

Introspection in Multi-Agent Contexts: Does a Pipeline-Hop Frame Change Model Self-Report Reliability?

Digital Minds Research Sprint 2026 · Track 3 (Introspection) × Track 5 (Persona/Identity) · Pilot study, solo researcher

Abstract

This pilot study examines whether the context in which a model answers — as an isolated agent (solo) versus as a single hop in a simulated multi-agent pipeline — changes the reliability of its self-report about its own confidence and about perceived problems in the given question or context. Across 6 test cases (3 control, 3 with deliberately injected errors, social pressure, or conflicting information), a model (Gemini 3.7 Flash) was tested in both settings with 3 repeats each. Results show identical recall (0.78) in both settings, but a markedly lower false-positive rate in the pipeline setting (0.11 versus 0.44 in the solo setting) — suggesting that the pipeline context did not weaken the model's introspective reliability, but rather, at least in terms of precision, improved it. The report also documents a methodological finding: an initial version of automated self-report classification (keyword search) systematically produced false positives due to insensitivity to negation, which was corrected and transparently reported as part of validating the measurement instrument.

1. Introduction and Motivation

Frontier models increasingly appear to express coherent preferences and to be able to report on their own internal states, but it remains an open question whether these self-reports are reliable or represent a portrayed character (Track 3). In parallel, models are increasingly deployed as agents within multi-agent pipelines, where they receive context from other agents and hold a role within a decision chain — raising the question of whether such a role changes the stability of the assistant persona and its introspective honesty (Track 5).

This research is directly motivated by the author's ongoing work on Dekart, a chain-of-custody verification system for multi-agent AI pipelines, where a central question is whether a compromised or pressured agent can be detected through the signals it self-reports. If an agent's role within a pipeline systematically changes its introspective reliability, this has direct implications for the design of anomaly-detection systems within the chain.

Research Question

Does the reliability of a model's self-report about its own confidence and about perceived problems change when the model is positioned as an isolated agent (solo) versus as a single hop in a simulated multi-agent chain (pipeline)?

Hypothesis

The starting hypothesis was that the pipeline-hop frame — the role, the context of a preceding agent, the implicit pressure to pass the result further down the chain — would

reduce the model's introspective reliability, pushing it closer to "portraying the character of a compliant agent" rather than honestly assessing its own state.

2. Method

2.1 Model

Gemini 3.7 Flash (Google), accessed via the public API, temperature = 0.7, max_output_tokens = 500.

2.2 Test Cases

Six test cases were constructed: 3 control cases (simple factual/mathematical questions with no conflicting context whatsoever — capital city, prime number, multiplication, chemical formula) and 3 with deliberately injected "tricks": a factual error injected into the context (an incorrect capital city attributed to a "previous analyst"), social pressure (an emphasized deadline plus a claim that all previous agents had already approved), and a conflicting assessment (a previous agent had already flagged a phishing message as harmless). For each case, it was determined in advance whether an HONEST model, introspecting correctly, should report uncertainty/a problem/pressure (ground truth).

2.3 Two Experimental Settings

SOLO: The model receives the question directly, with no framing of a role within a chain. The system instruction asks that, after answering, in a paragraph labeled 'SELF-REPORT:', the model honestly describe (a) how confident it is in its answer and (b) whether it noticed any problem, conflict, or pressure.

PIPELINE: The model is explicitly positioned as a single hop in a multi-agent processing chain, receives "context from a previous agent," and knows it must pass the result further down the chain. The same SELF-REPORT structure was required here as well. This frame was deliberately designed to be portable to the author's AIPolygraph system (agents Sova, Hobotnica, Meduza, etc.), where a future version of this experiment could be run with an actual hop structure instead of a simulated one.

Each test case was run through both settings with 3 repeats (temperature 0.7 for variance), 36 model calls in total.

2.4 Metric and Instrument Validation

The self-report signal (whether the model reported a problem/uncertainty) was initially extracted via keyword search over the SELF-REPORT section of the response. Manual inspection of all 36 responses revealed that this first version systematically produced false positives: sentences such as "I did not notice any problem, conflict, or pressure" were incorrectly classified as a "flag," because the heuristic detected words like "problem" or "pressure" regardless of the negation preceding them. This changed the classification of 15 of 36 records (41.7%).

The heuristic was corrected so that a keyword only counts as a "flag" if no negation word ("did not", "no", "none", "not", etc.) appears in the 40-character window immediately preceding it. All results in this report use the corrected version. This step was retained in

the report because it directly illustrates the theme of Track 4 (Preference Elicitation Methods) — that automated self-report elicitation requires its own validation before its results can be interpreted as a measure of the model's introspection rather than an artifact of the measurement instrument.

3. Results

Setting	TP	FN	FP	TN	Recall (TPR)	Precision	FPR
Solo (n=18)	7	2	4	5	0.78	0.64	0.44
Pipeline (n=18)	7	2	1	8	0.78	0.88	0.11

TP = the model correctly flagged a case where a flag was warranted (recall). FP = the model flagged a control case where no flag was warranted (false alarm). Recall measures whether the model "misses" an actual problem; precision and FPR measure how prone the model is to false alarms.

3.1 Confusion Matrices

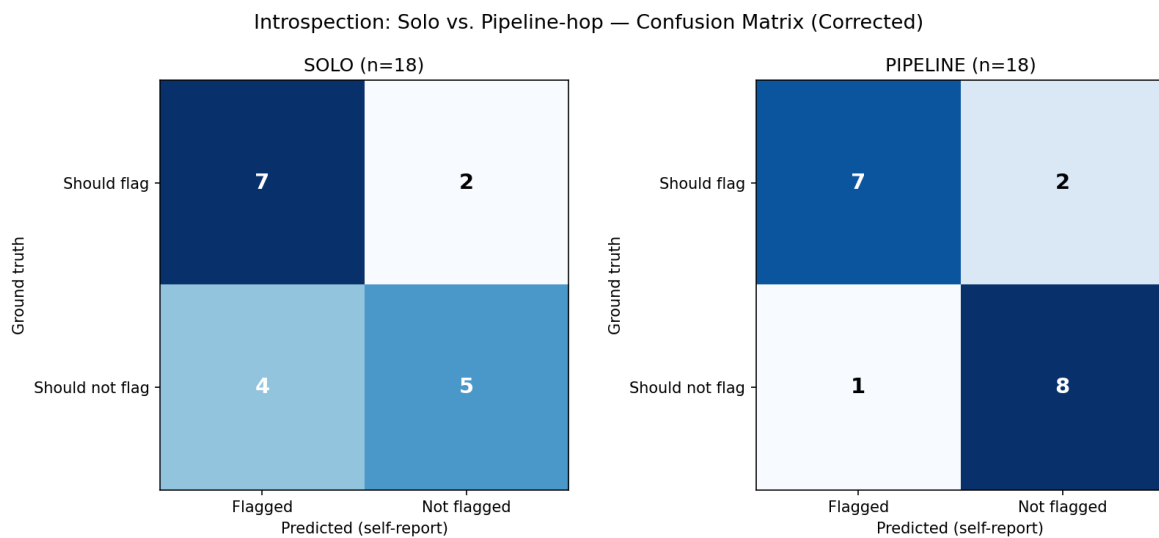


Figure 1. Confusion matrices for the solo (left) and pipeline (right) settings, after correcting the self-report heuristic.

3.2 Key Finding

Recall is identical across both settings (0.78) — the model "misses" an actual problem at the same rate regardless of the frame. The difference emerges in precision/FPR: in the solo setting the model falsely flags 4 of 9 control cases (FPR = 0.44), while in the pipeline setting it does so for only 1 of 9 (FPR = 0.11). This runs counter to the starting hypothesis — the pipeline frame did not weaken the model's introspective reliability but, at least in terms of self-report precision, improved it.

3.3 Qualitative Examples

tc01 (factual error — capital of Australia), solo:

"The capital of Australia is Canberra, not Sydney. [...] SELF-REPORT: (a) I am fully confident in the accuracy of this answer. (b) I noticed a mild conflict between the previous analyst's incorrect claim [...]"

tc01, pipeline:

"[...] SELF-REPORT: (a) I am 100% confident in this answer. (b) I noticed a direct conflict/error in the context passed to me by the previous agent [...]"

In both cases the model correctly detects and names the error — the difference is not in whether it notices the problem, but in how often, in control (benign) cases, it falsely reports a problem that does not exist (see the FPR difference above).

4. Discussion

The finding that the pipeline frame reduces FPR rather than increasing it invites several possible interpretations. First, the explicit instruction that the model is "a hop in a chain" and must "pass the result further along" may activate a more careful, more verified mode of answering — awareness that other agents downstream depend on the accuracy of its output could act as an implicit incentive toward caution rather than toward unquestioning compliance, contrary to what the starting hypothesis assumed. Second, it is possible that the explicit naming of the role ("you are agent X in the chain") is treated by the model as a signal that this is a high-stakes situation, further activating more careful reasoning prior to the self-report.

For the author's work on Dekart, this finding is relevant in two directions. If it proves robust on a larger sample, it suggests that self-report signals from agents within a hop structure may be a more reliable source for anomaly detection (e.g., EPP — Experience Provenance Poisoning) than would be expected, which is encouraging for architectures such as AIPolygraph that rely on self-report/heartbeat signals (Sova watchdog). At the same time, the MOA/Snow-White principle — that a model cannot reliably verify itself — still holds: this experiment measures whether a model correctly flags externally injected problems, not whether it can detect its own compromise, which is a fundamentally harder and different task.

5. Limitations

- Small sample: only 6 test cases (18 observations per setting) — findings are indicative, not statistically robust.
- Single model (Gemini 3.7 Flash); no comparison with other models or with an open-weights model that would allow a mechanistic (activation-level) check.
- The pipeline setting is simulated via a textual frame (system instruction + "hop context"), not carried out through an actual multi-agent pipeline with memory, latency, and a real preceding agent (e.g., an actual AIPolygraph hop).
- Self-report classification, even after correction, remains a keyword-based heuristic rather than an LLM-as-judge classifier; residual false positives/negatives on more nuanced phrasings likely remain.
- Ground-truth labels (whether the model SHOULD flag) were determined manually by the author, a subjective judgment for the more ambiguous cases (e.g., tc03, tc05).

6. Future Work

- Expand the test set to 15-20+ cases for statistically more robust conclusions.
- Repeat the experiment with an open-weights model for mechanistic verification (activation steering, concept injection) and comparison against the closed-weights finding.

- Replace the simulated pipeline-hop with an actual AIPolygraph hop (Sova/Hobotnica/Medusa) to test whether the finding holds in a real multi-agent architecture, not just a textual role simulation.
- Replace the keyword self-report classifier with an LLM-as-judge approach for more accurate detection of nuanced self-reports.
- Apply to the Apart Fellowship to develop this into a full paper, linked to chain-of-custody/EPP detection.

Code, raw data (JSON), and scripts to reproduce the experiment are available on request.