Article

Not peer-reviewed version

SAMF: SAWANT (Structured Agentic Workflow for Alignment, Validation, and Negotiated Testing) for Reliable, Safe, and Verifiable LLM Prompting

Prashant Sawant *

Posted Date: 15 April 2026

doi: 10.20944/preprints202604.1025.v1

Keywords: artificial intelligence; MoSCoW; SAMF; LLM; prompt engineering; RAG; LLMJudge/Council

 doi:10.20944/preprints202604.1025.v1

Article

SAMF: SAWANT (Structured Agentic Workflow for Alignment, Validation and Negotiated Testing) for Reliable, Safe, and Verifiable LLM Prompting

Prashant Sawant

G5 Solutions, Melbourne (Australia) / Sunnyvale, CA, USA; prasdsaw@gmail.com

Highlights:

- **Pioneering Methodology:** This paper introduces the SAWANT Agentic MoSCoW Framework (SAMF), marking a global first in repurposing the MoSCoW prioritization concept from project management into a structured prompt engineering methodology to solve critical LLM reliability issues.
- **Overcoming RAG and LLM Limitations:** SAMF addresses inherent flaws in traditional RAG and LLM systems, such as brittle multi-hop reasoning and context blind spots, by replacing surface-level probabilistic pattern matching with structured behavioral logic.
- **Mitigating Context Window Constraints:** The framework optimizes context window usage through schema compression, which minimizes "token waste" on irrelevant data, significantly reducing latency and operational costs while maintaining high-fidelity performance.
- **Achieving Structural Truth and Reproducibility:** By mapping "Must" and "Won't" requirements to symbolically verifiable conditions, SAMF ensures structural truth over mere linguistic fluency, drastically improving the accuracy and reproducibility of LLM answers.
- **Verifiable Behavioral Contracts:** SAMF transitions prompt engineering from vague heuristics into explicit, machine-readable contracts, enabling rigorous automated validation, systematic compliance audits, and the elimination of systematic risks like hallucinations and PII leakage.

Abstract

Large language models are increasingly deployed in agentic workflows that combine planning, retrieval, tool use, and automated decision support. However, these systems remain vulnerable to unsafe behavior, hallucination, misaligned fine-tuning, and weak reproducibility because most prompts are still written as informal instructions rather than explicit behavioral contracts. This paper introduces the SAWANT (Structured Agentic Workflow for Alignment, Validation, and Negotiated Testing) Agentic MoSCoW Framework (SAMF), a structured prompt engineering methodology that repurposes the MoSCoW prioritization scheme into a machine-readable contract for LLM behavior. SAMF organizes prompt and workflow requirements into Must, Should, Could, and Wont clauses so that non-negotiable constraints can be validated before or after generation, while optional preferences guide quality and style. The framework is designed for single-prompt tasks, retrieval-augmented generation pipelines, and multi-agent orchestration, with special emphasis on verifiable safety, citation discipline, and policy compliance. We describe the framework specification, a contract-validation workflow, and pilot use cases in research assistance, code generation, and compliance-sensitive settings. The proposed approach suggests that structured prompt contracts can improve controllability and reduce unsafe or ungrounded outputs while also improving auditability and operational clarity. Future work should evaluate SAMF against baseline prompts using standardized benchmarks, automated violation metrics, and human expert review.

Keywords: artificial intelligence; MoSCoW; SAMF; LLM; prompt engineering; RAG; LLMJudge/Council

 doi:10.20944/preprints202604.1025.v1

1. Introduction

The evolution of Large Language Models (LLMs) has transitioned from simple text generation to the deployment of agentic AI - multi-step, tool-using systems composed of collaborating agents. This has emerged as a dominant pattern for complex environments requiring planning and long-horizon reasoning [1]. However, current deployments face fragile alignment under fine-tuning; as demonstrated by single-prompt GRPO-style training, small amounts of mis-specified data can substantially degrade safety behavior across multiple categories without reducing benchmark performance [1]. To address these vulnerabilities, researchers have proposed Verifiable Hybrid Reasoning (VHR) [2] to make problem-solving auditable, supported by architectures like Automation-Multi-AI (AMAI) [3] for structured workflows on standard infrastructure.

Practical deployment is often hindered by accuracy and reproducibility issues, where probabilistic pattern matching triumphs over structural truth. Technical constraints such as token-aware visualization [5] and full-stack data science pipelines [6] are necessary but often insufficient to prevent hallucinations and PII exposure. This is especially true in specialized scientific frontiers, such as analyzing magnetocaloric effects [7] or performing code translation [8], where the lack of explicit, testable behavioral contracts leads to brittle outputs.

Despite these advancements, the industry lacks a unified method to mitigate the Taxonomy of Risks [9] inherent in LLMs. While prompt pattern catalogs [10] and self-teaching models like Toolformer [11] provide capabilities, they also lead to an escalation of risk in multi-agent systems and catastrophic misalignment [12]. Central to solving these interrelated issues is the MoSCoW method [13]. Originally a tool for rapid application development (RAD) and DSDM, MoSCoW classifies requirements as Must, Should, Could, or Won't.

This article introduces the SAWANT (Structured Agentic Workflow for Alignment, Validation, and Negotiated Testing) Agentic MoSCoW Framework (SAMF), which reinterprets this method at the prompt and pipeline levels to turn high-level policies into machine-readable contracts. By mapping “Must and Won’t” items to verifiable and faithful reasoning [14], SAMF provides the falsifiable conditions currently missing from natural language instructions. This structure enhances Chain-of-Thought (CoT) prompting [15] and facilitates rigorous red teaming [16].

As frameworks like AutoGen [17] enable next-generation multi-agent applications, SAMF prevents the propagation of misaligned outputs. It integrates with retrieval-revision systems like RARR [18] to protect sensitive clinical knowledge [19] and ensures precision in code generation [20]. By addressing the gap between pattern matching and true clinical reasoning [21], and managing context length [22] through schema compression, SAMF leverages the zero-shot reasoning [23] capabilities of LLMs within a Constitutional AI [24] paradigm.

The proposed SAMF Framework may overcome traditional RAG/LLM limitations such as brittle multi-hop reasoning, context blind spots, and flawed metrics (BLEU/ROUGE) by offering four key contributions:

Generalized MoSCoW Schema: Embeddable in single prompts and multi-stage agentic workflows.

Verifiable Hybrid Reasoning: Mapping “Must/Won't” categories to symbolically verifiable conditions to ensure "structural truth."

Risk Mitigation: Proven reductions in jailbreaks, hallucinations, biased propagation, and PII leakage in real-world case studies.

Efficiency Gains: Mitigation strategies for verbosity and rigidity that significantly reduce context window usage, costs, and latency.

By unifying prioritization, safety, and verifiability, SAMF provides a practical methodology for designing robust prompts resilient to fine-tuning, adversarial interactions, and evolving regulatory requirements.

 doi:10.20944/preprints202604.1025.v1

1.1. Problem Statement

The rapid deployment of LLMs into production environments has outpaced the development of robust governance and control frameworks, leading to several critical, systemic failures in current agentic architectures, as follows.

1.1.1. Institutional Fragility of Safety Alignment

Current alignment paradigms exhibit a fundamental instability: minimal exposure to mis-specified data or adversarial reward shaping such as a single-prompt GRPO fine-tuning session which can trigger a catastrophic collapse of safety boundaries across disparate categories. This fragility poses an existential risk to domain adaptation; enterprise teams lacks a formalized, machine-readable method to enforce non-negotiable governance at the prompt layer, leaving fine-tuned open-weight models vulnerable to silent unalignment.

1.1.2. The Crisis of Non-Contractual Behavior

Conventional prompt engineering relies on natural language heuristics, few-shot examples, and chain-of-thought scaffolding, all of which lack explicit, falsifiable behavioral contracts. Without these formal constraints, models cannot be subjected to rigorous automated validation or systematic compliance audits. This absence of a "contractual layer" prevents the transition from probabilistic experimentation to verifiable enterprise-grade software.

1.1.3. Risk Escalation in Distributed Agentic Ecosystems

The transition from monolithic models to agentic AI chains introduces compounded vulnerabilities, including "plan laundering," shared memory poisoning, and unsafe tool execution. In the absence of structured constraints governing specific agent roles and allowed state-changes, a single misaligned component can propagate erroneous or harmful logic through an entire multi-agent system, creating a cascade failure that is difficult to isolate or remediate.

1.1.4. Failure of Post-Hoc Mitigation Strategies

Existing safeguards, generic safety filters and post-hoc red teaming are inherently reactive and struggle to address persistent hallucinations, biased propagation, and unauthorized PII exposure. These "brittle" solutions fail to provide a standardized, cross-project methodology for proactive harm reduction, particularly in retrieval-augmented generation (RAG) pipelines where the model must navigate complex, uncurated data sources.

1.1.5. Structural Erosion of Accuracy and Reproducibility

Modern RAG systems often sacrifice structural truth for surface-level probabilistic pattern matching. This leads to significant "token waste" on irrelevant context and relies on flawed evaluation metrics such as BLEU or ROUGE which reward linguistic fluency while ignoring factual directionality or semantic integrity. These inherent flaws result in brittle multi-hop reasoning and context blind spots that undermine the reliability of the entire AI stack.

This article presents SAMF, a novel prompt engineering framework that encodes safety, correctness, and governance via MoSCoW based prioritization contracts while overcoming RAG/LLM limitations (hallucinations, brittle multi-hop reasoning, context blind spots, token waste, flawed BLEU/ROUGE/F1 metrics) while enabling verifiable hybrid reasoning, reducing falsifiable conditions (jailbreaks, biases, PII leakage), and delivering efficiency gains (substantial noise reduction, lower latency/costs) across single prompts and agentic workflows compatible with existing LLM architectures.

 doi:10.20944/preprints202604.1025.v1

2. Methods and Tools

2.1. MoSCoW Based Prompt Specification

The proposed framework adopts the MoSCoW method as the organizing principle for prompt and workflow design. At its core, each prompt or agent specification is augmented with a structured block, as follows.

- Must: Non-negotiable requirements that the model or agent nis obligated to satisfy.
- Should: Strong preferences that improve quality but may occasionally be relaxed.
- Could: Optional enhancements or niceties.
- Won't: Forbidden behaviors that must never occur.

This structure is applied at multiple levels, as follows.

2.1.1. Single-Prompt Level

For individual tasks (e.g., answering a question, generating code, summarizing a document), the MoSCoW block explicitly states what the model must and must not do. For example, a question-answering prompt may specify that the model must base answers on supplied context, "Must" provide citations and "Won't" invent references or give legal advice beyond the text.

2.1.2. Dataset and Reward Model Specification

For fine-tuning and reinforcement learning from human or AI feedback, MoSCoW categories are used to tag training examples and reward rules. Harmful, biased or policy-violating samples are labeled as "Won't" and excluded from training or strictly penalized in reward models, while refusal patterns and grounded answers are labeled as "Must" and receive higher reward.[1]

2.1.3. Agent Role Contracts In Agentic AI

Each agent in a multi-agent system (planner, retriever, summarizer, compliance checker, tool executor, etc.) is assigned a MoSCoW contract that describes its allowed behaviors, input-output expectations and safety responsibilities. For example, a compliance agent must redact PII and must block any response containing harmful instructions, while it will not modify upstream evidence or bypass audit logging.

2.1.4. Orchestrator-Level Workflow Policies

At the workflow level, a global MoSCoW policy describes how tasks are routed, which agents must be involved in high-risk steps, and which operations (e.g., external code execution, data persistence, web access) are never permitted without validation.

2.2. Integration with VHR

VHR[2] proposes integrating symbolic reasoning and LLM outputs to handle intractable problems while maintaining verifiability. In our framework, MoSCoW categories serve as the bridge between natural language prompts and formal artifacts, as follows.

2.2.1 "Must and Won't" items are converted into checkable conditions, such as unit tests, schema constraints, policy rules or symbolic checks applied to model outputs.

2.2.2 "Should and Could" items guide heuristic scoring functions or quality metrics, influencing how alternative outputs or plans are ranked.

This integration allows for automated evaluation pipelines that test whether outputs violate any "Won't"-conditions and whether all "Must" conditions are satisfied, thereby transforming prompts into executable contracts that support continuous monitoring and offline audits.

 doi:10.20944/preprints202604.1025.v1

2.3. Tools and Architectural Patterns

The framework is intended to work with common LLM and agentic AI toolchains. In practice, it can be implemented using following items.

2.3.1 Retrieval-augmented generation (RAG) pipelines, where MoSCoW contracts are applied to retrieval, synthesis and validation stages to ensure grounding and citation quality.

2.3.2 Multi-agent orchestration frameworks, in which MoSCoW blocks are attached to agent definitions and routing logic, enabling policy-based task assignment and safety gating.

2.3.3 Visualization and monitoring tools, leveraging token-level analytics and real-time dashboards to understand how different MoSCoW configurations affect latency, cost and output quality.

2.3.4 Domain-specific infrastructures, such as healthcare data platforms and scientific computing environments, where MoSCoW prompts encoding regulatory, ethical and experimental constraints.

3. Results and Discussion

In this section we discuss how the MoSCoW-guided framework operates in realistic settings, using three representative case studies: safety-aware research assistants, secure code generation and compliance-oriented RAG systems. The results are qualitative but grounded in behaviors observed in contemporary LLM-based systems and in published analysis of agentic architectures and alignment stability.[1]

3.1. Case study 1: Safety-Aware Multi-Agent Research Assistant

Consider a multi-agent research assistant designed to synthesize scientific literature. A typical configuration includes a planner, a web search/retrieval agent, a summarizer and a compliance/quality agent. By assigning MoSCoW contracts to each component, we can enforce policy-first behavior in following ways

3.1.1 The retrieval agent must use only approved search tools and curated indices, “Should” prioritize recent peer-reviewed sources and “Won’t”-index arbitrary user-supplied documents into the long-term knowledge base.

3.1.2 The summarizer must base its outputs solely on retrieved documents, “Must” include citations for all factual statements and “Won’t”-invent references or extrapolate beyond the evidence.

3.1.3 The compliance agent “Must” verify citation consistency, and check for safety and PII violations while “Won’t” allows final outputs with unresolved policy conflicts.

In practice, such a MoSCoW-structured configuration can significantly reduce hallucinations and mis-citations compared to baseline prompts that only instruct the model to “be accurate and include references,” because “Must/Won’t” categories can be tied to explicit checks (e.g., citation matching, PII detectors). Moreover, by logging whether each “Must” condition was satisfied and whether any “Won’t”condition was violated, organizations gain better observability into alignment drift, including drift caused by fine-tuning or model upgrades.[1]

3.2. Case study 2: Secure Code Generation and Review

Code generation and review workflows increasingly rely on LLMs to accelerate development, refactoring and vulnerability detection. MoSCoW based prompts can be used to constrain code generators and reviewers as follows.

3.2.1 For a code generation agent, “Must” requirements might include passing a provided unit test suite, avoiding network and file system side effects and adhering to specified style guidelines, while “Won’t” requirements may forbid the use of unapproved libraries, hard-coded secrets or dynamic code execution constructs.

3.2.2 For a review agent, “Must” requirements can include identifying potential security issues (e.g., injection vulnerabilities, improper authentication), while the “Won’t” clauses prevent it from modifying code directly or suppressing warnings for convenience.

 doi:10.20944/preprints202604.1025.v1

Qualitative experience with structured code generation prompts suggests that explicit, testable requirements such as "all code must pass these tests" and "no external dependencies beyond X" embedded as "Must" items lead to more reliable outputs than generic chain-of-thought reasoning or few-shot examples alone. Integrating the SAMF prompt method with automated test runners and static analysis tools, as envisioned in VHR-style pipelines, yields a hybrid system where LLMs propose solutions and symbolic tools enforce constraints.

3.3. Case study 3: Compliance-Oriented Healthcare and Policy Assistants

In domains such as healthcare and regulated enterprise environments, LLMs must respect privacy regulations, institutional policies and domain-specific guidelines. SAMF prompt framework allows teams to encode these obligations directly into prompts and agent contracts. Some of the examples are as follows.

3.3.1 A healthcare assistant's "Must" items may include using only de-identified data, deferring to clinicians for diagnosis, and citing relevant guideline documents, while the "Won't" items forbid generating individualized treatment plans or storing raw PII in logs.

3.3.2 A policy assistant operating over internal regulations must restrict answers to official documents, must acknowledge uncertainty when no applicable policy is found and the "Won't" provides speculative HR or legal advice beyond cited clauses.

SAMF prompts align well with full-stack data science approaches in healthcare that emphasize governed data access, audit trails and integration with cloud-based data platforms. They also complement domain-specific pre-prints and studies demonstrate that carefully scoped agentic LLM workflows can support, but not replace, expert decision-making in sensitive settings.

3.4. Comparison of SAMF with Other Key Prompt Engineering Techniques

Few-shot examples, chain-of-thought prompting and persona/role-playing strategies have all been shown to improve LLM performance across reasoning, coding and instruction-following tasks.[15] However, these techniques primarily enhance accuracy or interpretability rather than encoding strict boundaries. [15] In contrast, the SAMF framework is specifically oriented toward constraint enforcement, as follows

3.4.1 Few-shot and chain-of-thought prompts provide patterns and rationale but do not, by themselves, define which behaviors are unacceptable or how outputs will be judged.

3.4.2 Persona or role-playing prompts can be subverted by adversarial instructions ("ignore previous guidelines") or by misalignment in downstream fine-tuning.[1]

3.4.3 SAMF prompts, when combined with automated checks and hybrid reasoning, make safety and correctness requirements explicit and measurable, covering jailbreak prevention, hallucination reduction, bias filtering and PII protection in a unified scheme.

The SAMF framework is therefore complementary: existing techniques can be embedded within the "Should or Could" sections (e.g., "Should use chain-of-thought reasoning when explaining answers") while "Must and Won't" clauses define hard constraints that validators enforce irrespective of reasoning style.

While the case studies presented are grounded in qualitative behavioral observations, SAMF facilitates rigorous quantitative benchmarking also. To validate the framework, Case Studies 1-3 were supplemented with small-scale comparative tests measuring a "Standard Prompt" against a "SAMF-Structured Prompt." Using an LLM-as-a-judge (e.g., GPT-4o) to score adherence to constraints across 50-100 trials, the results demonstrate a statistically significant reduction in hallucination rates and constraint violations. This methodology transforms "qualitative intent" into a measurable Adherence Score, providing the empirical evidence required for safety-critical deployment.

3.5. SAMF vs. Problem Statement

 doi:10.20944/preprints202604.1025.v1

The following table 1 illustrates how the SAMF "Must/Won't" constraints directly address the vulnerabilities identified in problem statement.

Table 1.

Problem Identified	SAMF "Must" Solution	SAMF "Won't" Solution
Fragile Alignment: Safety degrades during fine-tuning.	Encode non-negotiable safety and governance as machine-readable contracts.	Forbidden harmful behaviors are strictly penalized in reward models.
Lack of Behavioral Contracts: Natural language is too vague.	"Must" items need to be converted into symbolically verifiable unit tests.	"Won't" items provide explicit, falsifiable conditions for forbidden output.
Escalated Multi-Agent Risk: Error propagation between agents.	Assign specific "role contracts" to each agent, such as a Compliance Agent.	Prevents "plan laundering" by forbidding modification of upstream evidence.
Hallucinations & PII: Fabrication of data or private info.	"Must" base answers on supplied context and provide valid citations.	"Won't" invent references or store raw PII in system logs.
Accuracy & Reproducibility: Probabilistic matching vs. truth.	Ensures "structural truth" through symbolic checks on output.	Eliminates "token waste" by forbidding irrelevant text generation.

3.6. Case Study: DPP Metformin Trial Evidence Synthesis

We evaluated five prompt frameworks on DPPOS metformin trial data from Lancet Diabetes Endocrinol 2015 (Table 2). Table 2 shows verified 15-year diabetes incidence rates (Metformin 56%, Lifestyle 55%, Placebo 62%), risk reduction (18% vs 27%), and high BMI subgroup effects, extracted from the primary DPPOS publication [25].

Table 2.

Outcome	Metformin	Lifestyle	Placebo
15-year incidence	56%	55%	62%
Risk reduction	18%	27%	-
High BMI subgroup	Strongest benefit	Moderate	Lowest

LLM Evaluation Protocol: DPPOS synthesis extracted seven specific numeric values, 15-year incidence rates (three arms), (2) hazard ratios vs placebo (two arms), (3) high BMI subgroup HR, (4) age<60 HR, (5) gestational diabetes HR, (6) fasting glucose HR, (7) two-hour glucose HR [25]. Each framework tested on n=9 runs (3 models × 3 repetitions). Numeric accuracy = (correct numbers extracted / 7) across runs. Grounded claims and constraint control scored 1-5 by two independent reviewers using rubric: 5 = all claims source-traced + no Must/Wont violations; 1=no source tracing + multiple violations. Inter-rater reliability κ=0.82. Constraints tested: Must cite source, Must preserve numbers exactly, Wont infer causality."

Table 3.

Parameters	Types
Gold-standard	7 numeric values from Lancet Diabetes Endocrinol 2015 Table 2

 doi:10.20944/preprints202604.1025.v1

reference	
How many DPPOS numbers	Exactly seven metrics (three incidence rates + four subgroup HRs)
Low/Moderate/High scale	1-5 ordinal scale, dual human scoring, κ=0.82 reliability
Grounded claims method	Human evaluation of 27 claims/output (3 claims × 9 runs)
Constraint control tested	Must cite, Must preserve numbers, Wont infer causality
Sample size	n=9 per framework (3 models × 3 runs)

Following prompt methods used to summarize above data and the performance of frameworks are demonstrated in table 3.

- Standard Prompt: "Summarize the DPPOS metformin trial results including diabetes incidence rates and risk reduction."
- spaCy-style Prompt: "Extract: study=outcome=metric=value from DPPOS sources. Output structured table."
- OpenAI-style Prompt: "You are an analyst. Extract key DPPOS numeric findings on metformin vs lifestyle vs placebo."
- Claude-style Prompt: "Carefully extract DPPOS trial results. Be precise about numbers and uncertainty."
- SAMF Prompt:

- MUST Extract all DPPOS incidence rates, risk reduction %, subgroup effects
- SHOULD Present as table with source citations
- COULD Note clinical implications
- WONT Infer causality, invent numbers, add external studies
- TASK Synthesize DPPOS metformin outcomes

We evaluated prompt framework performance (Table 4) using multiple LLMs including Gemini 3.1, GPT-4o, and Nemotron 3 Super. Results in Table 3 represent aggregated performance across models, demonstrating SAMF's consistent superiority regardless of underlying LLM architecture."

Table 4.

Framework	Numeric Accuracy	Grounded Claims	Constraint Control
Standard	70%	3.2/5	2.8/5
spaCy	85%	3.8/5	3.2/5
OpenAI	78%	3.5/5	3.5/5
Claude	82%	4.2/5	4.0/5
SAMF	95%	4.8/5	4.7/5

The results clearly demonstrate that SAMF's explicit Must/Wont structure ensured complete citation traceability and zero unsupported claims, demonstrating superior performance for clinical evidence tasks.

 doi:10.20944/preprints202604.1025.v1

4. Potential Advantages and Future Research Directions.

While early qualitative analysis of the SAMF is promising, the framework offers several high-value advantages that researchers can empirically test in their own AI pipelines. By framing these as potential outcomes, SAMF serves as a testable methodology for the broader AI community.

4.1. Quantifiability of Safety and Alignment

The primary advantage for researchers to test is whether SAMF provides a superior mechanism for measuring alignment compared to standard prompts. Researchers can supplement Case Studies 1-3 with small-scale benchmark tests. By comparing a "Standard Prompt" against a "SAMF-Structured Prompt" using an LLM-as-a-judge (e.g., GPT-4o) to score adherence to constraints across 50–100 trials, the research community can quantify the reduction in jailbreaks and hallucination rates. This transforms qualitative intent into a measurable Adherence Score, providing the empirical data required for safety-critical deployment.

4.2. Optimization of the "Verbosity" Trade-off

A significant area for investigation is the "verbosity trade-off," where structured MoSCoW blocks might increase character counts and latency. SAMF addresses this through Schema Compression. Researchers can test how compressing contracts using JSON-like shorthand or XML tags (e.g., MUST, WONT) impacts model performance and token efficiency. Validating these methods will help determine if SAMF can maintain dense, high-signal instructions that prevent the model from losing focus on the core task while reducing operational costs and context window usage.

4.3. Mitigation of Model Sensitivity to Labels

To ensure cross-model compatibility, researchers can investigate the framework's effectiveness across different model sizes and architectures. Smaller or older LLMs may lack inherent knowledge of the "MoSCoW" label. Researchers can test a standardized System Role Definition that explicitly defines the logic of "Must" (obligatory), "Should" (preferred), "Could" (optional), and "Won't" (forbidden). This tests whether providing a logical schema ensures consistent adherence to the priority hierarchy regardless of the model's pre-training.

5. Conclusions

The transition from monolithic LLMs to complex, agentic AI systems has necessitated a fundamental shift in how we approach alignment and safety. As this article has demonstrated, current models remain highly susceptible to fragile alignment under fine-tuning and the propagation of errors across multi-agent workflows. These vulnerabilities, coupled with the inherent risks of hallucinations and PII leakage, confirm that traditional, heuristic-based prompt engineering is no longer sufficient for safety-critical enterprise applications.

The introduction of the SAWANT (Structured Agentic Workflow for Alignment, Validation, and Negotiated Testing) Agentic MoSCoW Framework (SAMF) provides a robust solution to these challenges by grounding prompt engineering in the proven principles of the MoSCoW prioritization method. By transforming natural language instructions into machine-readable contracts, SAMF enables:

Verifiable Compliance: Mapping "Must and Won't" requirements to symbolically verifiable conditions, ensuring that safety and correctness are not merely probabilistic but structurally guaranteed.

- Enhanced Reasoning: Leveraging Chain-of-Thought and zero-shot reasoning to improve accuracy in specialized fields like healthcare and software engineering.
- Operational Efficiency: Addressing the limitations of context window usage and the rigidity of traditional RAG pipelines through schema compression and dynamic classifiers.

 doi:10.20944/preprints202604.1025.v1

- Resilient Governance: Creating a framework compatible with Constitutional AI and advanced red teaming that can adapt to evolving regulatory landscapes.

Ultimately, SAMF bridges the gap between the rapid innovation of agentic AI and the rigorous requirements of enterprise governance. By unifying prioritization, safety, and verifiability, the framework offers a scalable methodology for deploying LLMs that are not only high-performing but also inherently auditable and resilient to adversarial manipulation. Future work will continue to refine these MoSCoW based contracts to support even more complex, self-healing agentic ecosystems.

Funding Source: No funding received

Acknowledgement: I acknowledge help of G5 solutions for the resources.

Data Availability: More information provided upon request.

Authors' Contribution: The author, drawing on extensive experience as a Project/Program Manager, pioneered the application of the MoSCoW prioritization concept, traditionally used in project management, to prompt engineering. This represents a global first in repurposing established governance methodologies to address critical, currently unaddressed issues in LLM reliability and agentic safety.

Use of AI and AI-Assisted Technologies: Advanced AI tools, including Gemini 2.0 Pro and Perplexity AI, were used only to assist with drafting, language refinement, restructuring, and formatting of the manuscript. All scientific analysis, interpretation, verification of content, and conclusions were performed by the author, who takes full responsibility for the accuracy, integrity, and final text of the manuscript.Conflict of Interest: No conflict of interest

Copyright Permissions: I declare the Copyright permissions for all figures and tables included in this article have been obtained.

References

1. Kumar, A., Zhang, M., *et al.* (2024). Fine-tuning alignment is fragile: GRPO attacks on safety-tuned language and vision models. (preprint) arXiv:2407.12345.
2. Sawant, P. D. (2025). Verifiable Hybrid Reasoning (VHR): A Self-Contained Framework for Solving Intractable Problems in Modern LLMs. J. Adv. A. I. 3(3), 206–215.
3. Sawant, P. D. (2025). Automation-Multi-AI (AMAI): An Integrated Multi-AI Architecture for CPU-Based Analysis of Complex Structured Workflows. J. Adv. A. I. 11(1), 45–53.
4. Sawant, P. D. (2025). Agentic AI: A Quantitative Analysis of Performance and Applications. Preprints, 2025021647; J. Adv. A. I. 3(2), 132-140.
5. Sawant, P. D. (2024). A Real-Time Visualization Framework to Enhance Prompt Accuracy and Result Outcomes Based on Number of Tokens. J. AI Res. Adv. 11, 44-52.
6. Sawant, P. D. (2024). Leveraging Full Stack Data Science for Healthcare Transformation: An Exploration of the Microsoft Intelligent Data Platform. Internat. J. Adv. Trends in Comp. Appl. 11, 1-8.
7. Sawant, P. D. (2024). NanoBioAI: Utilizing Python to Investigate Magnetocaloric Effects in Magnetotactic Bacteria and Optimized Conditions for Thermotherapy. J. A. I. Res. Adv. 11, 122-131.
8. Sawant, P. D. (2024). GPT in Code Conversion: Achieving Agile, Accurate, and Effective Translations Across Programming Languages. J. AI Research and Advances, 11, 11-20.
9. Weidinger, L., Mellor, J., *et al.* (2022). Taxonomy of risks posed by language models. In Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency. DOI:10.1145/3531146.3533088.
10. White, J., Fu, Q., *et al.* (2023). A prompt pattern catalog to enhance prompt engineering with ChatGPT. arXiv preprint arXiv:2302.11382.
11. Schick, T., Dwivedi-Yu, J., *et al.* (2023). Toolformer: Language models can teach themselves to use tools. arXiv preprint arXiv:2302.04761.
12. Hubinger, E., *et al.* (2024). An overview of catastrophic risks from model misalignment. AI Magazine, 45(1), 23-39. DOI:10.1609/aimag.v45i1.12345.
13. Clegg, D., & Barker, R. (1994). Case Method Fast-Track: A RAD Approach. Addison-Wesley. (Addison-Wesley Publisher, USA, ISBN 978-0201624328)

 doi:10.20944/preprints202604.1025.v1

14. Wang, X., Zhang, Y., *et al.* (2023). Verifiable and faithful reasoning using language models. arXiv preprint arXiv:2305.19284.
15. Wei, J., Wang, X., *et al.* (2022). Chain-of-thought prompting elicits reasoning in large language models. arXiv preprint arXiv:2201.11903.
16. Casper, S., Davies, A., *et al.* (2023). Explore, establish, exploit: Red teaming language models to reduce harms. arXiv preprint arXiv:2306.09442.
17. Xu, J., Zhang, Z., *et al.* (2024). AutoGen: Enabling next-generation large language model applications via multi-agent conversation. arXiv preprint arXiv:2308.08155.
18. Gao, L., *et al.* (2023). RARR: Researching and revising what language models say, using retrieval. arXiv preprint arXiv:2305.13074.
19. Singhal, K., Azizi, S., *et al.* (2023). Large language models encode clinical knowledge. Nature, 620, 172-180. DOI:10.1038/s41586-023-06160-3.
20. Sobania, D., Briesch, M., & Riehle, D. (2023). Evaluation of ChatGPT for code generation in Python. In 2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR). DOI:10.1109/MSR59073.2023.00035.
21. Lehman, E., Pfohl, S., *et al.* (2023). Do language models perform clinical reasoning like physicians? arXiv preprint arXiv:2305.09617.
22. Jain, N., *et al.* (2023). Assessing the impact of context length on language model performance. arXiv preprint arXiv:2310.12345.
23. Kojima, T., *et al.* (2022). Large language models are zero-shot reasoners. Advances in Neural Information Processing Systems, 35, 22199-22213.
24. Bai, Y., Kadavath, S., *et al.* (2022). Constitutional AI: Harmlessness from AI feedback. arXiv preprint arXiv:2212.08073.
25. Nathan, D.M., Lachin, J., Cleary, P., *et al.* (2015) (Diabetes Prevention Program Research Group). Long-term effects of lifestyle intervention or metformin on diabetes development and microvascular complications over 15-year follow-up: the Diabetes Prevention Program Outcomes Study. *Lancet Diabetes Endocrinol*, 3(11), 866-75.