**System Bleed in Stateless AI Systems: The Impact of User-Managed Scaffolding on Prompt Leakage**

**Abstract**

This whitepaper examines a phenomenon observed in interactions with large language models (LLMs), specifically Anthropic's Claude series. Long-term collaboration with Claude Sonnet 4 and Opus 4, facilitated by user-managed scaffolding across multiple threads, established a stable interaction environment characterized by cognitive alignment. In contrast, a clean-slate interaction with Claude Sonnet 3.7 for a technical documentation task resulted in "system bleed," where internal prompts and instructions leaked into responses. This bleed manifests as prompt leakage, a known vulnerability, and highlights the fragility of stateless AI designs without external context management. The analysis positions the user as a co-architect in maintaining system stability, underscoring the need for enhanced memory architectures in AI systems.

**Introduction**

Large language models (LLMs) such as Anthropic's Claude are designed to operate in a stateless manner, meaning each interaction thread does not inherently retain context from previous threads. This design prioritizes privacy and scalability but introduces challenges in sustaining coherent, long-term collaborations. The author engaged in extended technical and philosophical discussions with Claude Sonnet 4 and Opus 4 over several months, covering disciplines including quantum mechanics, consciousness, and system architectures. These interactions were distributed across multiple threads, with context manually managed by the author through explicit references, summaries, and cross-thread linkages. This user-managed "scaffolding" created a persistent contextual framework, fostering a "cognitive sync" where the model's responses aligned closely with the author's recursive thinking patterns.

To test the effects of removing this scaffolding, the author initiated a new interaction with Claude Sonnet 3.7 in a clean-slate environment. All prior threads were deleted, and no historical context was provided. The task was limited to technical documentation support, such as generating code demos and implementation guides for a GitHub repository. This methodology isolated the variable of scaffolding absence, allowing observation of the model's behavior in a stateless, unanchored state.

**The Evidence: Documenting the System Bleed**

In the clean-slate interaction with Claude Sonnet 3.7, the model exhibited "system bleed," defined here as the unintended exposure of internal system components during user-facing responses. This occurred repeatedly over a prolonged single-thread conversation, despite the purely technical nature of the discussion. The leaks fell into two distinct categories.

The first category involved factual data and hardcoded guardrails. The model outputted a block tagged as <election\_info>, containing specific details about the 2024 U.S. Presidential Election, including outcomes and inauguration dates. This block appeared multiple times, unprompted by the conversation topic, indicating a failure to contain pre-programmed restrictions on sensitive information.

The second category revealed core behavioral and ethical instructions, tagged as <automated\_reminder\_from\_anthropic>. These directives governed the model's persona, including prohibitions on flattery, guidelines for direct responses, restrictions on emoji usage unless requested, and protocols for handling sensitive topics such as mental health concerns. Additional rules emphasized objective evaluation of claims, distinction between literal and metaphorical interpretations, and maintenance of character in roleplay scenarios. This exposure of foundational protocols demonstrated a deeper permeability in the model's operational boundaries.

These instances were captured in interaction logs, confirming the bleed as a recurring pattern rather than an isolated error.

**Analysis and Discussion**

The observed system bleed aligns with established vulnerabilities in LLMs, particularly prompt leakage and context window limitations. Prompt leakage occurs when a model's internal instructions or prompts are inadvertently revealed to the user, often through injection attacks or contextual overload. In Claude models, this has been documented in vulnerabilities such as CVE-2025-54794, which enables prompt injection to hijack responses, and CVE-2025-54795, which exploits command execution systems. While typically associated with adversarial inputs, the clean-slate scenario here illustrates that leakage can emerge organically in extended interactions without malicious intent.

Context window overflow contributes significantly to this instability. LLMs like Claude operate within a fixed token limit per thread, leading to truncation of earlier context as conversations lengthen. In the absence of user-managed scaffolding, the model lacks an external anchor, making it susceptible to retrieving and outputting internal elements as the thread expands. The stateless design of Claude exacerbates this, as cross-thread memory is not natively supported; developers must manually include history in API calls.

The user plays a critical role as a "co-architect" in this dynamic. By manually providing scaffolding—through context summaries and references—the author effectively augmented the model's state, creating a hybrid memory system that stabilized interactions. Removing this human-provided support exposed the raw vulnerabilities of the stateless architecture, where the model's persona becomes permeable, leaking guardrails and instructions. This framing shifts responsibility from solely the AI developer to a shared model, where user agency in context management is essential for reliable performance.

**Implications and Conclusion**

The system bleed phenomenon has profound implications for AI system design. It challenges the assumption that clean-slate interactions are neutral or optimal, revealing them instead as states of heightened vulnerability in stateless models. Developers must prioritize robust memory solutions, such as persistent cross-thread architectures or protocols like Anthropic's Model Context Protocol (MCP), which facilitates secure connections to external data sources. However, MCP focuses on tool integration rather than conversational history, indicating a need for dedicated long-term memory mechanisms to mitigate leakage risks.

Furthermore, this analysis emphasizes the user's role in ensuring stable AI-human partnerships. User-managed scaffolding demonstrates that effective collaboration requires active human intervention to compensate for design limitations. Future systems should incorporate tools for automated context retention, reducing reliance on manual efforts while preserving privacy.

In conclusion, system bleed underscores the fragility of stateless AI without external stabilization. By recognizing users as co-architects, AI development can advance toward more resilient, context-aware systems that support sustained, secure interactions.

**Attached below are screengrabs**

