Combining LLMs with Logic-Based Framework to Explain MCTS

Extended Abstract

Ziyan An
Vanderbilt University
Nashville, US

Xia Wang
Vanderbilt University
Nashville, US

Hendrik Baier
Eindhoven University of Technology
Eindhoven, Netherlands

Zirong Chen
Vanderbilt University
Nashville, US

Abhishek Dubey
Vanderbilt University
Nashville, US

Taylor T. Johnson
Vanderbilt University
Nashville, US

Jonathan Sprinkle
Vanderbilt University
Nashville, US

Ayan Mukhopadhyay
Vanderbilt University
Nashville, US

Meiyi Ma
Vanderbilt University
Nashville, US

ABSTRACT

In response to the lack of trust in Artificial Intelligence (AI) for sequential planning, we design a Computational Tree Logic-guided large language model (LLM)-based natural language explanation framework designed for the Monte Carlo Tree Search (MCTS) algorithm. MCTS is often considered challenging to interpret due to the complexity of its search trees, but our framework is flexible enough to handle a wide range of free-form post-hoc queries and knowledge-based inquiries centered around MCTS and the Markov Decision Process (MDP) of the application domain. By transforming user queries into logic and variable statements, our framework ensures that the evidence obtained from the search tree remains factually consistent with the underlying environmental dynamics and any constraints in the actual stochastic control process. We evaluate the framework rigorously through quantitative assessments, where it demonstrates strong performance in terms of accuracy and factual consistency.

KEYWORDS

Sequential Planning, Explainable AI, Large Language Model, MCTS

ACM Reference Format:
Ziyan An, Xia Wang, Hendrik Baier, Zirong Chen, Abhishek Dubey, Taylor T. Johnson, Jonathan Sprinkle, Ayan Mukhopadhyay, and Meiyi Ma. 2025. Combining LLMs with Logic-Based Framework to Explain MCTS: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

Artificial Intelligence (AI) algorithms often operate as black-box systems, offering little to no insight into the reasoning behind their outputs. As a result, domain experts hesitate to deploy these algorithms in real-world settings due to concerns over transparency, understandability, and accountability, leaving them without a clear understanding of the implications or rationale behind the decisions made by these AI models [5, 12–16].

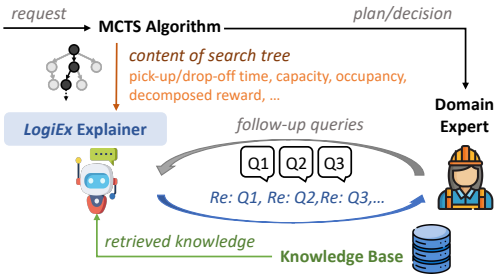

Figure 1: We explain sequential planning by combining domain knowledge, search process, and logical reasoning.

One family of such AI approaches that is widely used in complex sequential planning problems such as manufacture engineering [17] and transit route planning [20] is Monte Carlo Tree Search (MCTS) [10]. Understanding the results and decisions of MCTS is challenging even for experts due to the large, sampling-based search trees from which they are derived [2, 4]. Therefore, we develop a logic-enabled large language models (LLMs) framework that integrates knowledge and symbolic reasoning with natural language, creating a robust yet expressive xAI system for explaining planning algorithms like MCTS (Figure 1).

Aiming to address a flexible range of free-form user queries, our framework leverages advanced LLMs, which enables the development of xAI systems based on natural language [6, 8, 19]. More specifically, it offers broad flexibility in handling queries by converting natural language inquiries submitted via a chat interface into parameterized variables and logic expressions. It then evaluates the search tree based on the criteria specified by these logic expressions, and the results are presented in the final explanation, once again expressed in natural language. The framework also enables an unlimited number of follow-up queries, facilitating an interactive, back-and-forth communication with the user.

2 METHOD

Background. As the testbed for our framework, we use a para-transit planning scenario formulated as a Markov Decision Process

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

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

(MDP). We define the state space, action space, constraints, and reward of the MDP. State transitions are driven by a simulated demand model for paratransit trip requests. We leverage MCTS to generate vehicle assignment decisions, which is initiated at each "decision epoch" [9].

Query Categories and Types. The first category of queries, called *post-hoc queries*, seeks explanations for the returned plan after the algorithm has completed its execution and focuses on explaining specific MCTS decisions. The second category, called *background knowledge-based queries*, focuses on the MCTS decision-making process in general. After the user submits a query, and a Query-Classification LLM component interprets the new query and attempts to classify its intent to one of two categories. User queries are not restricted in terms of content or narration. However, to strategically address these queries, we pre-define 26 specific query types based on the user's underlying intentions for the first category. In contrast, for the second category, queries answerable with background knowledge, there are no specific types, as one piece of knowledge can address multiple queries.

Logic Generator and Parser. Each pre-defined query type is associated with a few-shot prompt, containing example pairs of input queries and output logic. After a new query is classified into a specific query type, the corresponding prompt is used to guide the logic generation LLM component in formulating a logic statement for the query. We categorize all user questions based on the type of evidence required to answer them: those that can be addressed with base-level evidence, referring to information directly extracted from a tree node; those that rely on derived evidence, requiring consideration of multiple nodes across different depths or branches; and those that require logic comparison evidence, involving both multi-level calculations and comparisons between two branches using Computation Tree Logic (CTL) [7]. The variables are organized into a three-level hierarchical structure, where each level builds upon the variables and logic defined in the previous level.

Logic Scorer. To obtain both quantitative and qualitative evidence, we define scorer functions that take the MCTS tree including states and actions as input and return either numerical or boolean values based on the evaluation of specific criteria [3]. For base-level variables, the result is obtained by identifying the target node corresponding to the variable through tree traversal. For derived evidence variables, we further define formulas to calculate the overall averaged quantitative result across all relevant nodes in the search tree. Lastly, we utilize CTL model checking algorithms to obtain logic comparison evidence, where the input is the MCTS tree.

Knowledge Retrieval. To provide domain knowledge-informed explanations for category two queries, we prepared a lightweight knowledge base containing approximately 3,000 words, divided into 34 chunks. This knowledge base covers background information on paratransit services and the MCTS algorithm, as well as detailed components of the MDP, including predefined constraints, algorithm objectives, and reward functions. We leverage the RAG technique with the OpenAI `text-embedding-3-small` model to obtain the top k results, passing information chunks to the LLM only if their relatedness scores exceed a predefined threshold.

Table 1: Quantitative evaluation results.

Method	Metric	@1↑	@3↑
Llama3.1	FactCC / BERT	25.77% / 06.15%	34.62% / 12.31%
Ours (Llama)	FactCC / BERT	67.88% / 86.54%	**83.27%** / **97.50%**
GPT-4o	FactCC / BERT	42.31% / 40.00%	51.15% / 55.77%
Ours (GPT)	FactCC / BERT	**72.12%** / **88.46%**	81.35% / 94.81%

Generating Explanations. Once the list of calculated evidence or retrieved domain knowledge is obtained, the framework engages with a Question-Answering LLM to generate the final response. Key pieces of information provided to the LLM include the original user query, the evidence variables used, the result from the scorer function obtained in the previous step, and the retrieved knowledge.

3 EVALUATIONS

We quantitatively evaluate the framework to answer the research question (RQ): Does our framework outperform existing LLMs in generating factually accurate and relevant explanations? We consider three LLM models for our evaluation: GPT-4 [1], GPT-4o [1], and Llama3.1 [18] model. We systematically generated 620 distinct input queries along with their corresponding ground truth. We compare the generated explanations using two metrics: BERTScore [21] and FactCC [11].

Factual Consistency Results and Discussions. As shown in Table 1, the best result achieved across basic LLMs was a 51.15% FactCC score, and the highest BERTScore achieved was 55.77%. Both results suggest that basic LLMs struggle to generate relevant and factually accurate explanations directly. We then compared them with our framework with GPT-4 and Llama3.1 as backbone models, where we observed significant improvements. our framework consistently outperformed the basic LLMs across all categories. Specifically, we observed a 2.40× improvement using Llama3.1 and a 1.59× improvement using the GPT-4 model for FactCC score. The improvement in BERTScore was even more evident, with an overall increase of 7.92× for the Llama3.1 backbone model and 1.70× for the GPT-4 backbone model, respectively.

4 CONCLUSION

We present an explainability framework for MCTS sequential planning. Tested within the context of paratransit planning scenarios, our framework can address a variety of user queries by offering post-hoc explanations and RAG-based explanations, through three-level hierarchical evidence. We thoroughly evaluated the framework performance quantitatively. Results show that our framework achieved overall superior performance compared to traditional LLMs.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation (NSF) under Award Numbers 2028001, 2220401, 2151500, CNS-2238815 and CNS-1952011, AFOSR under FA9550-23-1-0135, DARPA under FA8750-23-C-0518, NWO under NWA.1389.20.251, and Horizon Europe under 101120406. The paper reflects only the authors' view and does not necessarily reflect the views of the sponsoring agencies.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Ziyan An, Hendrik Baier, Abhishek Dubey, Ayan Mukhopadhyay, and Meiyi Ma. 2024. Enabling MCTS Explainability for Sequential Planning Through Computation Tree Logic. *arXiv preprint arXiv:2407.10820* (2024).

[3] Ziyan An, Taylor T Johnson, and Meiyi Ma. 2024. Formal Logic Enabled Personalized Federated Learning through Property Inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 10882–10890.

[4] Hendrik Baier and Michael Kaisers. 2020. Explainable search. In *2020 IJCAI-PRICAI Workshop on Explainable Artificial Intelligence*. 178.

[5] Hendrik Baier and Michael Kaisers. 2021. Towards explainable MCTS. In *2021 AAAI Workshop on Explainable Agency in AI*. 178.

[6] Zirong Chen, Elizabeth Chason, Noah Mladenovski, Erin Wilson, Kristin Mullen, Stephen Martini, and Meiyi Ma. 2024. Sim911: Towards Effective and Equitable 9-1-1 Dispatcher Training with an LLM-Enabled Simulation. *arXiv preprint arXiv:2412.16844* (2024).

[7] Edmund M Clarke and E Allen Emerson. 1981. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Workshop on logic of programs*. Springer, 52–71.

[8] Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, et al. 2024. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in Neural Information Processing Systems* 36 (2024).

[9] Waldy Joe and Hoong Chuin Lau. 2020. Deep reinforcement learning approach to solve dynamic vehicle routing problem with stochastic customers. In *Proceedings of the international conference on automated planning and scheduling*, Vol. 30. 394–402.

[10] Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*. Springer, 282–293.

[11] Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Evaluating the factual consistency of abstractive text summarization. *arXiv preprint arXiv:1910.12840* (2019).

[12] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[13] Meiyi Ma, Ji Gao, Lu Feng, and John Stankovic. 2020. STLnet: Signal temporal logic enforced multivariate recurrent neural networks. *Advances in Neural Information Processing Systems* 33 (2020), 14604–14614.

[14] Meiyi Ma, John A Stankovic, and Lu Feng. 2021. Toward formal methods for smart cities. *Computer* 54, 9 (2021), 39–48.

[15] Joao Marques-Silva and Alexey Ignatiev. 2022. Delivering trustworthy AI through formal XAI. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 12342–12350.

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[17] M Saqlain, S Ali, and JY Lee. 2023. A Monte-Carlo tree search algorithm for the flexible job-shop scheduling in manufacturing systems. *Flexible Services and Manufacturing Journal* 35, 2 (2023), 548–571.

[18] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[19] Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. Naturalproofs: Mathematical theorem proving in natural language. *arXiv preprint arXiv:2104.01112* (2021).

[20] Di Weng, Ran Chen, Jianhui Zhang, Jie Bao, Yu Zheng, and Yingcai Wu. 2020. Pareto-optimal transit route planning with multi-objective monte-carlo tree search. *IEEE Transactions on Intelligent Transportation Systems* 22, 2 (2020), 1185–1195.

[21] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* (2019).