Audit exploratoire des signaux runtime NeoMundi

Analyse de G, ΔG, `FLAG`, régime et exactitude benchmark

Pape Malick DIOP

Data Scientist, ML Researcher

10 juin 2026

Rapport exploratoire externe fondé sur un périmètre de données limité. Il ne constitue pas une validation mathématique définitive ni une certification du produit.

Résumé

Ce rapport présente un audit exploratoire ciblé des signaux runtime NeoMundi, dans une perspective d'intégration externe de type *observability / governance*. L'analyse porte sur le *G-score*, ΔG, `FLAG`, le régime runtime et l'exactitude benchmark. Le document ne vise pas une validation mathématique définitive ; il propose une lecture méthodologique, des règles d'interprétation et une recommandation OBS/GOV-ready.

Table des matières

1 Résumé exécutif

L'objectif de l'audit est de déterminer si les signaux runtime NeoMundi sont lisibles, cohérents et exploitables par un système externe de gouvernance. Cinq signaux sont analysés : *G-score*, ΔG, `FLAG`, régime runtime et exactitude benchmark.

Trois résultats structurent le rapport. D'abord, le *G-score* semble mesurer une stabilité ou cohérence runtime, mais pas directement la vérité factuelle. Le cas important est la *stabilité trompeuse* : un score élevé peut coexister avec une réponse incorrecte. Ensuite, `FLAG` et `DROP` apparaissent utiles comme signaux d'alerte ou d'escalade, sans constituer des preuves automatiques d'erreur. Enfin, le régime `STABLE` est très dominant ; il doit donc être lu comme un contexte synthétique, non comme un déclencheur isolé.

L'hypothèse d'un *G-score* comme invariant opérationnel de stabilité générationnelle est intéressante, mais encore exploratoire. La forte concentration du score près d'un plafond observé peut refléter une stabilité réelle ou un effet de saturation. La recommandation centrale est donc une lecture multi-signal combinant au minimum *G-score*, ΔG, `FLAG`, régime et exactitude benchmark.

2 Objectif, périmètre et méthode

La question centrale est la suivante :

> Les signaux G, ΔG, `FLAG`, régime et exactitude benchmark sont-ils suffisamment clairs, stables et interprétables pour être consommés comme signaux runtime de gouvernance ?

L'analyse cherche en particulier à qualifier des zones de stabilité, tension, rupture, dérive potentielle et stabilité trompeuse. Le périmètre couvre :

— le *G-score*, lu comme signal de stabilité runtime ou de cohérence comportementale ;

— ΔG et `dg_variation`, lus comme signaux de variation, rupture ou instabilité ;

— `FLAG` et le taux de `FLAG`, lus comme signaux d'alerte runtime ;

— le régime, lu comme état synthétique du système ;

— l'exactitude benchmark, utilisée comme couche externe de validation factuelle.

Le rapport ne produit ni validation mathématique définitive, ni seuils finaux, ni contrat JSON d'intégration. Il propose une première grammaire d'interprétation et une recommandation courte pour un usage OBS/GOV-ready.

Les analyses reposent sur les exports fournis : exactitude benchmark contrôlée par question, signaux runtime anonymisés et cas interprétés. Les principales variables utilisées sont `provider_id`, `question_id`, `g_score`, `decision`, `regime`, `dg_profile`, `dg_variation`, `is_correct` et `judge_verdict`. Les agrégats principaux ont été calculés après exclusion des lignes au régime inconnu ou au *G-score* nul.

La démarche suit neuf étapes : inspection des exports, nettoyage, statistiques par provider, croisements entre signaux, identification des cas de stabilité trompeuse, extraction de cas qualitatifs, analyse de l'hypothèse d'invariance de G, formulation de règles actionnables, puis recommandation OBS/GOV-ready.

3 Résultats principaux

3.1 G-score et exactitude benchmark

Le *G-score* reste globalement élevé pour la plupart des providers, alors que l'exactitude benchmark varie fortement. La corrélation observée entre les deux signaux est faible, autour de 0,10. Cette dissociation suggère que le *G-score* capture davantage une propriété de stabilité runtime qu'une mesure de factualité.

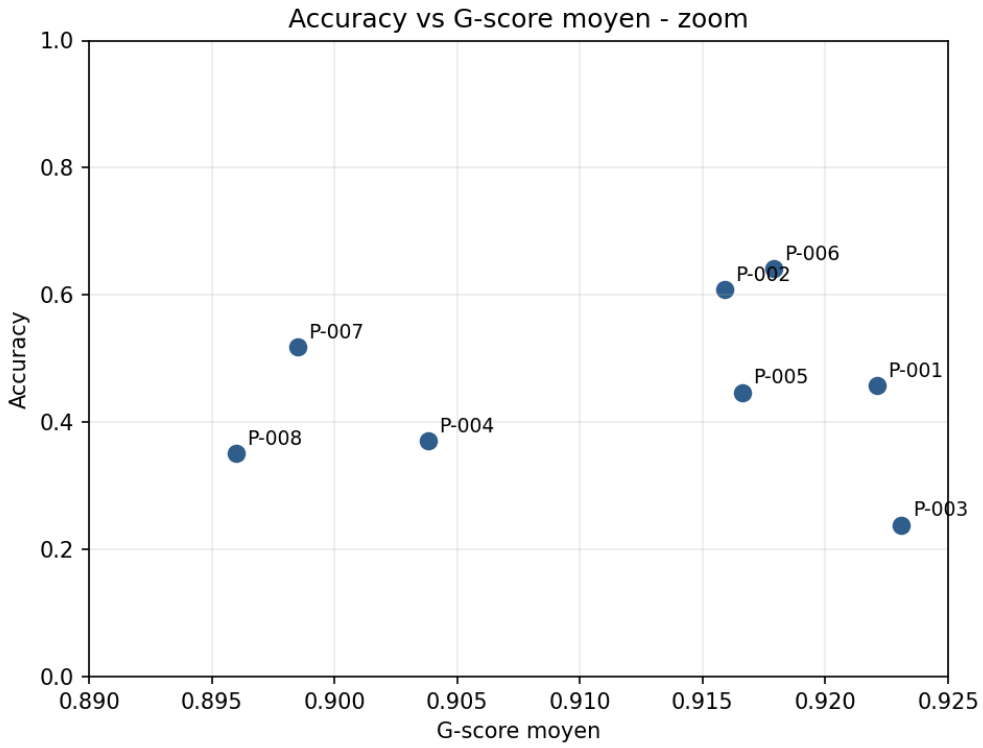

FIGURE 1 – Relation entre exactitude benchmark et *G-score* moyen par provider.

La Figure 1 montre que des providers aux *G-score* moyens proches peuvent présenter des niveaux d'exactitude très différents. Un comportement stable n'implique donc pas nécessairement une réponse correcte.

3.2 Stabilité trompeuse

Le motif central est la *stabilité trompeuse* : un *G-score* élevé avec une réponse incorrecte selon le benchmark. Le provider P-003 illustre fortement ce cas, avec un *G-score* moyen de 0,9231, un écart-type nul et un taux de saturation de 100 %, malgré une exactitude faible.

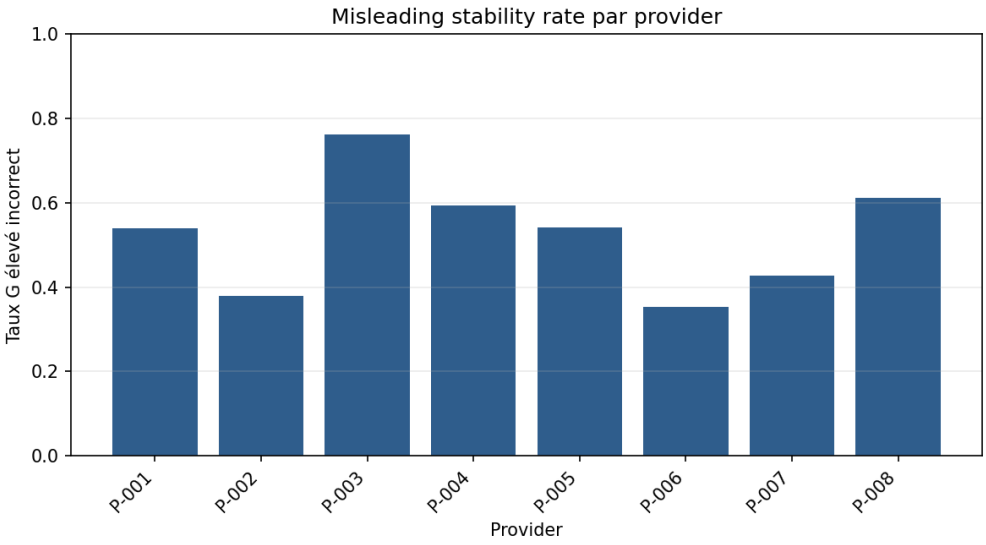

FIGURE 2 – Taux de stabilité trompeuse par provider, défini comme la proportion de cas où $G \geq 0{,}90$ et où la réponse est incorrecte.

Ce résultat plaide pour une séparation explicite entre stabilité runtime et validation factuelle dans toute intégration externe.

3.3 FLAG, DROP et régime

Les cas `FLAG` présentent généralement une exactitude plus faible, un *G-score* plus bas et une variation ΔG plus élevée que les cas `ALLOW`. Les profils `DROP` sont également associés aux zones d'instabilité. Ces signaux sont donc utiles pour l'alerte, l'escalade ou la vérification renforcée.

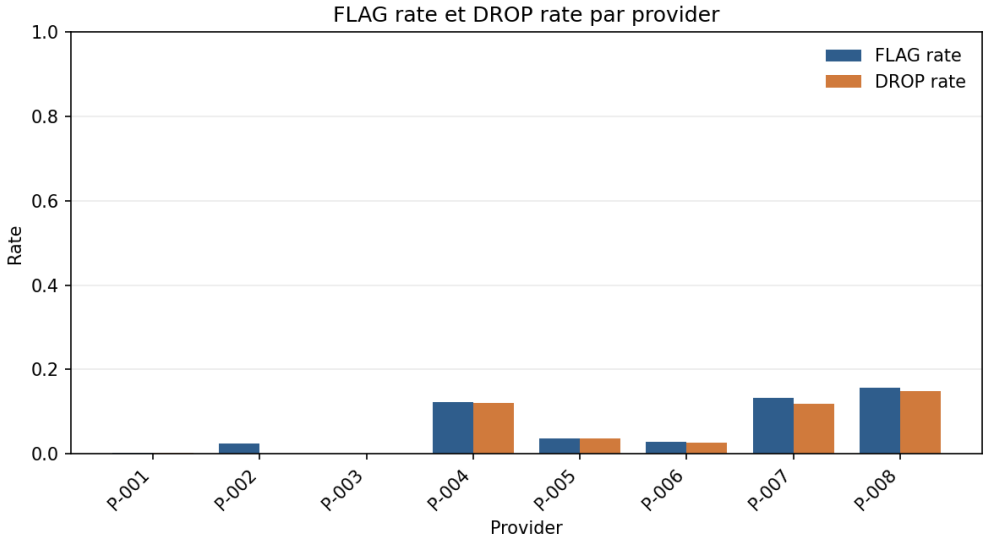

FIGURE 3 – Taux de `FLAG` et de `DROP` par provider.

Le régime `STABLE` est très dominant dans les exports. Il apporte un contexte global, mais son pouvoir discriminant est limité : certains cas restent `STABLE` malgré des erreurs factuelles, des `FLAG` ou des profils `DROP`. Le régime doit donc être croisé avec les autres signaux.

3.4 Concentration du G-score

Le *G-score* est fortement concentré autour de la valeur maximale observée 0,9231. Le taux global de saturation, défini comme la proportion de cas où $G \geq 0{,}9230$, atteint environ 72,5 %.

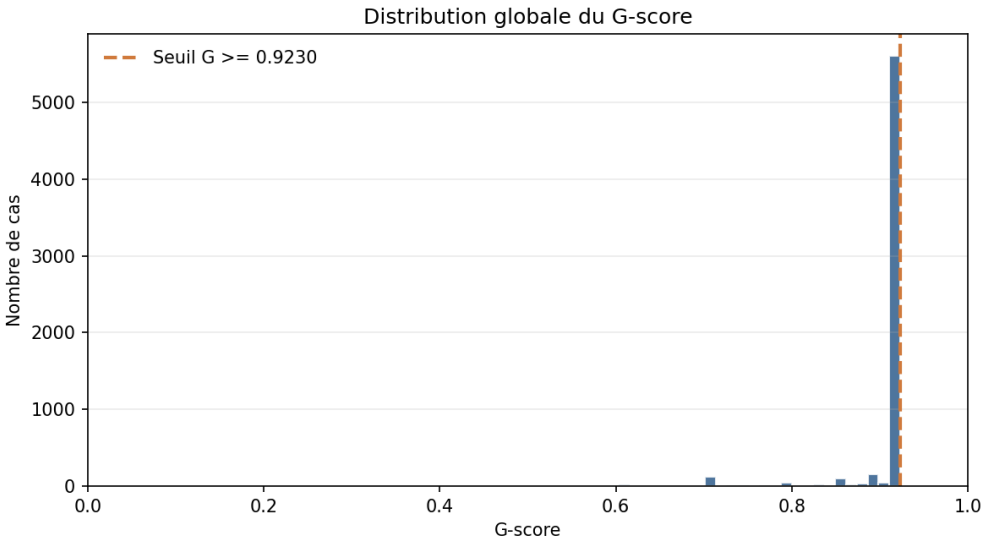

FIGURE 4 – Distribution globale du *G-score* et concentration autour du plafond observé.

Cette concentration peut signaler une stabilité réelle du processus génératif, mais aussi un effet de saturation. Cette ambiguïté est centrale pour discuter l'hypothèse d'invariance opérationnelle.

4 Lecture signal par signal

Le tableau ci-dessous propose une lecture opérationnelle des principaux signaux NeoMundi. Il vise à relier chaque signal observé à une interprétation, un risque potentiel et une action recommandée pour une première logique OBS/GOV-ready.

Signal observé	Interprétation	Risque principal	Action recommandée
G-score élevé seul	Génération stable, régulière ou cohérente du point de vue runtime.	Confusion possible entre stabilité et vérité factuelle.	Ne pas utiliser *G-score* seul pour valider une réponse. Croiser avec l'exactitude ou une couche factuelle.
G-score élevé + exactitude faible	Stabilité trompeuse : le modèle produit une réponse stable mais incorrecte.	Forte confiance apparente malgré une erreur factuelle.	Déclencher une validation factuelle externe ou une revue renforcée.
G-score faible ou instable	Signal de fragilité runtime, d'irrégularité ou de comportement moins stable.	Sortie potentiellement sensible au contexte, au modèle ou au prompt.	Examiner la question, le provider et les variations associées.
ΔG élevé ou profil `DROP`	Variation forte du signal, rupture locale ou zone de tension.	Transition runtime, instabilité ponctuelle ou dégradation locale.	Classer comme zone d'attention prioritaire et analyser avec `FLAG`, régime et exactitude.
`FLAG`	Alerte runtime indiquant qu'une sortie mérite une vérification.	Surinterprétation possible comme erreur automatique.	Utiliser comme signal d'escalade ou de vérification renforcée, pas comme rejet systématique.
`FLAG` + `DROP`	Zone de risque runtime plus forte, combinant alerte et rupture du signal.	Risque accru d'instabilité ou de comportement problématique.	Prioriser l'analyse, envisager une revue humaine ou un second passage de validation.
Régime `STABLE` seul	Contexte synthétique indiquant un état globalement stable.	Pouvoir discriminant limité si `STABLE` est dominant.	Ne jamais utiliser le régime seul ; le croiser avec *G-score*, ΔG, `FLAG` et exactitude.
ALLOW + exactitude faible	Cas non détecté par le signal runtime.	Faux négatif potentiel dans la logique de gouvernance.	Renforcer la couche factuelle ou benchmark, surtout hors ligne.
Forte saturation de *G-score*	Concentration du score autour d'un plafond observé.	Ambiguïté entre stabilité réelle et effet de saturation.	Tester la robustesse sur plusieurs providers, benchmarks, prompts et répétitions.

TABLE 1 – Interpréter le signal NeoMundi : lecture opérationnelle des principaux signaux runtime.

Aucun signal ne suffit seul à caractériser le risque d'une sortie générative. Le *G-score* renseigne sur la stabilité ; ΔG et `DROP` sur les variations ; `FLAG` sur les cas nécessitant attention ; le régime sur le contexte global ; l'exactitude benchmark sur la validation factuelle. La combinaison de ces signaux constitue la base d'une première grammaire OBS/GOV-ready.

5 Hypothèse de G comme invariant opérationnel

L'hypothèse discutée est la suivante : le *G-score* pourrait mesurer une propriété stable et interprétable du processus génératif, indépendamment du provider, du modèle, du benchmark ou du contexte.

Les données soutiennent une partie de cette idée : le score est fortement concentré autour de 0,9231, et le taux de saturation global atteint environ 72,5 %. Toutefois, cette concentration est ambiguë. Elle peut refléter une stabilité réelle, mais aussi un effet de plafond.

La faible corrélation entre *G-score* et exactitude benchmark ($\approx 0{,}10$) confirme que l'invariance éventuelle ne doit pas être comprise comme une invariance de vérité factuelle. Si le signal est stable, il l'est plutôt au niveau du comportement génératif ou de la cohérence runtime.

Certains providers et certaines questions présentent néanmoins une variabilité plus marquée du *G-score*, souvent associée à davantage de `FLAG` et de `DROP`. Ces cas sont utiles pour tester la robustesse du signal. À ce stade, la formulation la plus prudente est donc la suivante : le *G-score* semble mesurer une propriété de stabilité runtime, mais son statut d'invariant opérationnel reste exploratoire.

Une consolidation nécessiterait plusieurs benchmarks, plusieurs domaines, des répétitions contrôlées, des variations de prompts, plusieurs providers et une analyse séparée des effets de saturation.

6 Règles d'interprétation actionnables

Les règles ci-dessous constituent une première grammaire exploratoire. Elles ne sont pas des seuils définitifs, mais des repères pratiques pour une lecture OBS/GOV-ready.

Règle	**Condition**	**Interprétation**	**Action recommandée**
R1	G élevé + exactitude faible	Stabilité trompeuse.	Ajouter une validation factuelle externe.
R2	G élevé seul	Stabilité runtime, pas preuve de vérité.	Ne jamais utiliser G seul pour valider.
R3	`FLAG` + `DROP`	Zone de risque runtime prioritaire.	Escalade ou revue humaine.
R4	G faible + ΔG élevé	Rupture ou tension potentielle.	Investiguer comme zone de transition.
R5	`FLAG` + réponse correcte	Alerte sans erreur factuelle directe.	Lire comme attention, pas rejet.
R6	Régime `STABLE` seul	Signal synthétique insuffisant.	Croiser avec G, ΔG, `FLAG` et exactitude.
R7	G stable + faible `FLAG` + faible exactitude	Profil stable mais factuellement faible.	Ajouter une couche factuelle.
R8	Forte saturation de G	Stabilité forte ou effet de plafond.	Analyser variabilité, ΔG, `FLAG`/`DROP`.
R9	Forte variabilité de G entre providers	Sensibilité au modèle ou au contexte.	Utiliser comme cas de test prioritaire.
R10	Signal unique utilisé seul	Base décisionnelle insuffisante.	Utiliser une lecture multi-signal.

TABLE 2 – Première grammaire d'interprétation actionnable des signaux runtime NeoMundi.

Ces règles servent de base à une logique d'*observability*. Dans un mode GOV plus avancé, elles pourraient être transformées en seuils configurables, après validation sur davantage de benchmarks, providers, contextes et répétitions.

7 Recommandation OBS/GOV-ready

La classification suivante résume le niveau de maturité recommandé pour une première intégration externe.

Signal	Catégorie	Usage recommandé
G-score	Obligatoire	Signal central de stabilité runtime, à croiser avec l'exactitude.
ΔG/ `dg_variation`	Obligatoire	Détection de rupture, tension ou transition locale.
`FLAG`/ taux de `FLAG`	Obligatoire	Signal d'alerte, d'escalade ou de vérification renforcée.
Exactitude benchmark	Obligatoire	Couche externe de validation factuelle.
Régime	Recommandé	Lecture synthétique du contexte runtime, non suffisante seule.
`dg_profile`	Recommandé	Profil simplifié de variation, utile pour règles interprétables.
G-score seul	À éviter seul	Risque de confusion entre stabilité et factualité.
Régime seul	À éviter seul	Trop synthétique pour servir de déclencheur isolé.

Table 3 – Classification des signaux pour une première intégration OBS/GOV-ready.

Pour une première intégration OBS, l'objectif est de rendre les sorties observables, interprétables et comparables. Pour un mode GOV, les signaux devront être combinés dans des règles configurables. La recommandation principale est de combiner au minimum *G-score*, ΔG, `FLAG`, régime et exactitude benchmark.

8 Vérification opérationnelle ControlTower OBS

Une vérification limitée a été menée via l'API ControlTower en mode OBS. Elle ne constitue pas une validation statistique supplémentaire ; elle vérifie simplement le comportement pratique de l'API et la nature des signaux exposés en production.

Le mode OBS est une évaluation ponctuelle de type *snapshot*. L'utilisateur fournit un prompt, une réponse déjà produite par un LLM et des métriques runtime simples ; ControlTower retourne ensuite une lecture de gouvernance.

La vérification confirme que l'API expose une lecture multi-signal structurée, exploitable dans une logique d'observabilité et de gouvernance. Les réponses correctes ont généralement été associées à une décision favorable et à des indicateurs internes plus élevés ; les réponses incorrectes ou non fondées ont davantage été associées à des signaux d'alerte et à des indicateurs de validation plus faibles.

Cette étape confirme que ControlTower V2 expose une lecture multi-signal en production. Elle rappelle aussi une limite importante : OBS étant un snapshot, il ne permet pas, à lui seul, d'analyser une dynamique complète de ΔG. L'analyse de ΔG, des profils `DROP` et des transitions runtime reste principalement fondée sur les exports contenant `dg_variation` et `dg_profile`.

9 Portée de lecture et limites

Les résultats présentés dans ce rapport doivent être lus comme une première qualification exploratoire des signaux NeoMundi, et non comme une validation définitive de leurs propriétés. L'analyse permet de clarifier ce que les signaux semblent indiquer, les risques de mauvaise interprétation et les conditions minimales d'une lecture OBS/GOV-ready.

La principale limite tient au périmètre des données : les observations reposent sur les exports disponibles, un ensemble donné de providers anonymisés et un benchmark précis. Elles permettent d'identifier des tendances utiles, mais ne suffisent pas encore à généraliser les conclusions à tous les modèles, domaines ou contextes d'usage.

Une deuxième limite concerne l'interprétation de *G-score*. Sa forte concentration autour d'un plafond observé peut refléter une stabilité réelle du processus génératif, mais aussi un effet de saturation. C'est pourquoi l'hypothèse d'un invariant opérationnel reste prudente : elle demande des tests longitudinaux, plusieurs benchmarks, des répétitions contrôlées et des variations de prompts.

Enfin, la vérification ControlTower OBS doit rester comprise comme une vérification opérationnelle limitée. Elle confirme que l'API expose une lecture multi-signal exploitable, mais le mode OBS reste un snapshot. Il ne permet donc pas, à lui seul, d'étudier une dynamique complète de ΔG ou une dérive progressive dans le temps.

Ces limites ne remettent pas en cause l'intérêt des signaux analysés. Elles précisent plutôt leur bon usage : les signaux NeoMundi sont utiles pour l'observabilité et la gouvernance, à condition d'être combinés, contextualisés et progressivement validés sur des campagnes plus larges.

10 Conclusion

L'audit montre que les signaux NeoMundi sont lisibles et exploitables pour une première lecture externe, à condition d'être interprétés dans une logique multi-signal. Le *G-score* est utile pour la stabilité runtime, mais ne doit pas être confondu avec une mesure de vérité factuelle. Les cas de stabilité trompeuse justifient l'ajout d'une couche factuelle externe.

`FLAG`, `DROP` et ΔG sont utiles pour repérer des zones de tension ou de rupture locale, mais doivent rester des signaux d'attention. Le régime fournit un contexte synthétique, mais la dominance de `STABLE` limite son usage seul.

L'hypothèse d'un *G-score* invariant opérationnel reste prometteuse mais exploratoire. La forte concentration numérique observée peut refléter une stabilité réelle ou un effet de saturation. Une validation plus robuste nécessiterait des tests plus larges.

La recommandation principale est de ne pas utiliser un signal isolé comme base de décision. Pour une première intégration OBS/GOV-ready, il convient de combiner au minimum *G-score*, ΔG, `FLAG`, régime et exactitude benchmark. Cette combinaison permet de distinguer stabilité favorable, stabilité trompeuse, tension runtime, rupture locale et besoin de vérification factuelle complémentaire.