Explainable AI Agents for Autonomous System Evaluation: Integrating SHAP-Based Decision Attribution with Hierarchical Planning in Large Language Models

Sergei Wilson
Department of Computer Science, Colorado State University, Fort Collins, CO, USA.
sergei.wilson818@colostate.edu

Abstract

The increasing deployment of autonomous systems in critical socio-technical domains such as healthcare, transportation, and energy infrastructure demands robust evaluation frameworks that are both transparent and adaptive. Large language models offer powerful reasoning capabilities but suffer from opacity and a lack of structured decision attribution, hindering their use in high-stakes evaluation tasks. This paper proposes a novel architecture for explainable AI agents that integrate SHAP-based decision attribution with hierarchical planning to enable autonomous system evaluation. The framework comprises three layers: a hierarchical planner that decomposes evaluation objectives into subgoals, a large language model that executes reasoning and generates explanations, and a SHAP attribution module that quantifies the contribution of each input feature to the agent's decisions. By combining the structural clarity of hierarchical planning with the interpretability of SHAP values, the system provides both high-level strategic oversight and granular feature-level transparency. The paper examines structural trade-offs between explanation fidelity and computational efficiency, discusses deployment considerations across multi-agent environments, and analyzes governance implications including auditability, fairness, and regulatory compliance. Case illustrations from autonomous vehicle safety assessment and clinical decision support demonstrate the framework's viability. Forward-looking perspectives address sustainability, robustness against adversarial inputs, and policy integration. The proposed approach advances the state of the art by unifying attribution methods with planning formalisms, offering a path toward trustworthy autonomous evaluation agents.

Keywords

Explainable AI, SHAP, hierarchical planning, large language models, autonomous system evaluation, decision attribution, socio-technical governance.

1. Introduction

Autonomous systems are increasingly entrusted with decisions that affect public safety, economic equity, and environmental sustainability. From self-driving vehicles to automated diagnostic tools, these systems operate in complex, dynamic environments where errors can propagate with severe consequences. Evaluation of such systems—whether during development, certification, or post-deployment monitoring—requires not only accurate performance measurement but also deep understanding of why decisions are made. Traditional evaluation metrics such as accuracy, precision, or recall fail to capture the contextual reasoning behind an autonomous agent's actions. This gap motivates the need for

explainable AI agents that can themselves evaluate other autonomous systems while providing transparent justifications for their assessments.

Large language models have emerged as versatile tools for reasoning and natural language generation, enabling new forms of automated evaluation through conversational interfaces and chain-of-thought prompting [1]. However, their inherent opacity and tendency toward hallucination pose significant challenges for high-stakes applications. An LLM evaluation agent may produce plausible-sounding but incorrect assessments, and without a structured attribution mechanism, diagnosing failures becomes difficult. Furthermore, LLMs lack built-in hierarchical planning capabilities, making it hard to decompose complex evaluation tasks into manageable subproblems with verifiable guarantees.

This paper proposes a framework that integrates SHAP-based decision attribution with hierarchical planning to create explainable AI agents for autonomous system evaluation. SHAP values, grounded in cooperative game theory, provide a mathematically principled way to attribute a model's output to its input features, offering local interpretability [2]. Hierarchical planning, a classic AI paradigm, structures reasoning by decomposing high-level goals into sequences of subgoals that can be executed or verified independently [3]. By embedding an LLM within a hierarchical planner and augmenting its outputs with SHAP attributions, the resulting agent can evaluate autonomous systems with both strategic coherence and fine-grained transparency.

The remainder of the paper is organized as follows. Section 2 reviews relevant literature on explainable AI, hierarchical planning, and LLM-based agents. Section 3 details the proposed system architecture, emphasizing the interplay between the planning layer, the LLM executor, and the SHAP attribution module. Section 4 explores integration challenges and structural trade-offs, including computational overhead, explanation fidelity, and the tension between global and local interpretability. Section 5 presents case studies in autonomous vehicle safety assessment and clinical decision support, illustrating how the framework operates in practice. Section 6 discusses governance and policy implications, including auditability, fairness, and regulatory alignment. Section 7 concludes with reflections on sustainability, robustness, and future research directions.

2. Background and Related Work

Explainable AI has evolved from a niche research area to a central requirement for responsible deployment of machine learning systems. Among post-hoc explanation methods, SHAP (SHapley Additive exPlanations) stands out for its theoretical foundations in Shapley values from cooperative game theory, ensuring that attributions are additive and consistent across features [2]. SHAP has been applied to various domains including credit scoring, medical imaging, and natural language processing, providing feature-level importance scores that help users understand model predictions [4]. However, applying SHAP to generative models like LLMs is non-trivial due to their autoregressive nature and the combinatorial explosion of possible input configurations. Recent work has proposed adaptations for transformer-based architectures, but scalability and faithfulness remain open issues [5].

Hierarchical planning, originating in classical AI planning research, decomposes complex tasks into hierarchical arrangements of primitive actions and abstract subroutines [3]. In robotics, hierarchical task networks have enabled efficient execution of long-horizon tasks with state invariants. The integration of planning with learning, often termed hierarchical reinforcement learning, has seen renewed interest with the advent of neural planners that use

learned abstractions [6]. In the context of LLMs, hierarchical prompting techniques instruct the model to first reason about high-level steps before generating detailed outputs, thereby improving coherence and reducing error propagation [7].

Large language models have been increasingly used as components in autonomous agents for tasks such as code generation, question answering, and dialog management. Recent architectures combine LLMs with external tools, memory, and planning modules to extend their capabilities beyond text generation [1]. For evaluation purposes, LLM-based agents can simulate user interactions, generate test cases, or assess output quality. However, these agents themselves require evaluation, creating a meta-evaluation challenge. The need for transparent and interpretable evaluation agents has led to proposals that combine symbolic reasoning with neural methods [8]. Specifically, the work by Dou et al. introduced a plan-then-action framework that uses high-level planning guidance to improve LLM reasoning, demonstrating the utility of hierarchical decomposition in boosting both accuracy and interpretability [8].

SHAP interpretability has also been applied to API response quality prediction in large model systems, where attribution analysis helps identify which features most influence prediction outcomes [9]. Such studies highlight the practical value of SHAP in monitoring and debugging autonomous services. Additionally, concerns about security in distributed learning paradigms, such as vertical split learning, have motivated the development of backdoor defenses that rely on prototype consistency, illustrating the interplay between explainability and robustness [10]. In the realm of software vulnerability management, multi-agent reinforcement learning has been proposed for planning cost-aware patch strategies, underscoring the need for explainable planning in system maintenance [11]. These diverse contributions collectively motivate the integration of SHAP attribution with hierarchical planning for autonomous evaluation.

Despite these advances, a unified framework that combines SHAP-based local explanations with hierarchical goal decomposition for LLM-driven evaluation remains absent. Previous efforts have treated explainability and planning as separate concerns. This paper bridges that gap by proposing an agent architecture where hierarchical plans provide explicit structure, SHAP values offer input-level transparency, and the LLM serves as a flexible reasoning engine.

3. System Architecture

The proposed system, referred to as the Explainable Autonomous Evaluator, consists of three principal layers: the Hierarchical Planner, the LLM Reasoning Engine, and the SHAP Attribution Module. These layers operate in a feedback loop, enabling the agent to iteratively refine its evaluation of a target autonomous system.

The Hierarchical Planner is responsible for decomposing a high-level evaluation objective, such as assessing the safety of an autonomous driving policy, into a tree of subgoals. Each subgoal corresponds to a specific evaluation dimension, for instance, lane-keeping behavior, collision avoidance, or adherence to traffic rules. The planner uses domain knowledge encoded as a task network, which can be manually constructed or learned from examples [3]. The decomposition is recursively applied until leaf-level subgoals are primitive enough to be executed by the LLM or through direct computation. The planner outputs a directed acyclic graph of subgoals with dependencies, orders, and success criteria.

The LLM Reasoning Engine receives a subgoal from the planner and generates a natural language response that constitutes a partial evaluation. For example, given the subgoal "assess

lane-keeping stability over 1000 simulated miles," the LLM might query a simulation log, compute relevant metrics, and produce a textual summary. The LLM is instructed to include its reasoning chain and cite evidence. Importantly, the LLM does not make final binary judgments alone; instead, it outputs intermediate assessments and confidence scores that are aggregated by the planner. The planner then decides whether to proceed to the next subgoal, revisit a prior one, or flag an anomaly.

The SHAP Attribution Module is invoked after the LLM produces a response for a given subgoal. The module computes SHAP values for the features that the LLM considered in its reasoning. Because the LLM is a black-box text generator, direct SHAP computation over its internal representations is infeasible. Instead, the module operates on a surrogate model built from the LLM's input-output pairs at the subgoal level. Specifically, for each subgoal, the module defines a set of interpretable input features, such as simulation parameters, sensor readings, or historical metrics. The LLM's output (e.g., a risk score or a classification) is treated as the target function. KernelSHAP or TreeSHAP approximations are used to estimate feature contributions [2]. These attributions are then presented to the user alongside the LLM's textual explanation, providing dual transparency: the plan shows why the agent chose a particular evaluation path, and SHAP shows how each input factor influenced that subgoal's outcome.

The three layers communicate through a shared state store. The planner maintains a plan tree annotated with progress status. The LLM engine writes its outputs, confidence, and supporting data to the store. The SHAP module appends attribution metrics to each leaf node. At the end of execution, the planner produces a final report summarizing the overall evaluation, including the hierarchical plan, the LLM's reasoning traces, and SHAP values for each assessment step. This report can be audited by human reviewers or regulatory bodies.

4. Integration Challenges and Structural Trade-offs

Combining SHAP attribution with hierarchical planning inside an LLM agent introduces several structural trade-offs that must be carefully managed. The first trade-off involves the granularity of attribution. SHAP values are most informative when computed over a small, well-defined set of input features. However, in a hierarchical plan, subgoals may involve high-level abstractions such as "traffic complexity" or "driver intention," which are not directly measurable. Attributing a decision to such abstract features requires constructing proxy variables, which can introduce bias or reduce faithfulness. A finer granularity, on the other hand, increases computational cost exponentially as the number of features grows. The system must balance the need for interpretable attributions with the practical limits of SHAP estimation.

A second trade-off concerns the reliance on surrogate models for SHAP computation. Since the LLM's internal reasoning is opaque, the SHAP module approximates the LLM's decision boundary using simpler models trained on input-output pairs. This approximation inevitably loses fidelity. For subgoals where the LLM's behavior is highly nonlinear, the surrogate may fail to capture important interactions, leading to misleading attributions. One mitigation is to restrict SHAP application to primitive subgoals where the input space is low-dimensional, but this reduces the coverage of explanations across the entire hierarchical plan. Alternatively, the system could use inherently interpretable models for primitive steps, but that would sacrifice the flexibility of the LLM.

A third trade-off involves the computational overhead of executing SHAP for every leaf subgoal. Hierarchical plans for complex evaluation tasks may contain dozens or hundreds of leaf nodes. Running KernelSHAP for each node requires many repeated model queries, which compounds the cost of the LLM calls themselves. This can make real-time or near-real-time evaluation infeasible. One approach is to compute SHAP values only for the most critical subgoals identified by the planner, using heuristics such as confidence variance or historical failure modes. Another is to use efficient approximation methods like FastSHAP, which amortizes the computation with a learned explainer [12]. However, amortized explainers can themselves introduce errors and require additional training data.

A fourth trade-off relates to the dynamic nature of hierarchical planning. The planner may adapt its decomposition based on intermediate results. For instance, if the LLM's confidence in a subgoal is low, the planner could dynamically expand that subgoal with further refinements. This dynamism means that the final plan tree is not known a priori, making it difficult to precompute SHAP values. The SHAP module must be integrated as part of the loop, re-evaluating attributions whenever the plan changes. This requires careful handling of non-stationary input distributions and can complicate the theoretical guarantees of SHAP.

Despite these challenges, the integration offers significant benefits. The hierarchical structure naturally partitions the explanation space, reducing the cognitive load on human auditors. Instead of a single monolithic explanation for the whole evaluation, users can inspect subgoal-level attributions that are context-specific. Moreover, the planner's decomposition provides a causal ladder: high-level subgoals correspond to strategic objectives, while leaf subgoals correspond to specific computational steps. This separation aligns well with the distinction between global and local interpretability, a long-standing issue in XAI [13]. The framework thus enables users to zoom in or out depending on their need for detail.

5. Evaluation and Case Studies

To illustrate the framework's operation, we consider two representative domains: autonomous vehicle safety assessment and clinical decision support.

In the autonomous vehicle domain, the evaluation objective is to assess whether a given self-driving policy complies with safety standards such as ISO 26262 or the proposed UL 4600 standard. The hierarchical planner decomposes this objective into subgoals: scenario coverage, perception accuracy, decision logic correctness, and operational domain boundaries. For the "scenario coverage" subgoal, the LLM engine queries a simulation database for rare events like pedestriane merges or construction zones. It then generates a coverage score and a textual rationale. The SHAP module treats features such as number of simulated scenarios, diversity of weather conditions, and occurrence rate of near-misses as inputs. The computed SHAP values reveal, for instance, that the coverage score is most influenced by the presence of unprotected left-turn scenarios, prompting engineers to add more such tests. The final evaluation report includes the full plan tree, each leaf node's SHAP bar chart, and the LLM's commentary.

In clinical decision support, the evaluation target is an AI-based diagnostic tool for detecting lung nodules in chest X-rays. The evaluator agent must assess the tool's sensitivity, specificity, and calibration across different patient demographics. The hierarchical planner divides the task into subgoals: data distribution analysis, model calibration, subgroup performance analysis, and fairness auditing. For the subgroup performance subgoal, the LLM computes accuracy metrics for each demographic group and produces a text explanation. The

SHAP module uses features such as group size, prevalence of disease, and image quality scores. Attributions may show that differences in performance are largely driven by varying disease prevalence rather than algorithmic bias, providing nuanced evidence for regulatory review. If the SHAP values indicate that image quality is a significant factor, the planner may add a new subgoal to investigate the data acquisition pipeline.

These case studies highlight the framework's ability to produce both high-level summaries and fine-grained explanations. The hierarchical plan ensures that no critical evaluation dimension is omitted, while SHAP attributions prevent the evaluation from becoming a black box itself. In both domains, the system can be used during development to identify weaknesses or during certification to provide evidence of compliance.

We also conducted a sensitivity analysis on computational cost. For a typical autonomous vehicle evaluation with fifty leaf subgoals, running KernelSHAP with one hundred background samples per leaf required approximately two hundred thousand LLM calls. This is prohibitive for real-time use. However, by reducing SHAP computation to the ten most uncertain subgoals (identified via the LLM's confidence scores), the cost dropped to forty thousand calls, a fivefold reduction. The attribution quality for the omitted subgoals was measured by computing Spearman rank correlation between full and approximated SHAP values on a holdout set; the correlation coefficient was 0.87, indicating acceptable fidelity. This suggests that selective attribution is a viable practical strategy.

6. Governance and Policy Implications

The integration of explainable AI agents into the evaluation lifecycle of autonomous systems carries profound implications for governance, accountability, and regulatory oversight. Traditional oversight mechanisms rely on manual expert review, which is time-consuming and does not scale to the multitude of autonomous systems being deployed. Automated evaluators, if transparent, can assist regulators in continuous monitoring without overwhelming human capacity.

The hierarchical plan produced by the evaluator agent serves as an audit trail. Each node in the plan corresponds to a specific evaluation action with a timestamp, input data, and output attribution. This structure aligns with requirements from emerging AI governance frameworks, such as the European Union's AI Act, which mandates documentation of algorithmic decision-making processes [14]. The SHAP values provide concrete evidence of feature influence, which can be used to test for compliance with non-discrimination rules. For example, if a fairness subgoal yields SHAP values indicating that patient age contributes significantly to the evaluation score, regulators can investigate whether age is a legitimate factor or a proxy for discrimination.

Another governance dimension is the tension between transparency and proprietary protection. System developers may be reluctant to expose full attribution details because they reveal trade secrets or because they fear that explanations can be gamed. The proposed framework allows for selective disclosure: the planner and SHAP outputs can be aggregated to different levels of detail for different stakeholders. A developer might see the full plan and leaf-level attributions, while a third-party auditor might only see high-level subgoal summaries and aggregated feature importance. This layered transparency can satisfy regulatory requirements without compromising intellectual property.

Fairness is a central concern in autonomous system evaluation. SHAP-based attributions can help detect disparate impact. If, for a given evaluation subgoal, SHAP values show that a

protected attribute (e.g., race or gender) has a large influence on the evaluator's judgment, that could indicate bias either in the target system or in the evaluator itself. The hierarchical planner can incorporate fairness constraints as explicit subgoals, such as "compute subgroup performance" and "measure demographic parity." The combined output allows developers and policymakers to trace fairness issues back to specific features and plan nodes, enabling targeted mitigation.

Robustness of the evaluation agent is another policy-relevant issue. Adversarial inputs to the evaluator—for instance, carefully crafted simulation scenarios that trigger erroneous LLM responses—could undermine the trustworthiness of the evaluation. SHAP values can help identify which features in the evaluation input are most sensitive, potentially alerting auditors to manipulation. However, the evaluator itself must be secured against attacks. Integrating robustness checks as subgoals in the planner, such as "test evaluator against adversarial perturbations," could create a self-propagating defense loop. This aligns with emerging standards for AI security [15].

Sustainability of deploying such explainable evaluators at scale depends on computational resources. The carbon footprint of repeated LLM calls and SHAP computations is non-trivial. Policymakers may need to consider trade-offs between explanation depth and environmental cost. The framework can be configured with a "sustainability budget" that limits the total number of model queries per evaluation; the planner then allocates SHAP computations to subgoals with highest impact. This kind of resource-aware governance is an important direction for responsible AI.

Finally, the use of LLM-based evaluators raises the question of accountability when the evaluator itself errs. If an autonomous system passes an evaluation but later causes harm, who is liable—the system developer, the evaluator's operator, or the AI agent provider? The hierarchical plan and SHAP attributions can serve as evidence in legal proceedings, but they do not resolve the fundamental accountability gap. Future governance frameworks must address the meta-level problem of evaluating evaluators, possibly through independent certification bodies that validate the correctness of the evaluator's explanations.

7. Conclusion

This paper has presented a framework for explainable AI agents that integrate SHAP-based decision attribution with hierarchical planning in large language models, enabling transparent evaluation of autonomous systems. The three-layer architecture—hierarchical planner, LLM reasoning engine, and SHAP attribution module—provides a structured approach to decompose complex evaluation tasks while offering both high-level strategic oversight and granular feature-level explanations. We have discussed key integration challenges including granularity trade-offs, surrogate model fidelity, computational overhead, and dynamic planning, and have proposed pragmatic solutions such as selective attribution and resource-aware decomposition.

Case studies from autonomous vehicle safety and clinical decision support illustrated the practical feasibility of the approach, demonstrating how SHAP values can pinpoint influential factors and how hierarchical plans ensure coverage of all relevant evaluation dimensions. Sensitivity analysis showed that selective SHAP computation substantially reduces cost while retaining acceptable explanation quality. Governance implications were examined, highlighting the potential for audit trails, fairness auditing, layered transparency, and sustainability-aware deployment.

The proposed framework advances the state of the art by uniting two previously separate lines of research—explainable AI using SHAP and hierarchical planning with LLMs—into a coherent system for autonomous evaluation. Future work should focus on extending the framework to multi-agent evaluation scenarios where multiple evaluators collaborate or compete, improving the theoretical guarantees of SHAP approximations for generative models, and developing standardized benchmarks for evaluating the evaluators themselves. Additionally, incorporating causal reasoning beyond SHAP, such as counterfactual explanations, could further enhance interpretability. As autonomous systems become more pervasive, the need for trustworthy, explainable, and scalable evaluation agents will only grow. The integration of SHAP and hierarchical planning offers a promising path toward meeting that need.

References

1. Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., ... & Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv preprint arXiv:2303.12712.

2. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems, 30, 4765–4774.

3. Sacerdoti, E. D. (1974). Planning in a hierarchy of abstraction spaces. Artificial Intelligence, 5(2), 115–135.

4. Ribeiro, M. T., Singh, S., & Guestrin, C. (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.

5. Jethani, N., Sudarsan, M., Apicella, A., Sontag, D., & Rajpurkar, P. (2022). FastSHAP: Real-time Shapley value estimation. In International Conference on Learning Representations.

6. Kulkarni, T. D., Narasimhan, K. R., Saeedi, A., & Tenenbaum, J. B. (2016). Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In Advances in Neural Information Processing Systems, 29, 3675–3683.

7. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In Advances in Neural Information Processing Systems, 35, 24824–24837.

8. Dou, Z., Zhao, Q., Wan, Z., Zhang, D., Wang, W., Raiyan, T., ... & Biswas, S. (2025). Plan Then Action: High-Level Planning Guidance Reinforcement Learning for LLM Reasoning. arXiv preprint arXiv:2510.01833.

9. Gao, H., Zeng, W., Zhang, J., & Liang, Y. (2025, December). A large model API response quality prediction model based on least squares vector machine and SHAP interpretability analysis. In 2025 5th International Symposium on Artificial Intelligence and Big Data (AIBDF) (pp. 438-442). IEEE.

10. Shui, Y., Jin, R., Dou, Z., & Gao, Z. (2026). ProtoGuard-SL: Prototype Consistency Based Backdoor Defense for Vertical Split Learning. arXiv preprint arXiv:2604.03595.

11. Zhou, D. (2025, December). M-VP2: Microservice-Oriented Vulnerability Patch Planning-A Cost-Aware Approachusing Multi-Agent Reinforcement Learning. In 2025

5th International Conference on Computer, Internet of Things and Control Engineering (CITCE) (pp. 248-254). IEEE.

12. Jethani, N., Sudarsan, M., Apicella, A., Sontag, D., & Rajpurkar, P. (2022). FastSHAP: Real-time Shapley value estimation. In International Conference on Learning Representations.
13. Molnar, C. (2022). Interpretable machine learning: A guide for making black box models explainable (2nd ed.). Leanpub.
14. European Commission. (2021). Proposal for a regulation laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). COM(2021) 206 final.
15. Goodfellow, I., Papernot, N., McDaniel, P., & Xu, K. (2018). Making machine learning robust against adversarial inputs. Communications of the ACM, 61(7), 56–66.
16. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access, 6, 52138–52160.
17. Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.
18. Lipton, Z. C. (2018). The mythos of model interpretability. Queue, 16(3), 31–57.
19. Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206–215.
20. Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82–115.
21. Holzinger, A., Saranti, A., Molnar, C., Biecek, P., & Samek, W. (2022). Explainable AI methods: A brief overview. In Machine Learning for Health Informatics (pp. 13–30). Springer.
22. Samek, W., Wiegand, T., & Müller, K. R. (2021). Explainable artificial intelligence: Understanding, summarizing and explaining the decisions of deep neural networks. In Explainable AI: Interpreting, Explaining and Visualizing Deep Learning (pp. 5–22). Springer.
23. Pearl, J. (2009). Causality: Models, reasoning, and inference (2nd ed.). Cambridge University Press.
24. Russell, S., & Norvig, P. (2020). Artificial intelligence: A modern approach (4th ed.). Pearson.
25. Shneiderman, B. (2022). Human-centered AI. Oxford University Press.