Behavior Tree Generation with LLM-MCTS-BT as a Pre-Planner Bridging Priors and Uncertainty

Extended Abstract

Haoming Wang
Intelligent Game and Decision Lab (IGDL)
Beijing, China
haomingwang@alumni.nudt.edu.cn

Xueying Wang*
Intelligent Game and Decision Lab (IGDL)
Beijing, China
w_xy93@163.com

Xiaoge Deng
Intelligent Game and Decision Lab (IGDL)
Beijing, China
dengxg@ustc.edu

Bin Li
Intelligent Game and Decision Lab (IGDL)
Beijing, China
libin_bill@126.com

Jingtao Qi
Intelligent Game and Decision Lab (IGDL)
Beijing, China
qijingtao14@163.com

Jinjing Sun
Intelligent Game and Decision Lab (IGDL)
Beijing, China
sunjinjing2015@buaa.edu.cn

Fu Li*
Intelligent Game and Decision Lab (IGDL)
Beijing, China
lifu11@alumni.nudt.edu.cn

ABSTRACT

While Behavior Trees (BTs) offer modular control for robotics, their manual construction requires significant expertise. Current Large Language Model (LLM) approaches to automation often struggle to leverage BT structure, lacking transparency, systematic exploration, and independence from prior knowledge. We propose LLM-MCTS-BT, a framework integrating LLMs with Monte Carlo Tree Search to emulate human design processes. Validated through simulation and LLM voting, our approach effectively automates the generation of robust, transparent Behavior Trees.

KEYWORDS

Behavior Trees; Large Language Models; Monte Carlo Tree Search

ACM Reference Format:
Haoming Wang, Xueying Wang, Xiaoge Deng, Bin Li, Jingtao Qi, Jinjing Sun, and Fu Li. 2026. Behavior Tree Generation with LLM-MCTS-BT as a Pre-Planner Bridging Priors and Uncertainty: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/VHJF3771

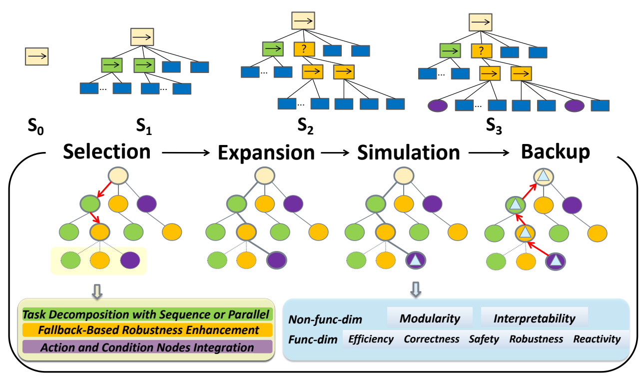

Figure 1: Overview of the LLM-MCTS-BT framework. The process spans four phases (S_0–S_3), illustrating BT structural evolution (top) and the MCTS mechanism (bottom).

1 INTRODUCTION

As a dominant paradigm for decision-making and execution in autonomous systems, Behavior Trees (BTs) have emerged as an effective framework for task planning and execution, gaining significant attention due to their modularity, reusability, and representational clarity [1, 4, 5, 8]. While Large Language Models (LLMs) offer new avenues for automated generation [10], existing methods based on prompt engineering [2] or fine-tuning [9, 11] often suffer from significant limitations: generated structures fail to fully leverage the hierarchical benefits of BTs, decision-making lacks transparency, and performance degrades severely without prior environmental knowledge due to an inability to handle uncertainty.

To address these challenges, we proposed a novel framework, LLM-MCTS-BT (shown in Figure 1), which synergistically integrates LLMs with Monte Carlo Tree Search (MCTS) for automated

*Corresponding author.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/VHJF3771

generation of BTs. By emulating the iterative design process of human experts, where structures evolve from simple sequences to complex hierarchies, the framework achieves a balance between functionality and robustness. Experiments in the ALFWorld environment demonstrate superior task completion rates in scenarios both with and without prior knowledge.

2 METHODOLOGY

We model BT generation as a combinatorial search problem $\mathcal{P}_{\text{search}} = \langle S, A(s), T, f \rangle$, where S is the state space of BT structures (initial state s_0 is a simple Sequence root), $A(s)$ is a state-dependent action space comprising node substitution, fallback insertion, and leaf node extension, $T : S \times A(s) \to S$ is a deterministic transition function, and $f : S \to \mathbb{R}$ is a multi-criteria evaluation function. The goal is to find $s^* = \arg\max_{s \in S} f(s)$ reachable from s_0.

2.1 Dynamic Action Space Generation

We employ the LLM to dynamically generate contextually relevant modifications for each state:

$$A(s) = \text{LLM}(\text{BT}_s,\ \tau,\ \mathcal{E}_0,\ P), \tag{1}$$

where τ is the task description, $\mathcal{E}_0$ is the initial environment state, and $P = \{p_1, p_2, p_3\}$ encodes three design principles:

- **Hierarchical Task Decomposition:** Uses *Node Substitution* to introduce nested Sequence or Parallel subtrees.
- **Robustness through Alternative Pathways:** Applies *Fallback Insertion* to create alternative execution branches.
- **Condition-Action Coupling:** Employs *Leaf Node Extension* to insert specific nodes for safety and granularity.

This focuses exploration on semantically promising regions rather than exhaustively enumerating all possible modifications.

2.2 Multi-Criteria Evaluation

We leverage the LLM to evaluate BT quality across functional properties (reactivity, efficiency, safety, correctness, robustness) and non-functional properties (modularity, interpretability) [7]. The evaluation implements a weighted aggregation:

$$\Delta = \frac{\sum_{i=1}^{n} w_i \cdot g_i(s)}{\sum_{i=1}^{n} w_i}, \tag{2}$$

where $g_i(s) \in \{-2, -1, 0, 1, 2\}$ represents LLM ratings per criterion and w_i are importance weights.

2.3 LLM-MCTS-BT Framework

Our framework adapts MCTS [3, 6] mechanism for BT generation:

Phase I: Selection & Expansion. Starting from the root, child nodes are recursively selected via UCB:

$$\text{UCB}(v') = \frac{Q(v')}{N(v')} + C\sqrt{\frac{2\ln(N(v))}{N(v')}}, \tag{3}$$

where $Q(v')$ and $N(v')$ are the cumulative value and visit count of node v', and C is the exploration constant. Upon reaching a node with unexplored children, the LLM generates candidate actions via Eq. (1), and one is applied to create a new child state.

Phase II: Simulation. The expanded node's BT is directly evaluated using the multi-criteria function (Eq. 2), replacing stochastic rollouts with structured LLM assessment.

Phase III: Backup. The reward Δ is propagated upward with depth-dependent decay: $Q(v) \leftarrow Q(v) + \Delta \cdot \lambda^{\text{depth}}$, where $\lambda \in (0, 1]$ ensures nodes closer to the evaluated node receive stronger updates.

3 EXPERIMENTS

3.1 Setup

We evaluate 40 household tasks (e.g., "put two soapbar in garbagecan") using **ALFWorld** [13], derived from the **ALFRED** dataset [12]. Testing spans **with prior knowledge** (full info) and **without prior knowledge** (partial observability) scenarios. We benchmark against Zero-Shot, One-Shot, and CoT baselines, optionally guided by Expert Policy trajectories (e.g., "go to toilet 1 → take soapbar 2 → ..."). Notably, our method operates *without* requiring such expert demonstrations. Metrics include **task completion rate**, **LLM-based approval**, and **LLM-based ranking**.

3.2 Results

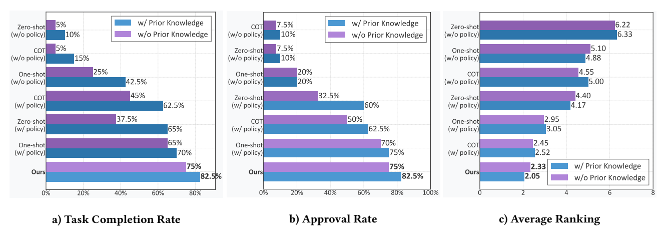

Figure 2: Evaluation across three dimensions: a) task completion rates, b) LLM-based approval rates, and c) average ranking scores. Our method achieves the best performance across all metrics under both information scenarios.

As shown in Figure 2, LLM-MCTS-BT dominates baselines with 82.5% (w/ prior knowledge) and 75.0% (w/o prior knowledge) task completion. LLM-based approval rates and ranking scores further confirm the structural quality and robustness of our generated BTs. Notably, our method avoids the severe dependence on expert demonstrations seen in baselines and maintains robustness under partial observability, demonstrating effective exploration beyond mere imitation. This advantage stems from MCTS-guided hierarchical decomposition replacing flat sequential structures, systematic integration of Fallback nodes for error recovery, and explicit condition-action coupling for precondition verification.

4 CONCLUSION

We present LLM-MCTS-BT, a framework combining LLMs' semantic understanding with MCTS's systematic exploration to automate BT generation through hierarchical task decomposition, fallback-based robustness enhancement, and condition-action coupling. Experiments demonstrate consistent superiority over baselines, particularly without expert demonstrations, bridging prior-knowledge-rich and prior-knowledge-constrained planning scenarios while preserving interpretability and modularity.

REFERENCES

[1] Bikramjit Banerjee. 2018. Autonomous acquisition of behavior trees for robot control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 3460–3467.

[2] Yue Cao and CS George Lee. 2022. Behavior-tree embeddings for robot task-level knowledge. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 12074–12080.

[3] Guillaume Chaslot, Sander Bakkes, Istvan Szita, and Pieter Spronck. 2008. Monte-carlo tree search: A new framework for game ai. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, Vol. 4. 216–217.

[4] Michele Colledanchise, Diogo Almeida, and Petter Ögren. 2019. Towards blended reactive planning and acting using behavior trees. In *2019 international conference on robotics and automation (ICRA)*. IEEE, 8839–8845.

[5] Michele Colledanchise, Ramviyas Parasuraman, and Petter Ögren. 2018. Learning of behavior trees for autonomous agents. *IEEE Transactions on Games* 11, 2 (2018), 183–189.

[6] Rémi Coulom. 2006. Efficient selectivity and backup operators in Monte-Carlo tree search. In *International conference on computers and games*. Springer, 72–83.

[7] Simona Gugliermo, David Cáceres Domínguez, Marco Iannotta, Todor Stoyanov, and Erik Schaffernicht. 2024. Evaluating behavior trees. *Robotics and Autonomous Systems* 178 (2024), 104714.

[8] Matteo Iovino, Edvards Scukins, Jonathan Styrud, Petter Ögren, and Christian Smith. 2022. A survey of behavior trees in robotics and ai. *Robotics and Autonomous Systems* 154 (2022), 104096.

[9] Riccardo Andrea Izzo, Gianluca Bardaro, and Matteo Matteucci. 2024. Btgenbot: Behavior tree generation for robotic tasks with lightweight llms. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 9684–9690.

[10] Fu Li, Xueying Wang, Bin Li, Yunlong Wu, Yanzhen Wang, and Xiaodong Yi. 2024. A Study on Training and Developing Large Language Models for Behavior Tree Generation. *arXiv preprint arXiv:2401.08089* (2024).

[11] Artem Lykov, Maria Dronova, Nikolay Naglov, Mikhail Litvinov, Sergei Satsevich, Artem Bazhenov, Vladimir Berman, Aleksei Shcherbak, and Dzmitry Tsetserukou. 2023. Llm-mars: Large language model for behavior tree generation and nlp-enhanced dialogue in multi-agent robot systems. *arXiv preprint arXiv:2312.09348* (2023).

[12] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10740–10749.

[13] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768* (2020).