LLMV-AGE: VERIFYING LLM-GUIDED PLANNING FOR AGENTIC EXPLORATION IN OPEN-WORLD RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have shown promise in enhancing reinforcement learning (RL) through task decomposition, yet their generated subgoals often lack reliability, leading to inefficient exploration and suboptimal policy learning. In this paper, we propose **LLMV-AgE** (**V**erification of **LLM**-guided planning for **Ag**entic **E**xploration), an RL framework that integrates LLM-guided subgoal planning with a hierarchical verification process to ensure both semantic validity and environmental feasibility. LLMV-AgE systematically assesses subgoal coherence, corrects invalid plans through iterative refinement, and aligns policy learning with reliable, goal-driven objectives. Empirical results on the procedurally generated Crafter benchmark demonstrate that LLMV-AgE significantly improves exploration efficiency and policy robustness by mitigating the impact of hallucinated subgoals and guiding agents toward more achievable goals.

1 INTRODUCTION

While humans effortlessly navigate new environments by leveraging prior knowledge and adapting to unfamiliar situations, autonomous agents face significant challenges in effective exploration (Sutton & Barto, 1998). Unlike humans, agents must acquire exploration strategies through trial-and-error interactions, often in environments lacking predefined guidance. This challenge becomes even more pronounced in procedurally generated open-world environments, where agents encounter dynamic, partially observable states, stochastic transitions, and sparse rewards. Unlike static or deterministic settings, these environments demand adaptive exploration strategies capable of generalizing beyond fixed patterns or exhaustive state enumeration. The complexity is further heightened in long-horizon tasks, where agents must seamlessly integrate local decision-making with high-level strategic planning to achieve diverse, evolving objectives (Cobbe et al., 2020; Mohanty et al., 2021; Hafner, 2022; Moon et al., 2023; Yuan et al., 2023; Zhou & Garg, 2023; Li et al., 2024; Andres et al., 2025). In such settings, fostering *agentic exploration*, where agents autonomously generate, evaluate, and adapt their exploration strategies, is crucial for robust performance.

To address these challenges, recent advances have explored the integration of large language models (LLMs) into reinforcement learning (RL) pipelines, introducing a powerful paradigm for high-level planning (Valmeekam et al., 2023; Huang et al., 2024). By leveraging their vast pretrained knowledge, LLMs can decompose complex tasks into structured subgoals, providing zero-shot, task-agnostic reasoning that enhances both exploration and decision-making (Du et al., 2023; Liu et al., 2024; Zhang & Lu, 2024). This *LLM-guided planning* approach holds great potential in environments where traditional, handcrafted heuristics fall short, particularly for long-horizon tasks that benefit from hierarchical decomposition. For example, in procedurally generated environments like Crafter (Hafner, 2022), LLMs can propose high-level subgoals such as "gather wood," "craft tools," or "build shelter," guiding the agent's low-level policy towards meaningful objectives.

However, despite their remarkable reasoning capabilities, LLMs frequently generate plans that suffer from *hallucination*, e.g., producing subgoals that are semantically incoherent (e.g., "eat stone"), contextually irrelevant (e.g., "plant tree" without the required resources), or infeasible to execute within the environment's constraints (Ahn et al., 2022; Du et al., 2023; Farquhar et al., 2024). This limitation arises from the inherent gap between the LLM's abstract, pretrained knowledge and the grounded, task-specific dynamics of the RL environment. Unlike classical symbolic planners, which offer explicit guarantees through formal representations, LLMs operate as black-box models,

making it non-trivial to assess the reliability of their outputs. This poses a significant risk for agentic exploration, as unreliable plans can misguide the agent's learning trajectory.

To bridge this gap, a key challenge is to systematically *verify* LLM-generated plans to ensure they are both meaningful and achievable within the target domain. This raises critical questions:

- *How can we verify and correct LLM-generated subgoals to ensure they are grounded in the RL task domain?*
- *How does the plan-verify-correct process impact the RL agent's policy learning performance?*

Addressing these questions is essential not just for enhancing agent performance but also for fostering agentic exploration, where agents autonomously generate, evaluate, and adapt their own plans with minimal human intervention (Durante et al., 2024). In this context, verification serves as a critical self-assessment mechanism, enabling agents to detect inconsistencies in their plans, reason about potential errors, and iteratively refine their strategies based on environmental feedback. This capability moves beyond static plan execution, empowering agents with a degree of autonomy that supports continual adaptation and robust decision-making in dynamic environments.

In this paper, we propose **LLMV-AgE** (*Verification of LLM-guided Planning for Agentic Exploration*), a verification framework designed to enhance LLM-guided exploration in RL. While LLMs can generate high-level plans to guide agents, these plans often contain unreliable elements that can mislead exploration and hinder learning. To address this, LLMV-AgE introduces a hierarchical verification process that systematically assesses the reliability of LLM-generated plans through two key dimensions: *semantic validity*, which ensures that subgoals are contextually coherent and meaningful within the task domain, and *feasibility*, which verifies whether these subgoals are achievable given the agent's current state and environmental constraints.

Built upon the solid foundation of *procedurally generated hierarchical Markov decision processes* (PG-HMDPs), LLMV-AgE exploits the hierarchical dependencies between achievements, subgoals, and environmental dynamics to enable principled verification. By systematically capturing the structural relationships embedded in the task hierarchy, our framework formalizes plan validity and feasibility as measurable properties grounded in the agent's decision-making process. Through this approach, LLMV-AgE bridges the gap between LLM-generated plans and reliable, adaptive agent behavior, paving the way for more trustworthy autonomous systems capable of robust exploration in complex environments.

2 Methodology

2.1 Problem Definition

We consider a **P**rocedurally **G**enerated **H**ierarchical **M**arkov **D**ecision **P**rocess (PG-HMDP) defined as the tuple $\langle \Omega, \mathcal{G}, \mathcal{O}, \mathcal{A}, R, P, \gamma \rangle$. Ω represents high-level *achievements* (e.g., *craft stone pickaxe*, *craft wood sword*) that the agent aims to unlock as many as possible within the environment. Unlocking each achievement is procedurally dependent on preceding achievements and prerequisite lower-level *subgoals* $g \in \mathcal{G}$ (e.g., *gather wood* and *place table* for unlocking the achievement *craft wood pickaxe*). This procedural achievement-unlocking process is governed by the transition function $P_\Omega(\omega' \mid \omega, g)$, which defines the probability of transitioning from one achievement ω to the next ω' based on the completion of subgoals.

Solving this long-horizon, sparse-reward problem requires progressive *planning* over achievements and subgoals. In this work, we leverage pretrained LLMs to guide planning at the subgoal level $\mathcal{G}$, proposing coherent *subgoal* sequences g as a high-level, training-free policy to guide the agent's learning of a low-level control policy $\pi(a \mid o, g)$, where $a \in \mathcal{A}$ is the agent's action based on its partial observation o and active subgoal g. The goal of the policy π is to maximize the expected cumulative reward: $\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(o_t, a_t \mid g_t)\right]$, where $\gamma \in [0, 1)$ is the discount factor that balances immediate and future rewards, and $R : \mathcal{O} \times \mathcal{A} \mid \mathcal{G} \to \mathbb{R}$ defines the reward function based on the agent's interactions, conditioned on the current planned subgoal g_t.

To effectively solve PG-HMDPs, it is crucial for agents to explore with strategically planned subgoals that align with procedural dependencies and maximize long-term achievements. We propose a

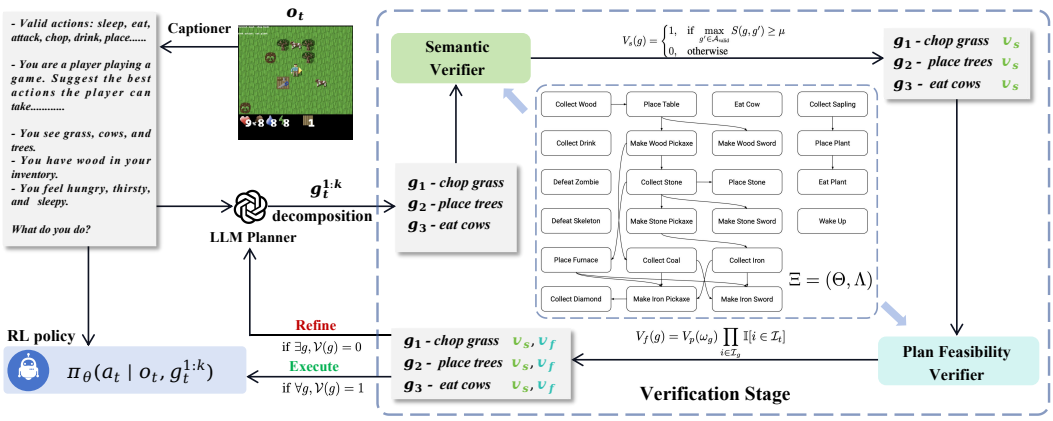


Figure 1: An overview of the **LLMV-AgE** framework. The LLM receives textual observations generated by a captioner, decomposes the task, and generates k subgoals $g_t^{1:k}$ as the plan. In the verification phase, subgoals are first evaluated by a ***semantic verifier***, followed by a ***plan feasibility verifier***. If none of the subgoals pass the validation, the system iteratively proposes new subgoals.

verification framework in Sec 2.2 to assess and constrain LLM-decomposed subgoals, guiding exploration towards plausibly useful goals that support efficient skill acquisition.

2.2 Verification of LLM-Guided Subgoal Planning

Subgoals g, expressed in natural language, directly influence the RL agent's control policy $\pi(a \mid o, g)$ and reward function $R(o, a \mid g)$, making their reliability critical. However, LLM-generated subgoals often suffer from two key issues: (1) *semantic validity*, which investigates if subgoals are irrelevant or incoherent, and (2) *plan feasibility*, which determines if subgoals cannot be achieved given the agent's current state. To address these challenges, we introduce a verification framework that systematically assesses subgoals for both contextual relevance and plan achievability, enhancing their alignment with the environment's dynamics and the agent's capabilities.

Semantic Verification Subgoals generated by LLMs Du et al. (2023) are expressed as action templates of the form $\{\text{verb}\} \times \{\text{noun}\}$, such as *"eat cow"* or *"eat zombie"*, where some combinations like the latter represent invalid actions in the task environment. To ensure semantic validity, we define a set of valid actions $\mathcal{A}_{\text{valid}}$ derived from the environment's task schema. Given a candidate subgoal $g = (\text{verb}, \text{noun})$, we introduce a semantic verification function $V_s(g)$ defined as:

$$V_s(g) = \begin{cases} 1, & \text{if } \max_{g' \in \mathcal{A}_{\text{valid}}} S(g, g') \geq \mu \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $S(g, g')$ measures the semantic similarity between g and valid subgoals $g' \in \mathcal{A}_{\text{valid}}$, and μ is a predefined similarity threshold. The similarity function $S(\cdot, \cdot)$ can be instantiated using embedding-based cosine similarity or other language-based semantic distance metrics. This formulation allows verification to tolerate minor lexical variations while rejecting subgoals that deviate from task-relevant semantics. For example, *"plant tree"* and *"grow tree"* may differ lexically but would achieve high similarity under S, thus passing verification.

Plan Feasibility Verification The procedural structure of achievements Ω is modeled as a directed acyclic graph (DAG) $\Xi = (\Theta, \Lambda)$, where Θ represents the set of nodes corresponding to achievements $\omega \in \Omega$, and Λ denotes the set of edges indicating dependencies between achievements. Specifically, an edge $(\omega_i, \omega_j) \in \Lambda$ signifies that achievement ω_j depends on the prior completion of ω_i. The feasibility of achieving ω_j is determined by the completion of its prerequisite achievements, denoted as $\text{Pa}(\omega_j)$, and is formalized as:

$$V_f(\omega_j) = \prod_{\omega_i \in \text{Pa}(\omega_j)} f_\omega(\omega_i), \tag{2}$$

where $f_\omega(\omega_i)$ evaluates the feasibility of transitioning from ω_i to ω_j based on the fulfillment of necessary dependencies. Each achievement ω is associated with a set of subgoals $\mathcal{G}_\omega = \{g_1, g_2, \ldots, g_n\}$. The feasibility of each subgoal g depends on the availability of required inventory items. Let $\mathcal{I}_g = \{i_1, i_2, \ldots, i_m\}$ denote the set of items required for subgoal g, and $\mathcal{I}_t$ represent the inventory at time step t. The feasibility of a subgoal is defined as:

$$V_f(g) = \prod_{i \in \mathcal{I}_g} \mathbb{I}[i \in \mathcal{I}_t], \tag{3}$$

where $\mathbb{I}[\cdot]$ is an indicator function that returns 1 if item i is present in the inventory at time t, and 0 otherwise. The overall verification score for achieving ω integrates both achievement-level dependencies and subgoal-level constraints:

$$V(\omega) = V_f(\omega) \cdot \prod_{g \in \mathcal{G}_\omega} V_s(g) \cdot V_f(g). \tag{4}$$

This unified formulation ensures that subgoals are both logically consistent within the procedural achievement structure and practically feasible given the agent's inventory dynamics. The verification process leverages hierarchical abstractions of PG-HMDP dynamics, synthesizing dependencies across achievements, subgoals, and inventory to capture the core complexities of procedural decision-making. By bridging high-level plans with low-level execution, the framework offers a scalable solution readily generalizable to other hierarchical MDPs with language-based planning.

2.3 Policy Training with Verified Subgoals

When a subgoal is identified as invalid, we apply a correction mechanism to generate meaningful alternatives. This *planning-verification-correction* process forms a *chain-of-planning*, where the primitives that the previous subgoal failed to satisfy are provided as cues to the LLM for self-correction. This iterative process prompts the LLM to refine subgoals progressively, enhancing their relevance and achievability.

The corrected subgoals are then used to guide the agent's planning by generating intuitive reward signals. Specifically, the agent's transitions are captioned and compared with the subgoal g_{verify} to produce a reward signal defined as:

$$\mathcal{R}_{\text{int}}(o, a, o' \mid g_{\text{verify}}) = \begin{cases} \max\limits_{j=1,\ldots,k} \Delta\left(C_{\text{transition}}(o, a, o', g_{\text{verify}})\right) & \text{if } > \lambda \\ 0 & \text{otherwise} \end{cases}$$

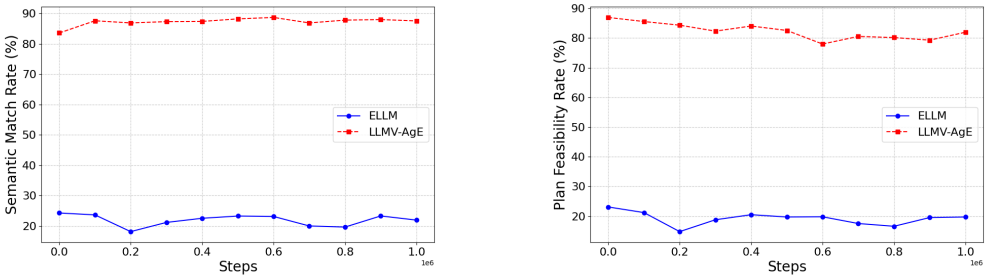
where λ is a similarity threshold, and $\Delta(C_{\text{transition}}(o, a, o'), g_{\text{verify}})$ represents the cosine similarity between language representations of transitions and subgoals, calculated as:

$$\Delta(C_{\text{transition}}(o, a, o'), g_{\text{verify}}) = \frac{E(C_{\text{transition}}(o, a, o')) \cdot E(g_{\text{verify}})}{\|E(C_{\text{transition}}(o, a, o'))\| \, \|E(g_{\text{verify}})\|}.$$

Here, $E(\cdot)$ is a text encoder that maps transitions and subgoals into embeddings. For each planning step, multiple (k) subgoals are provided, and the agent receives a reward based on the maximum similarity score among them, encouraging behaviors aligned with any of the refined subgoals.

3 Experiments

Experimental Settings We evaluate LLMV-AgE in the challenging Crafter environment, a 2D, procedurally generated, and partially observable open-world setting. To assess the impact of subgoal verification, we compare LLMV-AgE with a strong LLM-guided planning baseline, ELLM (Du et al., 2023). Following the standard practice of ELLM, initial goals are generated using a *verb-noun* template, which are then verified and corrected through LLM-guided refinement in LLMV-AgE. For task decomposition, we employ Codex (Chen et al., 2021) as the LLM, consistent with the ELLM setup. Each experiment is conducted for 1 million steps to ensure robust evaluation.



(a) Semantic Verification (V_s). (b) Plan Feasibility Verification (V_f).

Figure 2: **Verification Results for V_s and V_f.** The baseline ELLM Du et al. (2023) generates a substantial number of invalid subgoals. In contrast, our method LLMV-AgE not only *verifies* but also *corrects* these subgoals, significantly enhancing the reliability and effectiveness of LLM-guided planning and fostering agentic exploration behavior.

Evaluation Results for Verification We evaluate the quality of subgoals generated by ELLM and LLMV-AgE with respect to semantic validity and plan feasibility. Figure 2 presents the evaluation curves before and after applying our verification-guided subgoal correction. From the curves, we observe that ELLM without verification suffers from significant hallucination. ELLM achieves below 30% for overall semantic validity and plan feasibility for the planning, which hinders the policy learning performance. LLMV-AgE can verify and correct those infeasible plans. With this plan-verify-correct pipeline, our method significantly enhance the reliability of the plans. Moreover, the consistent improvement across both metrics is crucial for the agent to mitigate hallucination errors and ensuring efficient goal-oriented exploration.

Evaluation Results for Policy Training We present the policy learning results for the pretraining stage with LLM planning for our method LLMV-AgE as well as ELLM on Crafter in Figure 3. The figure shows that training the policy with verification-guided subgoals significantly improves learning efficiency and performance. Specifically, LLMV-AgE achieves an average cumulative score of 7.7, compared to 5.5 for the baseline ELLM, demonstrating a +2.2 improvement. This gain reflects enhanced exploration efficiency in the challenging open-world long-horizon sparse reward exploration problems, as the agent focuses on semantically meaningful and practically feasible subgoals, reducing time spent on unproductive actions. Additionally, we observed that it is easier for the downstream RL policy $\pi(\cdot)$ to guide agentic exploration when conditioned on verified subgoals, as the subgoal g serves as a direct input to the policy network. This direct conditioning enables the agent to better interpret the planned objectives.

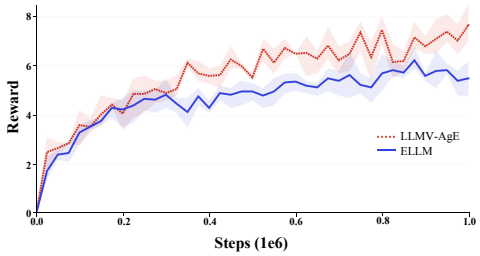


Figure 3: Learning curve for Crafter episodic reward. LLMV-AgE achieves significantly higher performance by unlocking more achievements compared to ELLM.

4 Conclusions

In this paper, we introduced LLMV-AgE, a verification framework designed to address the challenge of hallucinated plans in LLM-guided RL. By leveraging structured knowledge from PG-HMDPs and systematically verifying both semantic validity and plan feasibility, LLMV-AgE enhances the reliability of LLM-generated strategies, significantly improving exploration efficiency and decision-making in complex, open-world environments. Our results demonstrate that verification not only improves goal-directed exploration by aligning high-level plans with environmental constraints but also fosters agentic behavior, enabling agents to autonomously evaluate and refine their own plans. This capability strengthens trust in the LLM planner and supports more robust, adaptive decision-making. Moving forward, we aim to advance LLMV-AgE with an automatic verification pipeline that derives feasibility logic from agent playing records, further enhancing the agent's ability to self-assess and adapt in dynamic environments.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Alain Andres, Lukas Schäfer, Stefano V. Albrecht, and Javier Del Ser. Using offline data to speed up reinforcement learning in procedurally generated environments. *Neurocomputing*, 618:129079, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2048–2056, 2020.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*, pp. 8657–8677. PMLR, 2023.

Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, Katsushi Ikeuchi, Hoi Vo, Li Fei-Fei, and Jianfeng Gao. Agent AI: surveying the horizons of multimodal interaction. *CoRR*, abs/2401.03568, 2024.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. Understanding the planning of LLM agents: A survey. *CoRR*, abs/2402.02716, 2024.

Jiajian Li, Qi Wang, Yunbo Wang, Xin Jin, Yang Li, Wenjun Zeng, and Xiaokang Yang. Open-world reinforcement learning over long short-term imagination. *CoRR*, abs/2410.03618, 2024. URL `https://doi.org/10.48550/arXiv.2410.03618`.

Zeyuan Liu, Ziyu Huan, Xiyao Wang, Jiafei Lyu, Jian Tao, Xiu Li, Furong Huang, and Huazhe Xu. World models with hints of large language models for goal achieving. *CoRR*, abs/2406.07381, 2024.

Sharada P. Mohanty, Jyotish Poonganam, Adrien Gaidon, Andrey Kolobov, et al. Measuring sample efficiency and generalization in reinforcement learning benchmarks: Neurips 2020 procgen benchmark. *CoRR*, abs/2103.15332, 2021.

Seungyong Moon, Junyoung Yeom, Bumsoo Park, and Hyun Oh Song. Discovering hierarchical achievements in reinforcement learning via contrastive learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*, 2023.

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. *IEEE Trans. Neural Networks*, 9(5):1054–1054, 1998.

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models - A critical investigation. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023*, 2023.

Haoqi Yuan, Chi Zhang, Hongcheng Wang, Feiyang Xie, Penglin Cai, Hao Dong, and Zongqing Lu. Plan4mc: Skill reinforcement learning and planning for open-world minecraft tasks. *CoRR*, abs/2303.16563, 2023.

Wanpeng Zhang and Zongqing Lu. Adarefiner: Refining decisions of language models with adaptive feedback. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 782–799, 2024.

Zihan Zhou and Animesh Garg. Learning achievement structure for structured exploration in domains with sparse reward. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, 2023.