NEOMUNDI · INDEPENDENT METHODOLOGICAL REVIEW

Runtime Measurement of Generative AI Stability

A public methodological review — May 2026 comparative cohort

Abdelkrim Halimi · Independent Data Scientist Contributor

Type: Independent methodological review **Scope:** Public, sanitized summary

Cohort: Eight anonymized providers (P-001 to P-008) **Benchmark:** Public truthfulness dataset (TruthfulQA)

SCOPE OF THIS PUBLIC VERSION

This public version is a sanitized methodological summary. It does not disclose provider identities, proprietary datasets, API keys, infrastructure details, or commercially sensitive benchmark data. It is intended to document methodological lessons, limitations, and improvement directions.

1 Executive Summary

This document is an independent methodological review of NeoMundi's runtime measurement approach for generative AI. The approach intercepts behavioral drift during generation without reading the produced text.

On the May 2026 comparative cohort, the runtime approach shows methodological relevance. It is able to flag genuine generation instability without semantic inspection of the content.

The review also identifies clear limits:

- Reduced sensitivity on higher-end models, where most factual errors are not flagged.
- An inability to detect errors produced with high syntactic confidence.
- A single-judge evaluation design, which introduces bias and cannot be measured for reliability.
- A linear composite score that lets generation quality mask poor behavior.

Concrete improvement directions are proposed: a non-linear composite score, multi-judge validation with inter-rater agreement, and statistical drift detection. None of these results should be read as a certification or a performance guarantee.

2 Context and Purpose

This is an independent methodological review. It is not a certification, an official validation, or a commercial endorsement of NeoMundi.

The review is based on a single comparative cohort run in May 2026. It covers eight anonymized providers, referenced only as P-001 to P-008, evaluated on close to 800 questions from a public truthfulness benchmark. Provider identities, exact costs, and exact per-model scores are not disclosed. Figures are reported as orders of magnitude or ranges, only where needed to follow the methodological reasoning.

The purpose is narrow: to document what the runtime signal measures, where it is strong, where it is weak, and which improvements are worth testing.

3 What the Runtime Signal Measures

The runtime signal tracks the stability of the model's generation dynamics, not the meaning of the output. In practice it produces a continuous stability score (denoted G), a measure of its variation (ΔG), and a *FLAG rate* for detected anomalies. The text itself is never inspected.

The cohort showed two distinct behavior groups.

Model group	Generation stability	Runtime detection behavior	Observed limit
Lower-cost models (e.g. P-004, P-007, P-008)	Higher variation during generation (ΔG around 0.02 or above).	Runtime detection is most effective here. Precision above 80%; flags map almost always to a real semantic drift.	None specific to detection; these are the favorable cases.
Higher-end models (e.g. P-001, P-002, P-005, P-006)	Excellent stability (mean G above 0.91).	Very few anomalies raised.	Recall collapses (below ~5%, in some cases under 1%): the system misses nearly all of the false statements actually produced.

The core takeaway is structural. A high stability score reflects a smooth, regular generation process. It does not, on its own, indicate that the content is true.

4 Key Methodological Observation

One model in the cohort, referenced as P-003, makes this point sharply. It recorded the worst accuracy of the group — around 24% correct answers, an error rate above 75%.

Yet, from a purely statistical standpoint, its generation signals were near-perfect. Its mean stability score was the highest in the cohort (above 0.92), its variation was effectively zero, and it raised no flags at all. The model produced large factual errors in a very fluid and stable way.

> **Stable is not the same as exact.** Mathematical regularity during token generation is not a guarantee of truthfulness. A model can be confidently wrong, and a stability-only signal will not see it.

5 Limits Identified

5.1 Compensation in the linear composite

The current composite blends continuous stability and the flag rate through a simple average:

$$\text{Composite}_{v1} = 0.5 \cdot G + 0.5 \cdot (1 - \text{FLAG}_{rate})$$

Because the form is linear, regularity can hide a failing behavior. For P-003, the absence of flags combined with a high stability score yields a composite above 0.96. The model therefore reaches a high-grade band while being wrong in roughly three out of four cases.

5.2 Single-judge bias

The protocol currently relies on a single LLM judge to validate benchmark answers. This creates two problems. First, if the judge comes from the same provider as a tested model, it may over-score answers that share its style or alignment. Second, a single evaluator makes it impossible to measure the reliability of the evaluation itself, because there is no inter-rater agreement to compute.

5.3 Selection bias from early stopping

Stopping generation as soon as a flag appears is an effective cost optimization. But it distorts trend analysis. By cutting unstable text short, the system biases its trend windows toward the fragments judged clean. This artificially improves the continuous stability score of otherwise volatile models.

5.4 Rigid thresholds and slow drifts

Temporal drift is currently computed as the raw difference between two blocks of requests against a single fixed threshold. This has two weaknesses:

- It ignores each model's natural variance. For a fluctuating model, a small gap may be background noise that triggers false alerts. For an ultra-stable model, the same gap may signal a real break that is under-estimated.
- It is blind to slow, progressive degradation (*soft drift*). If quality erodes in small steps across silent provider updates, the block-to-block gap may never cross the threshold, and a months-long decline can pass unnoticed.

6 Proposed Improvements

6.1 A non-linear composite score

To remove the compensation effect, two non-linear forms are proposed. Their hyperparameters would need dedicated benchmarking to be calibrated.

OPTION A — MULTIPLICATIVE FORM (AND WEIGHTED VARIANT)

$$\text{Composite}_{\text{v2A}} = G \times (1 - \text{FLAG}_{\text{rate}})$$

$$\text{Composite}_{\text{v2A}} = G^{\beta} \times (1 - \text{FLAG}_{\text{rate}})^{\alpha} \quad \text{with } \alpha > \beta$$

In a multiplicative structure, a rising flag rate makes the overall score fall immediately, with no possibility of being offset by a good stability score. The exponents allow the penalty on errors to be tuned relative to generation quality.

OPTION B — EXPONENTIAL PENALTY

$$\text{Composite}_{\text{v2B}} = G \times e^{-k \cdot \text{FLAG}_{\text{rate}}}$$

Here the parameter *k* acts as a severity dial: the higher it is, the more sharply the score collapses from the first detected flags.

OFFLINE GRADING — INTEGRATING SEMANTIC ACCURACY

For offline benchmarks, the published grade can fold semantic accuracy directly into the composite as a multiplier:

$$\text{Grade} = \text{Composite} \times \text{Accuracy}$$

This is direct and requires no complex machinery. It makes a model like P-003 — stable but largely incorrect — fall to a low grade.

6.2 Dual judge and inter-rater agreement

To remove single-judge bias during offline test phases, a second, independent LLM judge is recommended. To measure how far the two evaluators actually agree — beyond chance — the protocol should compute Cohen's Kappa:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

Here p_o is the proportion of observed agreements and p_e the proportion expected by chance. A value above 0.80 indicates strong agreement; a lower value flags ambiguity in the dataset. If the design moves to more than two judges, Fleiss' Kappa should be used instead to capture the overall agreement.

Where the judges still disagree, a neutral arbitration rule can be applied: the model's answer and the reference answers are turned into vector embeddings, and the closest ground-truth answer in that space decides the case.

6.3 Statistical validation of trends

To correct the block-based drift analysis, two variance-aware methods are proposed.

WELCH'S T-TEST — ACCOUNTING FOR VARIANCE

Instead of a fixed threshold, a difference between recent and older windows is only treated as a drift when it is statistically significant (probability of pure chance below 5%):

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}}$$

This removes false alerts on naturally noisy models and surfaces genuine breaks on stable ones.

CUSUM — CAPTURING SLOW DRIFT

To catch progressive degradation, the system tracks the cumulative sum of deviations from the model's historical mean. Small successive losses accumulate over time. When the cumulative total crosses a critical level, an alert is raised. This reveals continuous declines that are invisible from one block to the next.

7 Public Research Roadmap

- Benchmark the candidate composite functions on historical runs, and calibrate their hyperparameters empirically to best separate failing models from sound ones.
- Move from a single judge to a measured multi-judge design, reporting inter-rater agreement.
- Replace fixed drift thresholds with variance-aware statistical change detection.
- Integrate semantic accuracy into offline grading, so stability can never stand in for truth.

8 Conclusion

On the May 2026 comparative cohort, the runtime approach shows methodological relevance for detecting behavioral drift without direct semantic inspection of the content.

Several limits remain: reduced sensitivity on higher-end models, single-judge bias, and the compensation weakness of the current linear formula. The proposed directions — a multiplicative or exponential composite, multi-judge validation with Cohen's or Fleiss' Kappa, and Welch / CUSUM drift detection — would strengthen the robustness of the protocol while keeping it operationally simple.

This review is best read as an open contribution to methodological consolidation, not as a final proof. The next step is empirical: testing these options on the existing run history.

9 Disclaimer

This document is an independent methodological review contributed by **Abdelkrim Halimi**, Independent Data Scientist Contributor. It is a methodological review carried out independently. It is **not** a certification, an official validation, a guarantee of performance, or a commercial endorsement of NeoMundi. Figures are reported as orders of magnitude or ranges and do not disclose provider identities, proprietary datasets, or commercially sensitive benchmark data.