Governance in Agentic Workflows: Leveraging LLMs as Oversight Agents

Imran Nasim
IBM UK; University of Surrey
imran.nasim@ibm.com, i.nasim@surrey.ac.uk

Abstract

Agentic workflows, composed of interconnected AI agents performing complex tasks, have become extremely popular both in the academic and industrial communities. While such workflows possess significant potential to help processes become more efficient and targeted, they also present risks. As an increasing number of agentic workflows are being deployed in real-world environments, a holistic governance approach is necessary to ensure accountability, safety, and traceability. This paper proposes a *Governance Judge* which can be incorporated to modern day workflows. By addressing practical issues such as performance monitoring, failure detection, and compliance auditing, this approach helps ensure robust oversight for agentic workflows in dynamic operational contexts.

1 Introduction

Autonomous agents have been widely recognized as a crucial pathway toward achieving artificial general intelligence (AGI), offering the capability to perform tasks through self-directed planning and independent actions, eliminating the need for human oversight. Leveraging recent advancements in Large Language Models (LLMs), these models serve as the central core of the agent, enabling human-like decision-making capabilities [Schick *et al.*, 2024; Shinn *et al.*, 2024]. While the promise of agentic workflows is significant, their widespread adoption and deployment in real-world environments raise concerns from a governance and regulatory perspective. To fully harness the benefits of agentic AI systems, it is crucial to ensure their safety by addressing potential vulnerabilities using a holistic governance framework [Chan *et al.*, 2023]. For deployed systems, one of the greatest challenges is implementing automated monitoring, as human evaluation becomes impractical and unsustainable for large-scale frameworks. In this study, we propose an automated monitoring framework leveraging a *Governance Judge*. This framework aims to provide actionable insights for system deployers and facilitate appropriate interventions. The central question addressed is: *How can a judge system be used to govern agentic workflow outcomes in a way that ensures safety, accountability, and operational effectiveness?*

2 Automated Monitoring of Agentic Workflows

Human users often lack the time or capacity to review agent activity logs at the required speed or scale. To address this, a secondary "monitoring" AI system can be deployed to automatically review the primary agentic system's reasoning and actions, ensuring alignment with user goals. A popular approach within the LLM community is the *LLM as a Judge* framework, which has been utilized across various industries for the automated evaluation of LLM outputs [Zheng *et al.*, 2023]. These monitoring systems are highly scalable, operating at significantly higher throughput compared to manual human evaluations. Furthermore, such frameworks can be prompted to generate their own chain-of-thought reasoning, providing transparent logic for their automated assessments [Saunders *et al.*, 2022]. However, these monitoring systems also pose challenges, particularly in terms of cost and bias. Employing smaller AI models for monitoring can reduce costs, but this approach introduces a risk that the smaller model may fail to detect certain misbehaviors of the primary system, potentially undermining the safety and reliability of the workflow. Additionally, recent studies suggests that relying on a single LLM as an evaluator can introduce systematic biases, while employing different LLMs or ensemble methods may lead to more robust and fair evaluations [Zheng *et al.*, 2023; Wang *et al.*, 2023]. To overcome these limitations and extend automated monitoring to broader agentic workflows, we propose the Governance Judge framework. This framework enhances existing approaches by introducing advanced capabilities for fairness, transparency, and decision-making across complex and interconnected systems.

3 Governance Judge Framework

3.1 Core Components

The Governance Judge framework is built on three primary modules, as shown in Figure 1:

i. Input Aggregation: The Governance Judge aggregates four critical inputs:

- **Initial Query:** The user-submitted query that initiates the agentic workflow.
- **Workflow Outputs:** The final output and intermediate steps produced by the agentic system.

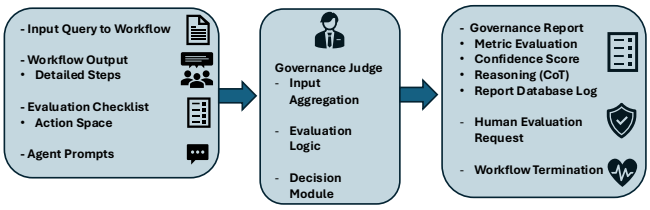

Figure 1: Governance Judge Schematic.

- **Evaluation Checklist:** A predefined set of metrics and criteria that evaluate the workflow's performance and alignment with user goals. The checklist should comprehensively cover the *action space* of the agent, ensuring all possible actions are monitored against expected standards.
- **LLM Prompts:** The instructions or inputs provided to the agents in the workflow. These are evaluated for clarity, alignment with user goals, and safety considerations.

ii. Evaluation Logic: The Governance Judge evaluates aggregated inputs against the provided checklist. This module:

- Produces classification scores for each metric in the evaluation checklist.
- Generates chain-of-thought (CoT) reasoning to provide transparency in its evaluations.
- Identifies discrepancies or risks that deviate from expected performance.

iii. Decision Module: Based on the evaluation results, this module determines the appropriate course of action:

- **Governance Report:** Logs metric evaluations, confidence scores, and CoT reasoning to a database for traceability and drift detection.
- **Human Evaluation Request:** Escalates issues that require manual oversight.
- **Workflow Termination:** Shuts down workflows that violate critical safety or performance thresholds.

3.2 Multi-Judge Systems and Classification Agents

In complex workflows, a single Governance Judge may not suffice. To address this, multiple Governance Judges can be deployed, each specializing in specific aspects of the workflow. These judges can operate independently or collaboratively, providing more granular oversight of the system. Governance Judges can also be tailored for domain-specific needs. For instance, a specialized judge can focus on monitoring adversarial vulnerabilities, ensuring ethical compliance, or evaluating performance against domain-specific metrics, such as financial regulations or healthcare safety standards. By incorporating techniques like Retrieval-Augmented Generation (RAG) systems [Lewis *et al.*, 2020], which leverage external knowledge bases, or fine-tuning on domain-relevant datasets, these domain-specific judges can provide highly targeted and effective evaluations. Furthermore, Governance Judges themselves can be classified as agents in certain scenarios. By classifying these judges as agents, the framework enables modular design where different judges are tailored to specific governance needs, enhancing both efficiency and adaptability.

3.3 Key Advantages

The Governance Judge framework offers several benefits:

- **Scalability:** Automated evaluation processes enable handling workflows of varying complexity and scale.
- **Fairness and Robustness:** Deploying multiple judges or ensemble models helps mitigate biases associated with single-model evaluations.
- **Traceability:** Comprehensive logging ensures all decisions, including prompt evaluations, are auditable, facilitating post-hoc analysis and continuous improvement.
- **Dynamic Monitoring:** The framework can adapt its evaluation criteria to accommodate evolving workflows, unforeseen risks, and changes in agent instructions, such as updates to LLM prompts.
- **Seamless Integration:** The seamless integration of the Governance Judge framework into existing agentic platforms, such as CrewAI and LangGraph, not only minimizes disruption but also reduces development costs and lowers adoption barriers by leveraging pre-existing infrastructures. Additionally, its modular architecture allows for rapid deployment with minimal modifications to existing infrastructures, enhancing its practicality and scalability in real-world applications.
- **Prompt Monitoring:** By incorporating LLM prompt evaluation, the Governance Judge ensures that agent instructions are clear, aligned with user goals, and safe. This allows for proactive detection of potential issues at the source and provides valuable feedback for improving prompt design. Additionally, prompt monitoring can serve as a feedback loop, offering insights to refine and optimize the prompts used by the primary agentic system, thereby enhancing its overall performance and alignment.

Conclusion

This paper proposes the *Governance Judge* framework as a solution for the automated monitoring of agentic workflows. The framework tackles key challenges such as scalability, fairness, and traceability through its modular design, which supports multiple judges and allows for domain-specific customization using techniques like Retrieval-Augmented Generation (RAG) and fine-tuning. By integrating LLM prompt monitoring, the framework enhances safety and transparency, enabling the identification and resolution of root causes of misalignment. Furthermore, its seamless compatibility with platforms like CrewAI and LangGraph ensures minimal disruption while providing comprehensive logging for traceability and drift detection. Future work could explore enhancing the Governance Judge framework through adaptive evaluation criteria that evolve with workflow dynamics, as well as improving the interpretability of LLM-based reasoning to build greater trust and transparency in its decision-making processes.

References

[Chan *et al.*, 2023] Alan Chan, Rebecca Salganik, Alva Markelius, Chris Pang, Nitarshan Rajkumar, Dmitrii Krasheninnikov, Lauro Langosco, Zhonghao He, Yawen Duan, Micah Carroll, et al. Harms from increasingly agentic algorithmic systems. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, pages 651–666, 2023.

[Lewis *et al.*, 2020] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[Saunders *et al.*, 2022] William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022.

[Schick *et al.*, 2024] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024.

[Shinn *et al.*, 2024] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[Wang *et al.*, 2023] Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*, 2023.

[Zheng *et al.*, 2023] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.