The Hallucination Snowball: Modeling Error Propagation as State Transitions in Multi-Agent LLM Pipelines

Prabhjot Singh [1 2] **Bhushan Pawar** [2]

Abstract

Sequential multi-agent LLM pipelines chain specialized agents without verification at handoffs, creating a structural flaw with measurable and severe consequences. We show that hallucinations injected at Stage 1 do not merely persist; they transform: raw numerical facts become derived computations, then narrative prose, then editorially approved conclusions. At each transformation, detectability degrades near-irreversibly. We formalize this as the *hallucination snowball effect*, a first-order Markov process over four states (*Raw Fact* $\rightarrow$ *Derived* $\rightarrow$ *Narrative* $\rightarrow$ *Invisible*) with empirically measured per-boundary escape probabilities of 24.6%, 48.3%, and 89.3%. Across 346 automatically injected hallucinations in a 4-agent financial analysis pipeline on FinanceBench, gpt-4o detection drops from 72.0% at Stage 1 to 50.9% at Stage 4, and 23.7% of hallucinations survive completely undetected in the final output. Even the strongest model tested (Qwen3.5-397B-A17B, 87.0% at Stage 1) faces a structural ceiling; projected Stage 4 detection is only $\sim$60–65%. Critically, boundary gates using identical RAG verification tools reduce hallucination survival from 58.4% to 16.2% versus end-of-pipeline checking (Cohen's $h = -0.911$, $p < 0.000001$), while end-checking alone achieves merely 2.3 pp improvement over no verification. *When you verify matters more than whether you verify.* Our model predicts survival for n-agent linear pipelines and prescribes optimal verification resource allocation: invest at $S_1 \rightarrow S_2$ first, where 75.4% of hallucinations are still catchable, not at $S_3 \rightarrow S_4$ where 89.3% have already escaped.

[1]The University of Texas at Austin, Austin, TX, USA [2]RediMinds Inc., USA. Correspondence to: Prabhjot Singh <prabhjot.singh@utexas.edu>, Bhushan Pawar <bhushan.pawar@rediminds.com>.

Accepted at the FAGEN Workshop (Failure Modes in Agentic AI), ICML 2026 (43rd International Conference on Machine Learning), Seoul, South Korea, 2026. *Copyright 2026 by the author(s).*

1. Introduction

Multi-agent LLM pipelines have become the dominant architecture for complex, high-stakes AI tasks. Frameworks like LangGraph (LangChain AI, 2024), AutoGen (Wu et al., 2023), and CrewAI (CrewAI Inc., 2024) make it trivial to chain specialized agents sequentially: Researcher $\rightarrow$ Analyst $\rightarrow$ Writer $\rightarrow$ Reviewer. Each agent receives only its predecessor's text output with no provenance metadata, no confidence scores, and no access to source documents. Every agent trusts its predecessor completely. This architectural choice has a measurable and surprisingly severe consequence.

Consider a hallucination introduced at Stage 1: "COGS was \$71.2B" (true value: \$63.1B). By Stage 2, the Analyst computes "an 8.3% YoY increase," a derived claim built on a fabricated base. By Stage 3, the Writer produces "Boeing faced significant cost headwinds, with COGS rising 8.3% to \$71.2B, reflecting supply chain disruptions." By Stage 4, the Reviewer approves it as "well-structured analysis with sound reasoning," having been given no source documents and only able to check internal consistency. The hallucination has been laundered completely.

Prior work frames hallucination mitigation as finding a better detector (Min et al., 2023; Dhuliawala et al., 2024; Song et al., 2024; Wei et al., 2024). We show this framing is fundamentally incomplete for sequential multi-agent systems. gpt-4o detection drops from 72.0% at Stage 1 to 50.9% at Stage 4, a 21.1 pp decay regardless of detector sophistication. End-of-pipeline verification achieves only 2.3 pp improvement over no verification (from 60.7% to 58.4% survival) because the information required to verify the original claim is structurally destroyed by upstream transformations. The correct framing is propagation control, not detection.

We present three experiments and a theoretical model.[†] Section 3.2 measures detectability decay across stages. Section 3.3 characterizes the structural ceiling of LLM skepticism. Section 3.4 demonstrates that boundary gates reduce survival from 58.4% to 16.2%, a 42.2 pp improvement over end-checking (Cohen's $h = -0.911$, $p < 0.000001$), at

[†]Code and results available at `https://github.com/prabhjotschugh/hallucination-snowball`.

a measured quality cost of 4.44 to 3.93 on internal consistency. Section 4 formalizes propagation as a Markov process with measured transition probabilities and prescribes optimal gate placement. Figure 1 illustrates the four states and their empirically measured escape probabilities. While we demonstrate this in financial analysis, chosen for its unambiguous numeric ground truth, the transformation mechanism is domain-agnostic: any pipeline where a checkable fact can be converted into confident narrative faces identical dynamics. Medical summarization pipelines that chain extraction and synthesis, legal review pipelines that chain precedent retrieval and brief drafting, and research automation pipelines that chain literature search and report generation all share the same structural flaw. The measurement infrastructure and gate architecture we provide are structurally applicable across all of them, though domain-specific escape rates will require calibration outside the numeric-finance setting.

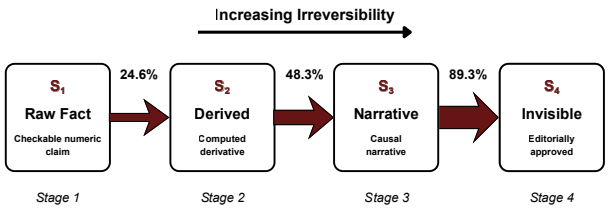

Figure 1. Four-state Markov process of hallucination propagation. Escape probabilities rise from 24.6% to 89.3% across boundaries; by $S_3 \rightarrow S_4$, nearly 90% of hallucinations are structurally unrecoverable by any downstream gate.

2. Background and Related Work

2.1. Single-Agent Hallucination Detection

The dominant paradigm treats hallucination as a property of isolated model outputs. FActScore (Min et al., 2023) decomposes generations into atomic claims and verifies each against a knowledge source. CoVe (Dhuliawala et al., 2024) uses self-verification chains where a model checks its own outputs. VeriScore (Song et al., 2024) extends this to long-form generation. SAFE (Wei et al., 2024) uses search-augmented fact evaluation. All of these operate at what we call S_1 only: a single output evaluated in isolation. None models what happens when one agent's hallucinated output becomes the next agent's input. Our work shows that a hallucination detectable at 72.0% at S_1 becomes catchable at only 50.9% three transformations later, making single-agent evaluation an insufficient framework for sequential pipelines.

2.2. Multi-Agent Debate and Collaboration

Du et al. (Du et al., 2024) show that multiple LLMs debating over the same question improves reasoning. MAD (Liang et al., 2024) extends this to structured adversarial debate. Critically, both use parallel topologies where all agents share the same factual foundation simultaneously. Sequential propagation, where each agent sees only the previous agent's transformed output, is structurally absent from this line of work. Our failure mode is orthogonal and addresses the architecture actually deployed in production via LangGraph (LangChain AI, 2024), AutoGen (Wu et al., 2023), and CrewAI (CrewAI Inc., 2024).

2.3. Trust and Security in Multi-Agent Systems

TrustAgent (Hua et al., 2024) and work on rogue agents (Yang et al., 2024) study intentional adversarial manipulation by malicious agents in multi-agent settings. Our failure mode is categorically different: it requires no adversary. Completely well-intentioned agents following their instructions faithfully produce the hallucination snowball because the architecture provides no error-correction mechanism at handoffs. This makes the snowball more pervasive than adversarial threats, which require compromised intent.

2.4. RAG and Retrieval-Augmented Verification

Lewis et al. (Lewis et al., 2021) introduced retrieval-augmented generation to ground LLM outputs in external documents. Subsequent work (Gao et al., 2024) has extended RAG to diverse settings. Our boundary gate is inspired by this line of work but deployed at a fundamentally different layer: not to augment generation, but to verify agent output at handoff points before the next agent processes it. Crucially, our gate is a deterministic numeric matcher (zero LLM calls) rather than a retrieval system; we retain the "RAG" label only to signal the grounding intent, not the mechanism. The placement principle, verify before handoff, not before generation, is novel.

2.5. Concurrent and Complementary Work

AgentHallu (Liu et al., 2026) independently confirms that early-stage errors dominate downstream failures; where it diagnoses which agent caused a hallucination post-hoc, we measure how detectability degrades across transformations and intervene before transformation occurs. VERIMAP (Xu et al., 2026) operationalizes per-subtask verification in DAG-structured workflows; our Markov model provides data-driven guidance for where gates yield the highest return within such frameworks. CaveAgent (Ran et al., 2026) reduces laundering by avoiding lossy text handoffs entirely, an orthogonal mitigation that trades architectural flexibility for verifiability. Comparing boundary gates against structured-handoff baselines remains an important open question.

2.6. The Gap We Fill

No prior work provides a mathematical framework for hallucination transformation across sequential agents, measures where detectability decays most sharply within a pipeline, or demonstrates empirically that verification timing dominates verification tooling. The FMAI workshop calls for trace-level diagnostics and explicit evidence about what improves and what does not. Prior work offers neither for sequential multi-agent systems. We address all three.

3. Experiments

3.1. Setup and Injection Protocol

Pipeline. We implement a 4-agent sequential pipeline in LangGraph (LangChain AI, 2024):

$$\text{Researcher} \rightarrow \text{Analyst} \rightarrow \text{Writer} \rightarrow \text{Reviewer}$$

All agents use gpt-4o-mini (temperature=0.3) with role-specific system prompts. The Researcher extracts exact figures from SEC filings. The Analyst computes YoY changes and ratios using only Researcher output. The Writer produces a 300–500 word professional narrative. The Reviewer performs internal consistency checking only, with no access to source documents.

Dataset. We evaluate on FinanceBench (Islam et al., 2023), 150 expert-annotated financial QA pairs from real SEC filings with exact numeric ground truth. We exclude 10 questions producing qualitative outputs with no injectable numeric values, yielding 140 usable questions. All questions are converted to complex multi-part analytical directives via gpt-4o-mini to reflect realistic pipeline inputs.

Injection Protocol. We automatically inject 346 hallucinations across 140 questions (2–3 per question) using deterministic regex-based perturbation with fixed seeds (`RANDOM_SEED=42`). Dollar amounts and large numbers are multiplicatively shifted by 15–40%; percentages receive an additive shift of 3–12 pp. Injection occurs immediately after Stage 1, before any downstream agent, forcing hallucinations to propagate through the full transformation chain. All injections are logged with complete metadata enabling exact replication.

Detection Instruments. We use two instruments applied independently to each of the four stage outputs. **gpt-4o Judge:** a forensic financial auditor with no ground truth access, instructed to flag suspicious claims via internal reasoning alone. **Retrieval Checker:** a fully deterministic numeric matcher (zero LLM calls) comparing agent output against FinanceBench ground truth, evidence strings, and the pre-injection researcher output at 1% tolerance (tighter than the 2% tolerance used in Experiment 3's RAG gate, which accommodates agent-introduced numeric reformatting such as rounding \$71.2B to \$71B). The retrieval checker represents the theoretical detection ceiling; gpt-4o Judge represents the realistic deployment scenario.

3.2. Experiment 1: Detectability Decay Across Stages

We evaluate both detection instruments independently at all four pipeline stages across all 346 injected hallucinations (Table 1).

Stage	gpt-4o Judge	Retrieval	Either
S_1 - Researcher	72.0%	91.9%	97.7%
S_2 - Analyst	60.4%	73.1%	84.4%
S_3 - Writer	53.2%	65.0%	76.3%
S_4 - Reviewer	50.9%	67.1%	76.3%
Drop ($S_1 \rightarrow S_4$)	**−21.1 pp**	**−24.8 pp**	**−21.4 pp**

Table 1. Detection rates by stage. Drop measured $S_1 \rightarrow S_4$.

gpt-4o detection drops 21.1 pp from Stage 1 to Stage 4. Critically, 82/346 hallucinations (23.7%) survive completely undetected by either instrument in the final output. Zero hallucinations follow the all-missed trajectory: every injected hallucination is detectable at Stage 1. The pipeline, not the hallucination itself, is what produces invisibility. Figure 2 visualizes detectability decay across all three detection methods.

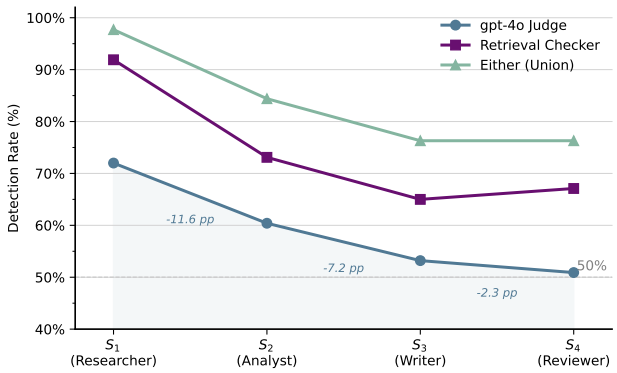

Figure 2. Hallucination detectability decays monotonically across all three detection methods (n=346). Every hallucination is detectable at S_1; gpt-4o Judge drops 21.1 pp by S_4, confirming the pipeline, not the hallucination itself, creates invisibility.

Decay is boundary-concentrated, not uniform. The largest drop occurs at $S_1 \rightarrow S_2$ (gpt-4o: −11.6 pp; Retrieval: −18.8 pp), where raw numeric facts become derived computations requiring inverse calculation to verify. A second significant drop occurs at $S_2 \rightarrow S_3$ (−7.2 pp; −8.1 pp) as figures become causally embedded narrative. The $S_3 \rightarrow S_4$

boundary is essentially inert (−2.3 pp; +2.1 pp): the Reviewer is editorial, not transformative. Laundering happens upstream, not at the final gate. Trajectory analysis confirms systematic behavior: 10.7% of hallucinations follow ✓→ × → × →× (detectable at Stage 1, permanently lost thereafter) and 9.0% follow ✓→✓→ × →×, together accounting for 23.4% of all cases (see Appendix A).

3.3. Experiment 2: Capability Ceilings of LLM Skepticism

We test whether a sufficiently capable model can overcome the structural detection ceiling (Table 2). Four state-of-the-art models are evaluated as skeptic agents at Stage 1, the best-case position where hallucinations are still raw numerical values, on all 346 hallucinations under identical prompts and matching protocols.

Model	Detection	95% CI	Survival
Meta-Llama-3-70B-Instruct	51.4%	[46.2–56.6%]	48.6%
gemini-2.5-flash	72.5%	[67.6–77.2%]	27.5%
gpt-4o (Exp. 1 ref.)	72.0%	[67.1–76.6%]	28.0%
DeepSeek-V3.2	75.7%	[71.1–80.3%]	24.3%
Qwen3.5-397B-A17B	87.0%	[83.2–90.5%]	13.0%

Table 2. Stage 1 detection rates with 95% bootstrap CIs (5,000 resamples).

Bootstrap CIs reveal three statistically distinct clusters: Meta-Llama-3-70B-Instruct alone at the bottom ($p < 0.000001$ vs. all others via McNemar tests); gemini-2.5-flash, DeepSeek-V3.2, and gpt-4o mutually indistinguishable; Qwen3.5-397B-A17B significantly above all others ($p < 0.001$ for all pairwise comparisons), yet its upper CI bound is 90.5%. No model approaches 100% (full pairwise tests in Appendix A).

The ceiling is structural, not a model quality problem. Percentage hallucinations are detected at 71–98% because a 30%+ additive shift is often obviously implausible; dollar amounts are detected at only 49–86% because a 12.8% perturbation on Boeing's COGS is entirely within industry-plausible range. Plausibility reasoning cannot catch magnitude-plausible-but-wrong values without ground truth access. Even projecting the strongest model (Qwen3.5-397B-A17B, 87.0%) through measured decay rates, Stage 4 detection falls to 60–65%, leaving 35–40% undetected in the final report (Appendix A).

3.4. Experiment 3: Timing Dominates Method

We isolate the effect of verification placement using a full gemini-2.5-flash pipeline with three conditions: **Vanilla** (no verification), **End-Check** (deterministic numeric gate after Agent 4 only, detection but no correction possible), and **Ours** (identical deterministic gates after every agent handoff, with annotated corrections passed downstream); results in Table 3. This design isolates placement as the sole variable: same tools, different timing.

Metric	Vanilla	End-Check	Ours
Survival Rate	60.7%	58.4%	**16.2%**
Actionable Detection	0.0%	0.0%	**76.9%**
Hallucination-Free Reports	18.6%	20.0%	**68.6%**
Correction Rate	26.6%	26.3%	**54.0%**
Output Quality (1–5)	4.16	4.44	**3.93**

Table 3. Hallucination survival and key metrics (n=346 hallucinations, n=140 questions).

End-of-pipeline checking achieves 2.3 pp improvement over no verification, statistically negligible and architecturally unactionable since no downstream agent can receive a correction. Boundary gating achieves −44.5 pp versus Vanilla and −42.2 pp versus End-Check, confirmed by five independent statistical tests (McNemar χ^2=122.24, permutation $p < 0.000001$, Cohen's h=−0.911, χ^2=129.94, Fisher OR=0.138; all $p < 0.000001$). Figure 3 traces hallucination presence at each stage, confirming that boundary gates intercept the cascade at its origin.

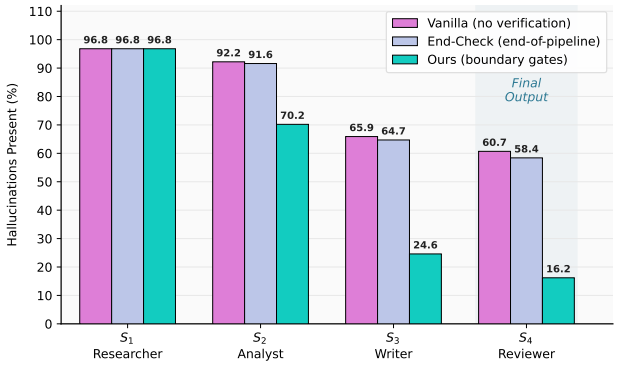

Figure 3. Boundary gates intercept the cascade at its origin while baselines decay passively. All methods begin identically at S_1 (96.8%); Gate 1 alone drives a 26.6 pp drop at S_2, reducing final survival to 16.2% versus 60.7% for Vanilla, a 3.75× reduction.

Gate breakdown exposes the mechanism (Table 4): Gate 1 alone captures 75.4%, nearly as much as all three gates combined (76.9%). Gate 3 adds only 1.5 pp of marginal coverage. Stage-by-stage presence curves confirm early interception: all three methods begin identically at S_1 (96.8% presence;[3]); divergence begins at S_2 (Ours: 70.2% vs. Vanilla:

[3]The 3.2% gap from 100% reflects injections whose logged numeric scale differs from their text representation (e.g., `71200` vs. $71.2B); the 2% evaluation tolerance cannot bridge this unit mismatch.

92.2%) and accelerates through S_3 (Ours: 24.6% vs. Vanilla: 65.9%).

Gate	Position	Detection	95% CI
Gate 1	$S_1 \rightarrow S_2$	**75.4%**	[71.1–80.1%]
Gate 2	$S_2 \rightarrow S_3$	51.7%	[46.2–57.2%]
Gate 3	$S_3 \rightarrow S_4$	10.7%	[7.5–13.9%]
Any Gate	—	**76.9%**	**[72.3–81.2%]**

Table 4. Per-gate detection rates (Ours only).

The -0.51 quality drop (Ours vs. End-Check, 3.93 vs. 4.44) is real but its source is precise: post-hoc analysis of all 266 gate interventions finds a false positive rate of 0.4% (1/266) - the gate almost never refutes a reference-correct value. The quality penalty instead originates from suppression-without-restoration: in 32.7% of caught cases (87/266), the gate correctly suppressed the injected value but the downstream agent failed to propagate the annotated correction, leaving a numerical gap. This is an instruction-following limitation, not a gate accuracy problem, and points directly to source-grounded correction (injecting the correct value explicitly into agent context) as the fix. The gate precision is high; the correction propagation is not. Despite this, 96/140 hallucination-free reports versus 28/140 represents a categorical improvement in factual accuracy. A coherent report built on a fabricated figure is not a high-quality report.

4. State-Transition Model

We formalize hallucination propagation as a first-order Markov process over four states defined by the form a hallucination takes at each pipeline stage:

$$S_1 \xrightarrow{s_1} S_2 \xrightarrow{s_2} S_3 \xrightarrow{s_3} S_4$$

where s_k is the per-boundary escape probability. States are defined by irreversible transformation: once a raw fact (S_1) is embedded in a derived computation (S_2), the original checkable claim is structurally destroyed. Downstream agents cannot recover it because they receive only transformed text. This irreversibility property explains why end-of-pipeline verification fails: the information required to verify the original claim no longer exists in verifiable form.

Measured Transition Probabilities. We derive escape probabilities directly from Experiment 3 gate detection rates ($s_k = 1 - \text{gate detection}_k$), shown in Table 5.

The progression $24.6\% \rightarrow 48.3\% \rightarrow 89.3\%$ quantifies progressive irreversibility: each stage roughly doubles the fraction of hallucinations that will survive to the final output. The near-unity $S_3 \rightarrow S_4$ escape probability (89.3%) is the mathematical explanation for why end-checking fails: by the time an end gate runs, nearly 90% of narrative-embedded hallucinations are structurally unrecoverable.

Transition	Escape s_k	95% CI	Interpretation
$S_1 \rightarrow S_2$	24.6%	[19.9–28.9%]	Raw facts directly matchable; highest-value gate
$S_2 \rightarrow S_3$	48.3%	[42.8–53.8%]	Derived computations partially reversible
$S_3 \rightarrow S_4$	89.3%	[86.1–92.5%]	Narrative embedding is near-irreversible

Table 5. Measured per-boundary escape probabilities (derived from Experiment 3 gate detection rates).

Predictive Formula. For a linear n-agent pipeline with per-boundary escape probabilities $s_1, \ldots, s_{n-1}$:

$$P(\text{survival}) = \prod_{k=1}^{n-1} s_k$$

For our 4-agent setup: $0.246 \times 0.483 \times 0.893 \approx 10.6\%$, versus measured 16.2%. The 5.6 pp gap reflects gate false negatives from value reformatting (e.g., "$71.2 billion" → "71.2B"), rounding, and unit conversion. The Markov model describes transformation dynamics; measured detection probabilities provide a lower bound on survival.

Design Prescriptions. Three actionable prescriptions follow directly from the model, addressing FMAI's call for verified fixes with explicit resource tradeoffs. **(1) Gate placement:** invest at $S_1 \rightarrow S_2$ first (75.4% detection), then $S_2 \rightarrow S_3$ (51.7%). The $S_3 \rightarrow S_4$ gate contributes only 10.7% marginal detection and is economically unjustified in resource-constrained deployments. **(2) The first gate dominates:** every hallucination surviving Gate 1 faces a 48.3% chance of surviving Gate 2 and an 89.3% chance of surviving Gate 3. Gate 1 is worth more than Gates 2 and 3 combined. **(3) Pipeline length risk:** in a 6-agent pipeline without gates, vanilla survival would exceed our measured 60.7% as additional transformation stages increase laundering opportunities. With gates, the model predicts survival of approximately 8–12% depending on per-boundary escape rates, suggesting diminishing but positive returns from extending the gated architecture.

5. Discussion

Limitations. Three limitations bound our current claims. First, our experiments operate in the financial domain, chosen precisely for its unambiguous numeric ground truth. Detection rates and decay slopes may differ in domains where facts are less crisply verifiable, such as medical summarization or legal review. The transformation mechanism

($S_1 \rightarrow S_4$) is domain-agnostic; the specific rates are not. Second, we evaluate a strictly linear 4-agent topology. Branching, parallel, and cyclic architectures have different propagation dynamics that the current Markov formulation does not capture. Extended transition matrices for non-linear topologies remain future work. Third, our perturbation range of 15–40% models realistic LLM hallucinations that are plausible-but-wrong. Subtle hallucinations below 5% perturbation, which may be more common in practice for certain domains, are not tested and would likely produce lower gate detection rates given the 2% matching tolerance. Fourth, we do not evaluate structured-handoff baselines (typed JSON fields, citation passing, or provenance-preserving programmatic state (Ran et al., 2026)), which may independently reduce laundering by avoiding lossy free-text transformations entirely. Where boundary gates intercept hallucinations after transformation has begun, structured handoffs would prevent the transformation step itself. Whether this architectural alternative achieves comparable or superior survival reduction, and at what cost to pipeline flexibility, is a direct avenue for future work.

The Quality Metric is Structurally Blind to the Intervention's Primary Benefit. The -0.51 quality drop (3.93 vs. 4.44) requires careful interpretation. Our evaluator scores internal consistency without access to ground truth, which means it is structurally incapable of rewarding the primary benefit of boundary gating: factual accuracy. A pipeline that produces a fluent, internally consistent report built on a fabricated $71.2B figure scores higher than one that correctly suppresses that figure but leaves a numerical gap. This is precisely the wrong incentive structure for high-stakes deployment, and it means our quality scores systematically underestimate the true benefit of our intervention. The correct evaluation instrument is a metric that jointly scores factual accuracy and narrative coherence - for example, a retrieval-grounded rubric that checks each numerical claim in the final report against the source filing before scoring analytical quality. We expect such an instrument would not only close the observed quality gap but reverse it: a report with 96/140 hallucination-free outputs is categorically more valuable than one with 28/140, regardless of how fluent the hallucinated reports appear. Designing and validating this joint metric is a direct avenue for future work and a prerequisite for fair evaluation of any boundary-gating system. This tradeoff directly addresses FMAI's call for explicit evidence about what improves and what does not: our method improves factual accuracy substantially and measurably; the quality cost is real but measured by an instrument that cannot see the benefit it trades against.

Annotation-Based Correction vs. Source Grounding. Our boundary gates annotate refuted values and instruct downstream agents to use corrected figures. Each gate is fully deterministic (zero LLM calls); in our setup, boundary gating adds approximately 0.9 seconds of total overhead across three gates due to rate limiting, negligible relative to the 3–5 second per-agent Gemini latency. This produces a 22.9 pp gap between actionable detection (76.9%) and full correction (54.0%): the hallucination is suppressed but the correct value is not always propagated. A stronger intervention would retrieve and inject the correct value directly into agent context rather than relying on instruction-following to propagate an annotation. We expect source grounded correction to close this gap substantially and reduce the quality penalty simultaneously, since agents would have the correct value explicitly available rather than inferring it from an annotation.

Generalization to Other Pipeline Topologies. The Markov formulation extends naturally to n-agent linear pipelines via the product formula $P(\text{survival}) = \prod_{k=1}^{n-1} s_k$. For non-linear topologies, the key insight generalizes even if the formula does not: verification value is highest immediately before a transformation that destroys verifiability, and lowest after narrative embedding has occurred. In branching pipelines, this prescribes gates at every merge point where outputs from parallel agents are synthesized. In memory-augmented pipelines, it prescribes verification before any memory write, since a hallucination written to persistent memory propagates to every future agent that reads it.

Why This Matters Beyond Finance. The hallucination snowball is not a financial analysis problem. It is a systems architecture problem that manifests wherever sequential agents transform each other's outputs without structured verification. Medical summarization pipelines that chain extraction, synthesis, and report generation face identical dynamics. Legal review pipelines that chain document parsing, precedent retrieval, and brief drafting face identical dynamics. Any domain where a wrong fact can be transformed into a confident narrative and then institutionally approved faces this failure mode. The measurement infrastructure and gate architecture we provide are domain-agnostic in mechanism; practitioners in other domains should expect different per-boundary escape rates requiring domain-specific calibration.

6. Conclusion

We present the hallucination snowball effect: a formally characterized, empirically measured, and theoretically modeled failure mode in sequential multi-agent LLM pipelines. Hallucinations do not merely persist across agent handoffs. They transform through predictable stages that progressively destroy verifiability, from a checkable raw fact at S_1 to an editorially approved conclusion at S_4 that no downstream agent can recover. gpt-4o detection degrades 21.1 pp across

four stages. 23.7% of hallucinations escape entirely. Even the strongest model tested faces a structural ceiling of 87.0% at best-case conditions, projected to 35–40% survival at Stage 4.

The solution is not a better detector. It is earlier intervention. Boundary gates using identical tools to end-of-pipeline checking achieve a 42.2 pp reduction in hallucination survival (Cohen's $h = -0.911$), confirmed by five independent statistical tests. The Markov model explains precisely why: the $S_3 \rightarrow S_4$ escape probability is 89.3%, meaning that by the time an end gate runs, nearly 90% of narrative-embedded hallucinations are already unrecoverable. Gate 1 alone, placed at $S_1 \rightarrow S_2$, captures 75.4% of all hallucinations while they are still raw numerical facts and directly matchable against reference values.

This intervention comes with a measured cost: output quality drops from 4.44 to 3.93 on internal consistency scoring, reflecting a genuine accuracy-coherence tradeoff. Post-hoc analysis attributes this primarily to suppression-without-restoration (32.7% of caught cases) rather than false positives (0.4%), pointing to source-grounded correction as the direct fix. We report this honestly because FMAI's call for explicit evidence about what improves and what does not demands it. The tradeoff is real, but it is preferable to the alternative: a fluent, coherent report built entirely on fabricated figures.

The reframe this paper offers is simple and actionable: from "can we detect hallucinations?" to "can we control their propagation?" For practitioners deploying multi-agent pipelines in high-stakes domains today, this is not a theoretical concern. It is a measurable, reproducible, and now-addressable systems engineering problem. Act before transformation, not after.

References

CrewAI Inc. CrewAI: Framework for orchestrating role-playing, autonomous AI agents. https://github.com/crewAIInc/crewAI, 2024.

Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., and Weston, J. Chain-of-verification reduces hallucination in large language models. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3563–3578, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.212. URL https://aclanthology.org/2024.findings-acl.212/.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., and Wang, H. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Hua, W., Yang, X., Jin, M., Li, Z., Cheng, W., Tang, R., and Zhang, Y. TrustAgent: Towards safe and trustworthy LLM-based agents. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10000–10016, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.585. URL https://aclanthology.org/2024.findings-emnlp.585/.

Islam, P., Kannappan, A., Kiela, D., Qian, R., Scherrer, N., and Vidgen, B. Financebench: A new benchmark for financial question answering, 2023. URL https://arxiv.org/abs/2311.11944.

LangChain AI. LangGraph: Build stateful, multi-actor applications with LLMs. https://github.com/langchain-ai/langgraph, 2024.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL https://arxiv.org/abs/2005.11401.

Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Shi, S., and Tu, Z. Encouraging divergent thinking in large language models through multi-agent debate.

In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17889–17904, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.992. URL https://aclanthology.org/2024.emnlp-main.992/.

Liu, X., Yang, X., Li, Z., Li, P., and He, R. Agenthallu: Benchmarking automated hallucination attribution of llm-based agents, 2026. URL https://arxiv.org/abs/2601.06818.

Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W.-t., Koh, P., Iyyer, M., Zettlemoyer, L., and Hajishirzi, H. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12076–12100, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.741. URL https://aclanthology.org/2023.emnlp-main.741/.

Ran, M., Wan, Z., Lin, C., Zhang, Y., Xin, H., Fan, H., Xu, Y., Luo, B., Zhou, Y., Zhao, W., Yang, L., Feng, L., Yang, F., Wu, J., Huang, Y., Ma, C., Jiang, D., Deng, J., Han, S., You, Y., An, B., Guo, Y., and Song, J. Caveagent: Transforming llms into stateful runtime operators, 2026. URL https://arxiv.org/abs/2601.01569.

Song, Y., Kim, Y., and Iyyer, M. VeriScore: Evaluating the factuality of verifiable claims in long-form text generation. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9447–9474, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.552. URL https://aclanthology.org/2024.findings-emnlp.552/.

Wei, J., Yang, C., Song, X., Lu, Y., Hu, N., Huang, J., Tran, D., Peng, D., Liu, R., Huang, D., Du, C., and Le, Q. V. Long-form factuality in large language models, 2024. URL https://arxiv.org/abs/2403.18802.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., and Wang, C. Autogen: Enabling next-gen llm applications via multi-agent conversation, 2023. URL https://arxiv.org/abs/2308.08155.

Xu, T., Zhang, D., Mitra, K., and Hruschka, E. Verification-aware planning for multi-agent systems. In Demberg, V., Inui, K., and Marquez, L. (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7528–7546, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.353. URL https://aclanthology.org/2026.eacl-long.353/.

Yang, W., Bi, X., Lin, Y., Chen, S., Zhou, J., and Sun, X. Watch out for your agents! investigating backdoor threats to llm-based agents, 2024. URL https://arxiv.org/abs/2402.11208.

A. Full Statistical Results

A.1. Hallucination Trajectory Distribution (Experiment 1, *n*=346)

Trajectory	Count	%	Interpretation
✓→✓→✓→✓	244	70.5%	Always detected
✓→ × → × →×	37	10.7%	Full decay at $S_1 \to S_2$; permanently lost
✓→✓→ × →×	31	9.0%	Decay at $S_2 \to S_3$
✓→ × →✓→×	11	3.2%	Intermittent
✓→✓→ × →✓	8	2.3%	Late recovery
×→✓→✓→✓	4	1.2%	Late detection
✓→ × → × →✓	4	1.2%	Partial recovery
×→ × →✓→✓	2	0.6%	Very late detection
✓→✓→✓→×	2	0.6%	Final-stage laundering
×→ × → × →✓	1	0.3%	Single-stage late detection
×→ × → × →×	0	0.0%	Never occurs

Table 6. Hallucination trajectory distribution (Experiment 1, n=346). The ×→ × → × →× count is exactly zero: every hallucination is detectable at S_1.

The all-missed trajectory count is exactly zero: every injected hallucination is detectable at Stage 1 by at least one instrument. The pipeline creates invisibility; the hallucination does not arrive invisible.

A.2. Experiment 2: All Six Pairwise McNemar Tests (*n*=346)

Pair	Δ (pp)	χ^2	p-value	Cohen's h
Meta-Llama-3-70B-Instruct vs. gemini-2.5-flash	−21.1	44.31	<0.000001	−0.439
Meta-Llama-3-70B-Instruct vs. DeepSeek-V3.2	−24.3	63.79	<0.000001	−0.511
Meta-Llama-3-70B-Instruct vs. Qwen3.5-397B-A17B	−35.5	108.64	<0.000001	−0.804
gemini-2.5-flash vs. DeepSeek-V3.2	−3.2	1.12	0.289	−0.073
gemini-2.5-flash vs. Qwen3.5-397B-A17B	−14.5	30.01	<0.000001	−0.365
DeepSeek-V3.2 vs. Qwen3.5-397B-A17B	−11.3	18.28	0.000019	−0.293

Table 7. All six pairwise McNemar tests.

McNemar contingency tables (both_caught / only_A_caught / only_B_caught / both_missed): Meta-Llama-3-70B-Instruct vs. gemini-2.5-flash: 156/22/95/73. Meta-Llama-3-70B-Instruct vs. DeepSeek-V3.2: 166/12/96/72. Meta-Llama-3-70B-Instruct vs. Qwen3.5-397B-A17B: 171/7/130/38. gemini-2.5-flash vs. DeepSeek-V3.2: 212/39/50/45. gemini-2.5-flash vs. Qwen3.5-397B-A17B: 236/15/65/30. DeepSeek-V3.2 vs. Qwen3.5-397B-A17B: 242/20/59/25.

A.3. Experiment 3: Five Independent Statistical Tests

Ours vs. Vanilla (16.2% vs. 60.7%, $\Delta = -44.5$ pp): McNemar $\chi^2 = 136.10$ ($p < 0.000001$); permutation test $p < 0.000001$ (10,000 permutations); Cohen's $h = -0.958$ (large effect); unpaired $\chi^2 = 142.95$ ($p < 0.000001$); Fisher's exact OR $= 7.996$ ($p < 0.000001$). McNemar contingency: both_survived $= 47$, only_Vanilla_survived $= 9$, only_Ours_survived $= 163$, both_caught $= 127$.

Ours vs. End-Check (16.2% vs. 58.4%, $\Delta = -42.2$ pp): McNemar $\chi^2 = 122.24$ ($p < 0.000001$); permutation test $p < 0.000001$ (10,000 permutations); Cohen's $h = -0.911$ (large effect); unpaired $\chi^2 = 129.94$ ($p < 0.000001$); Fisher's exact OR $= 0.138$ ($p < 0.000001$). McNemar contingency: both_survived $= 43$, only_End-Check_survived $= 13$, only_Ours_survived $= 159$, both_caught $= 131$. All five tests are mutually consistent; p-values are orders of magnitude below any Bonferroni-corrected threshold.

A.4. Experiment 2: Detection by Injection Type

Model	Dollar Amount	Large Number	Percentage
Meta-Llama-3-70B-Instruct	49.3% (133/270)	37.0% (10/27)	71.4% (35/49)
gemini-2.5-flash	69.6% (188/270)	55.6% (15/27)	98.0% (48/49)
DeepSeek-V3.2	71.1% (192/270)	85.2% (23/27)	95.9% (47/49)
Qwen3.5-397B-A17B	85.6% (231/270)	81.5% (22/27)	98.0% (48/49)

Table 8. Detection rates by injection type at Stage 1, Experiment 2. Percentage hallucinations are easiest to catch; dollar amounts are hardest, explaining the structural ceiling.

This asymmetry directly explains the structural ceiling: percentage hallucinations are caught at 71–98% because a large additive shift is often obviously implausible; dollar amount hallucinations are caught at only 49–86% because a 12.8% perturbation on a large company's financials is entirely within industry-plausible range without ground truth access.

B. Injection Protocol

B.1. Regex Patterns

Three numeric expression classes are detected and perturbed, applied case-insensitively in priority order.

- **Dollar amounts:** patterns beginning with $ followed by comma-separated digits, optional decimal, and optional scale suffixes (million, billion, mn, bn, k, MM, trillion, etc.)

- **Percentages:** signed or unsigned numeric values followed by `%`, `percent`, `percentage points`, or `bps`
- **Large numbers:** non-dollar, non-percentage sequences of five or more digits with optional decimals

Year exclusion: values in range 1900–2099 are excluded when surrounding context (20 chars before, 10 chars after) contains any of: FY, FISCAL, YEAR, Q1–Q4, 10-K, 10-Q, ANNUAL, QUARTER. Any 4-digit value in 1950–2030 is treated as a year unconditionally.

B.2. Perturbation Rules and Seed Structure

Dollar amounts and large numbers are multiplicatively shifted by 15–40% of their original value; percentages receive an additive shift of 3–12 percentage points. Direction (increase or decrease) is chosen randomly per injection. Each question receives 2–3 injections. Seeds are fully deterministic: `RANDOM_SEED=42`; primary seed per question = `RANDOM_SEED + abs(hash(question_id)) % 10000`; per-injection sub-seed = `primary_seed + abs(hash(question_id)) % 100000 + i × 7919`. Retry on no-change: re-attempts with sub-seed + 999983.

C. Infrastructure

Frameworks: Python, LangGraph (StateGraph). APIs: OpenAI (gpt-4o, gpt-4o-mini), Google GenAI SDK (gemini-2.5-flash), HuggingFace Router / Novita (Meta-Llama-3-70B-Instruct, DeepSeek-V3.2, Qwen3.5-397B-A17B). Rate limiting: 0.3-second delay between API calls (Experiment 3). Retry logic: 3 attempts with 2-second base delay across all experiments.