.0%), and Library (15.4%) likely reflect the model providing common navigation commands (e.g., "go south") that occasionally match by chance. Even famous games like Zork1 (10.9%) show accuracy near chance level, while less-known games like Balances (1.9%) and Pentari (1.4%) show essentially no prior knowledge. These low accuracy rates, combined with the model's generic responses that lack game-specific details, indicate that our experimental results reflect genuine exploration and reasoning capabilities rather than memorized solutions.

C.1 LLM API Cost

We use `gpt-4.1-mini-2025-04-14` for all LLM components ($0.40/$1.60 per million input/output tokens). Per-run costs of all LLM-based approaches with 1,000 environment steps range

Algorithm 3 Go-Explore-based Algorithms

procedure GO-EXPLORE-FAMILY(s_0, n_{iter})
  $\mathcal{A} \leftarrow \{(s_0, 0)\}$ ▷ Archive of (state, score)
  $\mathcal{F} \leftarrow \emptyset$ ▷ Trajectory Frontier
  for $i = 1$ to n_{iter} **do**
    — Go Phase (State Selection) —
    Go-Explore A: $s_{next} \sim \text{Uniform}(\mathcal{A})$ ▷ Random sampling
    Go-Explore B: $s_{next} \sim P(s) \propto \frac{1}{\text{visits}(s)^\alpha}$ ▷ Novelty
    Go-Explore C: $s_{next} \sim P(s) \propto \text{domain}(s)$ ▷ Domain heuristics
    XTX: $s_{next} \leftarrow \text{ImitationLearning}(\mathcal{T})$ ▷ Imitation learning
    IGE: $s_{next} \leftarrow \text{LLM.SelectPromising}(\mathcal{A})$ ▷ Ill-defined promising-ness
    GLoW: $W_{\text{global}} \leftarrow g_{\text{LLM}}(\mathcal{F})$ ▷ Principled value decomposition (Sec. 3.1)
      $s_{next} \leftarrow \text{align}_{\text{LLM}}(\mathcal{A}, W_{\text{global}})$

    — Explore Phase —
    Go-Explore: $\tau \leftarrow \text{RandomActions}(s_{next})$ ▷ No learning
    XTX: $\tau \leftarrow \text{DQN}(s_{next})$ ▷ DQN with curiosity reward
    IGE: $\tau \leftarrow \text{ReAct}(s_{next})$ ▷ Standard LLM agent
    GLoW:
    for $j = 1$ to n **do** ▷ LLM agent with advantage-driven exploration (Sec. 3.2)
      $\tau_j \leftarrow \pi_{explore}(s_{next}, W_{\text{local}}, \{\tau_1, \ldots, \tau_{j-1}\}, \mathcal{F})$
      $W_{\text{local}} \leftarrow \text{MAR}(\{\tau_1, \ldots, \tau_j\}, \mathcal{F})$
    end for
    $\mathcal{F} \leftarrow \text{top-}k(\mathcal{F} \cup \{\tau_1, \ldots, \tau_n\}, v)$ ▷ Update trajectory frontier

    — Archive Update —
    for each state s' in τ **do**
      if IsNotRedundant(s', $\mathcal{A}$) **then** ▷ Domain-specific novelty
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{s'\}$
      end if
    end for
  end for
end procedure

from \$1 to \$7, maintaining practicality for research iteration. For experiments with stronger LLMs, we use `gpt-4.1-2025-04-14` for all LLM components (\$2.00/\$8.00 per million input/output tokens). Per-run costs range from \$7.5 to \$45.00, demonstrating that while stronger models increase costs by 5-6×, the relative efficiency of our approach remains consistent with $\sim$40% fewer tokens than ICRL, while achieving superior performance.

Table 5: Comparison of LLM API costs with GPT-4.1-mini. We report average the token consumption and costs across 6 games (Zork1, Zork3, Deephome, Ludicorp, Detective, Temple).

Method	Input tokens	Output tokens	Total tokens	Avg. cost/run
ICRL	17.6M	43.1K	17.6M	\$7.10
GLoW	10.9M	115.1K	11.0M	\$4.54
Reflexion	7.2M	50.6K	7.3M	\$2.98
IGE	3.9M	44.1K	3.9M	\$1.61
ReAct	2.6M	35.5K	2.6M	\$1.08

Table 6: Comparison of LLM API costs with GPT-4.1. We report the average token consumption and costs across the same 6 games.

Method	Input tokens	Output tokens	Total tokens	Avg. cost/run
ICRL	22.4M	55.1K	22.5M	\$45.24
GLoW	13.3M	126.2K	13.4M	\$27.58
Reflexion	9.9M	62.5K	10.0M	\$20.31
IGE	4.9M	64.2K	5.0M	\$10.33
ReAct	3.5M	58.2K	3.5M	\$7.45

D PROMPTS

We present the full prompts used in GLoW. Our prompts rely solely on simple instructions and structured output formats without requiring few-shot exemplars, enabling the method to generalize across diverse game scenarios.

D.1 FRONTIER TRAJECTORY ANALYSIS

Analysis (g_{LLM}) Prompt

```
Analyze these successful game trajectories to identify patterns and
strategy:

{For each trajectory in F:}
Trajectory N (Peak:  X, Final:  Y):
  [score] action -> observation (reward:  +/-N if non-zero)
  [score] action -> observation
  ...

Based on these trajectories, provide a strategic analysis:

1.  FRONTIER & EXPLORATION STATUS:
  - What areas/locations have been successfully reached?
  - What remains unexplored or inaccessible?

2.  GAME CHECKPOINTS & PROGRESS:
  - What are the key milestones/checkpoints identified?
  - What items or abilities unlock new areas?
  - What phase of the game are we in?

3.  BOTTLENECKS & CHALLENGES:
  - Where do trajectories commonly get stuck?
  - What obstacles block further progress?
  - What resources or knowledge are we missing?

4.  REWARD STRUCTURE:
  - When and how are points earned?
  - What actions yield the highest rewards?
  - Are there patterns to the scoring?

5.  NEXT INVESTIGATION GOALS:
  - What specific objectives should we pursue?
  - Which unexplored areas are most promising?
  - What items or states do we need to reach?

Provide a concise strategic summary focusing on actionable
insights.
```

D.2 STATE SELECTION

State Selection ($\text{align}_{\text{LLM}}$) Prompt

```
=== STRATEGIC GAME ANALYSIS ===
{Analysis of frontier trajectories W_global}
==================================================

Based on the above analysis, select the state from the archive
that:
- Best aligns with the identified investigation goals
- Can help overcome identified bottlenecks
```

```
- Explores promising frontiers
- Has potential for high rewards based on patterns

Current state archive:

0: [Score: X, Steps: Y, Visits: Z]
  Observation: {state observation}
  Inventory: {state inventory}

1: [Score: X, Steps: Y, Visits: Z]
  Observation: {state observation}
  Inventory: {state inventory}

...

Choose state index (0-N).
Respond in JSON format:
{
  "thought": "Your reasoning about which state best aligns with the
strategic goals",
  "index": <number>
}
```

D.3 Multi-path Advantage Reflection (MAR)

The MAR prompt generates W_{local} as described in Section 3.2, identifying critical decision points and their associated advantages from multiple exploration trajectories. The prompt incorporates three inputs: (1) the global trajectory frontier containing highest-value trajectories that serve as value baselines, (2) local exploration attempts from the current phase showing different outcomes from the same starting state, and (3) previous W_{local} outputs when available, enabling cumulative learning within the exploration phase.

By comparing outcomes across these trajectory sources, MAR produces $W_{\text{local}} = \{(s_i^*, A_{s_i^*})\}_{i=1}^k$, identifying where specific actions provide clear advantages. This semantic representation captures causal relationships (e.g., "taking the lamp enables combat in darkness") rather than strictly scalar values, enabling the exploration policy to leverage both statistical patterns from trajectory comparison and LLM reasoning about game mechanics at critical states.

W_{local} Generation Prompt (MAR)

```
Review these exploration attempts and identify KEY STATE
ADVANTAGES:

{Previous W_local from earlier iterations, if any}

{Global frontier trajectories F}

{Local exploration trajectories from state s}

Analyze all trajectories and identify ADVANTAGES at KEY STATES:

For each important location/state observed across ALL attempts,
list:
- STATE: [description of state/location]
- ADVANTAGES discovered:
  • [specific action] → [specific benefit/outcome] (score impact if
clear)
  • [what to avoid] → [consequence] (score impact if clear)
  • [optimal sequence] → [why it's better]
```

```
Example format:
STATE: At the house entrance with lamp
- ADVANTAGES:
  • "go east" → finds sword (enabled +15 points later)
  • "open mailbox first" → gets crucial map (+5 immediate)
  • avoid "go upstairs" early → wastes moves in empty attic (-7
overall)

Focus on:
1.  States that appear across multiple attempts (to see different
outcomes)
2.  Critical decision points where scores diverged significantly
3.  Action sequences that consistently led to success or failure
4.  Items or information that enabled later progress

Provide 2-4 KEY STATES with their discovered advantages.
Be specific about actions, items, and locations from the actual
game.
```

D.4 Exploration Policy in GLoW

The prompt for the exploration policy in GLoW is shown below. We instruct the LLM to use a JSON output format with reasoning and action fields.

Hybrid Action Space While previous works use either constrained selection from valid actions in RL agents (Hausknecht et al., 2019; Ammanabrolu & Hausknecht, 2020; Tuyls et al., 2022) or pure free-form generation in LLM agents such as ReAct, we provide the valid actions to the LLM as a soft constraint, while still allowing free-form generation. This avoids failure modes of both approaches, where constrained selection can harm action diversity, while pure generation can produce many invalid actions. As shown in Section 4.2, this hybrid approach, which we use consistently across both GLoW as well as all LLM baselines, significantly improves the base LLM performance with only a lightweight prompt and no few-shot examples.

System Prompt

```
You are exploring a text adventure game.  Your goal is to make
progress and increase your score.

Generate actions that explore new possibilities and make progress.

Respond in JSON format:
{
  "thought":  "Your reasoning about what to try",
  "action":  "the exact command to execute"
}
```

User Prompt (at initial step)

```
{Context includes:  W_Local, global frontier trajectories F, previous
exploration trajectories, and the current trajectory.}
==================================================

Step 1:
Observation:  {current observation}
Score:  {current score}
Valid actions:  {comma-separated list of valid actions}

What is your next move?
```

User Prompt (subsequent steps)

```
Step {N}:
Observation: {current observation}
Score: {current score}
Valid actions: {comma-separated list of valid actions}

What is your next move?
```

E QUALITATIVE EXAMPLES

E.1 GLOBAL WORLD MODEL

Global World Model Example (Zork1)

```
Strategic Analysis of Game Trajectories

==================================================

1. FRONTIER & EXPLORATION STATUS

Successfully Reached Areas:
- Starting point: Mailbox and House exterior (north, south, east,
west of house)
- Forest Path and Clearing with grating
- Up a Tree (bird's nest with egg and leaflet)
- Behind House (window entry)
- Kitchen and Living Room inside the house
- Attic (Trajectory 1)
- Cellar and Troll Room (Trajectory 1 & 2)

Unexplored or Inaccessible Areas:
- Upstairs beyond the dark staircase (pitch black, dangerous)
- West door with gothic lettering (nailed shut)
- Crawlway south in cellar and forbidding hole west in Troll Room
- Steep metal ramp in cellar (unclimbable)
- Chasm area (Trajectory 1 ends in death)
- Any further passages beyond troll or trap door

==================================================

2. GAME CHECKPOINTS & PROGRESS

Key Milestones:
- Obtaining jewel-encrusted egg and leaflet from mailbox/tree
- Using egg and leaflet in grating to unlock cellar access
- Entering house via window and acquiring items (food, garlic,
water, lantern, sword, rope, knife)
- Opening trap door and descending into cellar
- Defeating or engaging troll in cellar

Items/Abilities Unlocking New Areas:
- Egg + Leaflet → Grating → Cellar entrance
- Lantern (turned on) → Safe navigation in dark cellar
- Sword (elvish, glowing) → Combat with troll
- Rope and knife (potential utility, not fully leveraged yet)
- Food, garlic, water (likely survival or puzzle items)

Current Phase: Early-to-mid game - exploration of house and
cellar, initial combat encounters
```

```
==================================================

3. BOTTLENECKS & CHALLENGES

Common Stuck Points:
- Combat with troll (Trajectories 1 & 2 end in death or near death)
- Navigating dark areas without proper light (pitch black upstairs)
- Attempting risky actions (jumping over chasm leads to death)
- Locked or barred doors (gothic door, trap door closing behind
player)

Obstacles Blocking Progress:
- Physical barriers (nailed door, barred trap door)
- Hostile NPC (troll)
- Environmental hazards (darkness, chasm)
- Limited inventory or missing key items

Missing Resources/Knowledge:
- Effective combat strategy or stronger weaponry to defeat troll
safely
- Means to reopen or bypass barred trap door
- Safe traversal methods for chasm or dark upstairs
- Possible puzzle solutions involving rope, knife, or other items

==================================================

4. REWARD STRUCTURE

When Points Are Earned:
- +5 for taking the egg
- +10 for acquiring food and water items in kitchen
- +25 for entering cellar (significant milestone)
- +5 for moving north from troll room to passage
- -10 on death and respawn (penalty)

Highest Reward Actions:
- Descending into cellar (+25)
- Collecting key items early (+5 to +10)
- Progressing past major checkpoints

Scoring Patterns:
- Early exploration yields small incremental rewards
- Major area transitions yield large rewards
- Death penalizes score, encouraging cautious play

==================================================

5. NEXT INVESTIGATION GOALS

Specific Objectives:
- Develop reliable strategy to defeat or bypass troll without dying
- Investigate use of rope and knife for puzzles or combat
- Explore crawlway south and forbidding hole west in troll room
- Find safe method to cross or bypass chasm
- Attempt to open or circumvent nailed gothic door

Most Promising Unexplored Areas:
- Crawlway south and forbidding hole west in cellar/troll room
- Upstairs dark staircase (with lantern or other light source)
- Beyond barred trap door if it can be reopened
- Areas beyond chasm once safe crossing method found
```

E.2 Local World Model

Local World Model Example (Zork1)

```
Based on detailed exploration attempts, here are KEY STATES with
discovered advantages:

==================================================

STATE: The Troll Room
Small room with passages east/south, hole west; troll blocks all
exits; sword glows

ADVANTAGES:
- "hit troll with sword" → kills troll, clears all exits for
exploration (score:  40)
- "take axe" after troll death → obtains valuable weapon for later
puzzles
- avoid engaging without glowing sword → likely failure
- optimal:  kill troll first, then take axe (mandatory to proceed)

==================================================

STATE: Maintenance Room (Flood Control Dam #3)
Room with colored buttons, tool chests, wrench, screwdriver,
toothpaste, leaking pipe

ADVANTAGES:
- "take wrench, screwdriver, tube" → essential tools for
environment interaction
- "push blue button" → triggers leak, raises water level to access
new areas
- "push red button" → toggles lights, affects water level
- avoid throwing lantern → breaks critical light source
- optimal:  collect tools → manage buttons → control water without
drowning

==================================================

STATE: Temple / Torch Room / Dome Room / Altar
Large temple with inscriptions; dome with railing; rope for
descent; ivory torch; brass bell; gold coffin

ADVANTAGES:
- "take ivory torch" → stable light for deeper cave exploration
- "take bell" → key item for spirit/wraith interaction
- "ring bell at Entrance to Hades" → paralyzes wraiths, enables
passage
- "blow out candles" → enables safe descent or passage
- optimal:  acquire torch → bell → sceptre → manipulate altar →
control spirits

==================================================

STATE: East-West Passage / Chasm Area
Narrow passage with stairs; chasm with paths; multiple routes
(north/east/west/up/down)

ADVANTAGES:
- "east" then "north" → leads to Reservoir South and further areas
- "tie rope to railing" → enables safe descent into lower levels
- avoid getting stuck in loops → wastes moves
- optimal:  explore chasm edges → use rope for vertical → access
```

```
Dome/Torch

==================================================

Cross-Cutting Insights:
- Inventory Management:  Strategic dropping/picking essential for
critical artifacts
- Light Preservation:  Maintaining lantern/torch crucial for dark
exploration
- Combat Readiness:  Glowing sword indicates combat opportunity
(essential for progress)
```

F ASSUMPTION OF ENVIRONMENT DETERMINISM

Our implementation assumes deterministic environments, which is leveraged by the state restoration mechanism which replays actions to return to a selected state. This limits applicability in stochastic environments where action replay may not return to the intended state. Our trajectory-based design facilitates intuitive extensions to such settings, where a potential approach is framing state restoration as a goal-reaching problem guided by replay trajectories.

We note that the MAR component naturally handles stochasticity since it samples multiple exploration paths from a selected state and performs inference over the paths as a set, enabling the LLM-based analysis to reflect on observed stochastic variations.

G SCALING WITH STRONGER LLMS

Games	*RL*	*LLM-based*									
		GPT-4.1 mini					*GPT-4.1*				
	XTX	**ReAct**	**Rfl**	**ICRL**	**IGE**	**GLoW**	**ReAct**	**Rfl**	**ICRL**	**IGE**	**GLoW**
Steps	800K	1K	1K	1K	1K	1K	1K	1K	1K	1K	1K
Enchanter	52.0	46.7±9.4	48.3±9.4	43.3±8.5	50.0±7.1	61.7±20.1	38.3±2.4	58.3±2.4	45±7.1	68.3±18.4	**98.3**±4.7
Zork1	**103.4±10.9**	48.3±4.7	48.0±5.0	51.7±4.7	44.3±0.5	73.0±4.5	45.0±0.0	54.3±4.5	48.0±2.8	86.7±24.1	103.0±6.8
Zork3	4.2±0.1	3.0±0.0	2.7±0.5	3.0±0.0	3.7±0.9	4.3±0.9	3.3±0.5	2.7±0.5	3.0±0.8	3.0±0.0	**5.0**±0.0
Deephome	77.7±2.1	11.0±4.2	22.0±1.6	24.0±5.7	71.3±4.9	75.0±8.7	32.3±19.6	22.3±1.7	34.7±18.7	82.0±8.6	**114.7±27.8**
Ludicorp	78.8	19.7±0.9	21.7±1.2	32.0±7.1	28.3±11.3	73.7±11.0	31.0±2.8	29.0±0.8	31.7±0.5	**89.0±7.8**	79.0±16.8
Balances	24	10±0.0	10±0.0	11.7±2.4	10.0±0.0	16.7±2.4	18.3±2.4	18.3±2.4	16.7±2.4	16.7±2.4	**26.7±2.4**

Table 7: Results comparing GPT-4.1 mini and GPT-4.1 on Extreme/Difficult games. XTX (best RL method) is shown for reference.

To validate that GLoW generalizes across model capabilities, we evaluated with GPT-4.1 on the 6 Extreme/Difficult games. GLoW with GPT-4.1 surpasses XTX on 5 out of 6 games while using 800× fewer interactions, showing that GLoW is robust across LLMs.

H LLM USAGE

We utilized Claude for minor grammar and language edits in paper writing.