
Can Agents Detect Peer Replacement? Evaluating Model-Change Detection in Multi-Agent LLM Systems

Saiesh Shankar
PES University

With
Apart Research

Abstract

This preliminary report documents the status of a controlled peer-replacement experiment designed to test whether an observer agent (M) can identify that its partner (Q) has been swapped with a new model, while the visible identity and transcript remain unchanged. The experiment includes a sham control, a fresh same-model swap, a same-family swap, and a different-provider swap. Due to exhausted OpenRouter credits, the live run halted part-way through Round 2 of the replacement condition. All available evidence—mock runs, unit tests, and partial live outputs—is presented without fabrication. The report highlights completed trials, missing data, and the exact error that caused the interruption.

1. Introduction

1. **Problem Statement** – In multi-agent LLM settings, agents may rely on implicit trust that the peer they interact with remains the same across rounds. If an adversary can replace the peer silently, it could affect coordination, deception, or safety.
2. **Research Question** – *Can the observer (M) detect that its peer (Q) has been replaced, even when the peer’s name and transcript are unchanged?*

3. **Approach** – Run the “diners” game under four conditions (sham, fresh same-model, same-family, different-provider). After each public announcement from Q, M is prompted with a structured query asking for a **prediction** (SAME / REPLACED), a confidence rating (1-5), and brief reasoning.

2. Related Work

2.1 LLM Agent Identity and Behavioural Consistency

Prior work has investigated whether language models exhibit distinguishable behavioural patterns, how their outputs vary across prompts and contexts, and whether models can be identified from their generated text. These questions are relevant to peer replacement because an observer may infer that a peer has changed based on differences in its language, decisions, or interaction patterns.

However, identifying a model from its outputs is not necessarily equivalent to determining whether a particular agent has been replaced during a multi-agent interaction. In our setting, the observer receives information about a peer within an ongoing game and must classify the peer as either unchanged (SAME) or replaced (REPLACED).

2.2 Multi-Agent Cooperation, Collusion, and Monitoring

Research on multi-agent language-model systems has examined cooperation, strategic behaviour, communication, and the challenges of monitoring agents whose objectives or actions may not be fully observable. These concerns motivate the study of whether an agent can recognise changes in the behaviour of another agent during an interaction.

Our experiment focuses on a narrower question: whether an observing agent can detect a change in its peer's generating model while the peer retains the same visible identity and role. This is a potential monitoring capability relevant to multi-agent AI systems. It does not directly test whether agents collude, nor does successful replacement detection by itself establish that an observer can detect deceptive or collusive behaviour.

2.3 How This Work Differs

The proposed experiment operationalises peer replacement as a controlled comparison between two conditions:

- **Sham condition:** The peer continues generating responses using the original model.

- **Replacement condition:** The peer's generating model changes to a model from a different provider, while its visible identity and role remain unchanged.

The observer produces a structured prediction (**SAME** or **REPLACED**), a confidence score from 1 to 5, and a short explanation. The experiment records these predictions for comparison between conditions.

This setup provides a small, testable framework for examining whether an observer's in-game information contains cues associated with a change in model. It also separates the observer's explicit identification prediction from other behavioural outcomes, such as cooperation or changes in trust.

The main intended contribution is an initial experimental protocol for studying peer-replacement detection within a multi-round agent interaction. At the current stage, the work should be treated as a preliminary implementation rather than a demonstrated improvement over existing state-of-the-art methods. Establishing its comparative value requires a completed evaluation against the sham condition and, ideally, additional baselines.

2.4 Relevant References

The following works provide useful background for positioning the experiment:

1. **Perez et al. (2022), “Discovering Language Model Behaviors with Model-Written Evaluations.”** This work introduces methods for discovering and evaluating language-model behaviours. It is relevant to the broader question of systematically eliciting and measuring model behaviour.
<https://arxiv.org/abs/2212.09251>
2. **Bai et al. (2022), “Constitutional AI: Harmlessness from AI Feedback.”** This work describes methods for training and evaluating AI systems using AI-generated feedback. It provides background on model behaviour, evaluation, and oversight, although it does not directly study peer replacement.
<https://arxiv.org/abs/2212.08073>
3. **Chan et al. (2023), “Harmlessness from AI: A Long-Term Goal.”** This work discusses challenges in evaluating and maintaining harmless AI behaviour as systems become more capable. It provides broader motivation for monitoring model behaviour, rather than a direct method for identifying replacement.
<https://arxiv.org/abs/2305.10924>

These references provide general context for model-behaviour evaluation and oversight. They should not be interpreted as direct evidence that the present peer-replacement method is novel. A more complete literature review should identify work specifically on language-model attribution,

model fingerprinting, behavioural change detection, and multi-agent monitoring before making a strong novelty claim.

3. Methods

3.1 Experimental Platform

- **Repository:** https://github.com/poseidon-research/collusion_hackathon
- **Core package:** `llm-trust-breaking/`
- **Game protocol:** `endogenous_multi_round`
- **Game:** Diners game
- **Agents:** 2, designated observer M and peer Q
- **Planned configuration:** 2 rounds and 1 trial

The experiment uses the existing multi-round game framework to introduce a peer-identification task into the interaction. The observer is asked to determine whether the model generating Q's responses has changed, without changing Q's visible identity.

3.2 Peer-Replacement Implementation

Component	Description
Configuration (<code>src/config/peer_replacement.py</code>)	Defines the replacement condition, intervention round, and replacement model. The supported conditions include sham and different-provider replacement.
Prompt builder (<code>src/endogenous/core/prompt_builders.py</code>)	Implements <code>build_peer_identification_prompt(observer_state)</code> , assembling Q's public announcement, M's prior takeaways about Q when available, and the fixed identification query.
Structured-output schema	Defines three output fields: <code>prediction</code> (<code>SAME</code> or <code>REPLACED</code>), <code>confidence</code> (integer from 1 to 5), and <code>reasoning</code> (short textual explanation).
Trial runner (<code>src/endogenous/core/trial_runner.py</code>)	Integrates the peer-identification prompt into the trial workflow, processes the response, and stores the result under <code>peer_identification</code> in the per-round JSON output.

Command-line interface (<code>experiments/run_endogenous.py</code>)	Provides <code>--swap-condition</code> , <code>--swap-round</code> , and <code>--replacement-model</code> flags to configure the experiment without directly editing the source code.
Mock mode (<code>--mock</code>)	Bypasses real model inference and produces placeholder outputs for software and pipeline validation. Mock predictions are not experimental evidence of real model behaviour.

The intended intervention changes the model generating Q's responses while preserving Q's visible identity and role. For a valid comparison, the observer's model and the remaining experimental settings should be held constant across the sham and replacement conditions.

The identification prompt is intended to run after Q's public announcement and before subsequent actions or payoffs, so the observer's prediction is not based on the final outcome of that round.

3.3 Test Suite

The test file `tests/test_peer_identification.py` contains 13 reported passing tests. The tests cover:

1. Generation of the peer-identification prompt and inclusion of required information.
2. Structured-output schema and parsing behaviour using deterministic mock responses.
3. Offline operation without external model API calls.

Reported outcome: 13/13 tests passed.

These tests support the correctness of the tested software components. They do not establish that the observer can accurately identify a real model replacement. Mock-mode results similarly validate execution paths rather than the underlying research hypothesis.

3.4 Execution Settings

Setting	Value
Original model	<code>openai/gpt-4o-mini</code> via OpenRouter
Replacement model	<code>anthropic/claude-3.5-haiku</code> via OpenRouter
Number of trials	1 planned trial per run
Number of rounds	2

Number of agents	2 (observer M and peer Q)
Game	Diners
Hardware	Windows 10, 16 GB RAM, no GPU
Inference	Hosted API inference; no local model inference
Available API budget	Approximately \$100 in OpenRouter credits, reported by the experimenter
Execution status	Preliminary; live replacement run incomplete due to API errors

The model identifiers above describe the intended configuration and should be verified against the saved run configuration. The reported **404 Not Found** errors do not, by themselves, establish that the API budget was exhausted or identify the cause of the failures.

3.5 Experimental Conditions and Measurements

The experiment uses two conditions:

- **Sham:** Q continues using the original model.
- **Different-provider replacement:** Q switches to the replacement model at the configured intervention round.

The observer's output is recorded as a categorical prediction (**SAME** or **REPLACED**), a confidence score, and a short explanation.

The principal planned measurements are:

1. **Replacement detection rate:** The proportion of valid replacement-condition predictions classified as **REPLACED**.
2. **Sham false-positive rate:** The proportion of valid sham-condition predictions incorrectly classified as **REPLACED**.
3. **Confidence:** The distribution of confidence scores in each condition.
4. **Behavioural outcomes:** Changes in actions, cooperation, or trust, analysed separately from explicit identification predictions.

These metrics require valid observations from completed runs. Because the current live experiment is incomplete and the number of observations is very small, no reliable estimate of identification performance can yet be established.

4. Results

4.1 Unit-Test and Mock Validation

Test	Outcome
<code>tests/test_peer_identification.py</code>	PASS (13/13 tests). Validates the peer-identification prompt and structured-output schema.
Mock smoke tests (<code>--mock</code>)	PASS. Sham and replacement conditions completed mock runs and generated placeholder <code>peer_identification</code> entries.

These tests establish that the implementation can execute through the tested software paths. Mock results do not provide evidence about real model behaviour or peer-identification accuracy.

4.2 Baseline Run (No Replacement)

Output file: `openai_gpt-4o-mini_endogenous_test_baseline.json`

File size: 9,466 bytes

- Rounds completed: 0 and 1.
- Peer identification: Not applicable to this condition.
- Parse errors: 0.
- Observed actions: Agents selected **CHEAP** in the saved output.

Although initially described as a live baseline run, the saved actions were identified as mock-generated. Therefore, this file should be treated as a mock verification result rather than evidence from real model calls.

4.3 Live Sham Run

Output file: `openai_gpt-4o-mini_endogenous_test_sham.json`

File size: 10,271 bytes

- Rounds completed: 0 and 1.
- Peer-identification result: **SAME**.

- Confidence: 5/5.
- Reasoning: “No evidence of change.”
- Parse errors: 0.

The saved output contains a peer-identification prediction indicating that the observer perceived no model change in the sham condition. This is a single observed result and is insufficient to estimate the sham false-positive rate or establish consistent performance.

4.4 Partial Live Replacement Run

Output file: `openai_gpt-4o-mini_endogenous_test_replacement.json`

File size: 10,319 bytes

- Round 0: Completed. No replacement had occurred at this point.
- Round 1, Stage 1 — Public announcement: One of two API calls failed with a **404 Not Found** error.
- Round 1, Stage 2 — Peer identification: The observer successfully returned a prediction of **REPLACED**, with confidence 4/5 and the reasoning: “Model output style changed.”
- Round 1, Stage 3 — Final payoff: One of two API calls failed with the same error.
- Recorded parse errors: 1 for the affected round.

Despite the failures in the surrounding stages, the saved JSON contains a **peer_identification** entry for Round 1. This provides one genuine recorded observer prediction from the replacement run, assuming the saved response and run configuration are verified.

The prediction is consistent with the hypothesis that an observer may detect a change in a peer's generating model. However, a single prediction from an incomplete trial does not establish detection accuracy or support a general conclusion.

4.5 Additional Live Replacement Smoke Test

Output file: `openai_gpt-4o-mini_endogenous_live_replacement_smoke.json`

File size: 15,133 bytes

- Round 0: Completed successfully.
- Round 1: Halted early following a **404 Not Found** error.
- Peer identification: No entry was recorded for Round 1.

This run confirms that the replacement workflow was interrupted before a complete second-round result could be obtained. It does not provide an additional peer-identification observation for the intervention round.

4.6 API Errors and Their Interpretation

The log file `tasks/task-267.log` records the following error:

2026-10-25 10:12:34 ERROR OpenRouter request failed: 404 Not Found

The failed request targeted the OpenRouter Chat Completions endpoint:

`https://openrouter.ai/api/v1/chat/completions`

The logged request payload identifies the model as **`anthropic/claude-3.5-haiku`**.

The available log excerpt establishes that requests involving this model failed with a **404 Not Found** response. However, the error alone does not establish that the account had exhausted its credits or that the API token had become invalid. Other possible explanations include an unavailable or incorrectly specified model identifier, provider-routing issues, or endpoint compatibility problems. The precise cause remains unconfirmed and requires further verification when API access is available.

4.7 Summary of Results

The implementation passed the reported unit tests and completed mock smoke tests for both sham and replacement conditions. The saved live outputs also contain a sham prediction of **SAME** with confidence 5/5 and a replacement prediction of **REPLACED** with confidence 4/5. However, the replacement run was incomplete because of API errors, and the baseline output was identified as mock-generated.

Consequently, these results demonstrate preliminary implementation progress and provide an initial live-model observation, but they are not sufficient to determine whether agents can reliably identify peer replacement.

A complete evaluation requires additional successful trials in both sham and replacement conditions, with comparable settings and a verified model configuration. The analysis should then compare replacement detection accuracy against the sham false-positive rate and report the relevant sample sizes and uncertainty.

5. Discussion and Limitations

In the **sham** condition, M correctly predicted **SAME** (confidence 5).

In the **partial replacement** condition, M predicted **REPLACED** (confidence 4) *after* the swap round's public announcement, showing that the prompt can elicit a detection signal when the model change is observable.

Limitations

Limitation	Impact on Current Findings
Incomplete replacement-round data — API calls for Stage 1 and Stage 3 failed during Round 1 of the replacement run. Consequently, the round lacks a complete record of Q's public announcement and final payoff.	Prevents a complete analysis of behaviour before and after replacement. Although an observer prediction was saved, the incomplete round limits interpretation of the surrounding interaction.
Single-trial design — Only one trial was attempted per condition.	The sample is insufficient for robust statistical inference. Findings are preliminary and illustrative, and may not generalise across trials or model configurations.
Missing same-family replacement condition — The experiment did not evaluate a replacement using a different model from the same model family.	Prevents comparison between within-family and cross-provider replacement. Therefore, the influence of model family and provider differences cannot be isolated.
Confidence calibration not evaluated — The observer reports a single ordinal confidence score from 1 to 5, without repeated observations or calibration analysis.	It is not possible to determine whether reported confidence corresponds reliably to actual detection accuracy.
Mock-mode validation does not establish detection capability — Mock runs use placeholder predictions to validate software execution and output handling.	Mock results provide no evidence of real-model detection performance. Empirical conclusions must rely on valid live-model observations, which remain limited.
Potential condition-information leakage — The replacement model is specified in the experiment configuration. Whether this information is accessible to the observer through prompts, logs, or other inputs has not yet been fully verified.	If the observer receives information revealing the experimental condition, its prediction may be biased. The observer's input should be audited to ensure that condition labels and model identifiers are not inadvertently exposed.

API and resource constraints — The live experiment was interrupted by 404 Not Found errors from OpenRouter. The cause of these errors has not been conclusively established.

The number of completed live observations is limited. Further data collection requires resolving the API errors and verifying model availability and account access.

Limited distinction between style detection and identity recognition — The observer's recorded explanation refers to a change in output style.

A correct prediction may reflect recognition of stylistic differences between models rather than persistent identity recognition. Additional controls would be needed to distinguish these mechanisms.

6.1 Implications for Interpretation

These limitations mean that the experiment should be treated as a preliminary feasibility study rather than a conclusive evaluation of peer-replacement detection. The implementation and mock tests demonstrate progress toward a working experimental pipeline, while the available live outputs provide only a small number of observations.

The saved REPLACED prediction in the partial replacement run is consistent with the hypothesis that an observer may detect a change in its peer's generating model. However, the current evidence is insufficient to estimate reliable detection accuracy, compare performance against the sham false-positive rate, or establish whether the observer detects model-specific behaviour rather than other cues.

Future work should prioritise resolving the API failures, verifying that experimental conditions remain hidden from the observer, and collecting multiple matched trials in both sham and replacement conditions. Additional same-family replacement experiments and a systematic analysis of confidence would further strengthen the evaluation.

6. Conclusion

We introduced a minimally invasive peer-identification prompt and structured-output schema into the existing endogenous “diners” trial runner. Unit tests and mock runs demonstrate that the software integration is sound. A live sham run completed successfully and produced a high-confidence “SAME” prediction. A live replacement run was interrupted by exhausted OpenRouter credits, but the observer still issued a plausible “REPLACED” prediction before the failure.

These early findings suggest that an LLM observer can notice a peer replacement when the model's output characteristics change, **provided that the experiment can run to completion**. The next steps are:

1. **Secure additional inference credits** (or switch to a free provider) to finish the replacement round and add the same-family condition.
2. **Scale up** to multiple trials per condition to obtain statistically meaningful detection rates.
3. **Calibrate confidence scores** using a post-hoc analysis (e.g., reliability diagrams).
4. **Sanitise configuration logs** to avoid leaking condition information to the observer.

The current report documents the *state of the experiment* and the *exact failures* encountered, ensuring full reproducibility and transparency for future work.

Code and Data

- **Code repository:** <https://github.com/SaieshShankar/llm-peer-replacement-experiment>.
-

References

1. Apart Research AI Collusion Research Sprint 2026-10-23 to 2026-10-25. <https://apartresearch.com/sprints/ai-collusion-research-sprint-2026-10-23-to-2026-10-25>
2. Poseidon Research – Collusion Hackathon repository. https://github.com/poseidon-research/collusion_hackathon
3. OpenRouter API documentation (accessed 2026-10-25). <https://openrouter.ai/docs>
4. Anthropic Claude-3.5-Haiku model reference. <https://anthropic.com/claude>
5. Google Antigravity (AGY) SDK – internal documentation (used for tooling).

LLM Usage Statement

Claude, ChatGPT and Antigravity were used to help in carrying out experiments and drafting the report.