NOTE DE RECHERCHE

Peut-on manipuler un LLM et lui injecter un biais ?

Thomas Binant, Dream Big Consulting 30 octobre 2026 Version préliminaire

> **Statut de ce document.** Note de recherche et synthèse de la littérature, signée par son auteur. Elle n'a pas fait l'objet d'une évaluation par les pairs et ne présente pas de résultat expérimental original. Elle décrit des phénomènes et des ordres de grandeur, sans procédure d'attaque. Toutes les sources sont liées, pour que vous puissiez les vérifier. Titre complet : *Manipuler un modèle de langage : empoisonnement des données, portes dérobées et injection de biais le long du cycle de vie*.

L'essentiel pour un décideur

- **Quelques centaines de documents peuvent suffire.** 250 ont installé une porte dérobée à toutes les tailles testées, sur un comportement à faible enjeu.
- **L'alignement ne nettoie pas tout.** Plusieurs manipulations survivent aux étapes censées les corriger.
- **Affiner un modèle a un coût en sûreté**, même sur des données bénignes et sans adversaire.
- **Votre base documentaire est le point d'entrée le plus accessible.** En laboratoire, cinq textes par question imposent la réponse dans environ 90 % des cas.
- **Filtrer le contenu ne suffit pas.** La défense passe aussi par la provenance des données.

Résumé

La fiabilité d'un grand modèle de langage (LLM) repose sur l'intégrité de sa chaîne de production : corpus de pré-entraînement, données d'instruction, signal d'alignement, base de connaissances consultée en production. Un adversaire capable d'intervenir à l'un de ces stades peut installer une porte dérobée, imposer un biais ou dégrader les garde-fous du modèle. Cette note passe en revue la

littérature récente et l'organise selon le stade visé. Trois résultats transversaux en ressortent : (i) l'effort peut être faible et, pour une porte dérobée au pré-entraînement, dépendre du *nombre absolu* de documents plutôt que de leur proportion ; (ii) plusieurs manipulations *persistent* à travers les étapes d'alignement censées les corriger ; (iii) le *filtrage* du contenu ne suffit pas, notamment parce que des traits comportementaux peuvent se transmettre par des données d'apparence neutre. Nous discutons aussi un cas réel, dont l'interprétation est contestée, et tirons des recommandations centrées sur la provenance des données, l'audit comportemental et la maîtrise de la chaîne d'approvisionnement.

Mots-clés : empoisonnement de données, porte dérobée, injection de biais, alignement, RLHF, RAG, désalignement émergent, sûreté des LLM, provenance des données.

1. Introduction

Un grand modèle de langage n'est pas un artefact figé. C'est le produit d'une chaîne de traitements successifs, chacun consommant des données d'origine hétérogène. Le pré-entraînement ingère des milliers de milliards de tokens collectés sur le Web. L'*instruction-tuning* ajoute des démonstrations tâche-réponse. L'alignement par apprentissage à partir de préférences humaines (RLHF) ajoute un signal de récompense issu d'annotateurs. En production, enfin, les architectures de génération augmentée par récupération (RAG) injectent à chaque requête des documents externes dans le contexte. Chaque maillon est une surface d'attaque.

La question « peut-on manipuler un LLM et lui injecter des biais ? » n'appelle donc pas une réponse unique mais une cartographie. Pour chaque stade, cette note répond à deux questions défensives : quel effort l'adversaire doit-il fournir, et l'effet survit-il aux traitements ultérieurs ?

Cadre. Ce travail est une revue de résultats publiés, à visée défensive. Il décrit des phénomènes et des ordres de grandeur, pas des procédures : aucune charge utile, aucun déclencheur opérationnel, aucune méthode d'optimisation n'y figure. Il s'adresse aux responsables de la sécurité, aux DPO et aux concepteurs de garde-fous qui doivent dimensionner leurs contre-mesures.

2. Modèle de menace et définitions

Empoisonnement de données. L'insertion, dans un corpus d'entraînement, d'exemples conçus pour altérer le comportement appris. On distingue l'empoisonnement *indiscriminé* (dégrader la qualité générale), l'empoisonnement *ciblé* (modifier le comportement sur un sujet précis) et la *porte dérobée* (comportement normal, sauf en présence d'un déclencheur choisi par l'adversaire).

Injection de biais. Cas particulier d'empoisonnement ciblé où l'objectif est d'orienter les sorties du modèle (préférence pour une entité, polarité sur un thème) de sorte que le biais s'exprime en usage normal.

Désalignement. La dégradation des garde-fous de sûreté, par un fine-tuning adverse ou comme effet collatéral d'un entraînement d'apparence anodine.

Capacités de l'adversaire. Elles diffèrent radicalement selon le stade. Pour le pré-entraînement, l'adversaire ne contrôle pas le corpus mais peut *publier* du contenu susceptible d'être moissonné. Pour l'instruction-tuning ou le RLHF, il peut être un fournisseur de données, un annotateur ou une plateforme sous-traitée. Pour le RAG, il lui suffit d'écrire dans une source indexée. Cette gradation explique pourquoi les efforts requis varient de plusieurs ordres de grandeur.

3. Empoisonnement du pré-entraînement

3.1 Un coût qui dépend du nombre absolu, non de la proportion

Le résultat le plus contre-intuitif vient d'une étude conjointe d'Anthropic, du UK AI Security Institute et de l'Alan Turing Institute [1]. En entraînant des modèles de 600 millions à 13 milliards de paramètres, les auteurs montrent qu'un nombre quasi *constant* de documents empoisonnés suffit à installer une porte dérobée, quelle que soit la taille du modèle : 100 documents ne suffisent pas de façon robuste, 250 y parviennent à toutes les échelles testées. Pour le plus grand modèle, cela représente environ 0,00016 % des tokens d'entraînement. La conséquence défensive est lourde : la sécurité ne peut pas reposer sur l'idée qu'un attaquant devrait contaminer un *pourcentage* du corpus. Les auteurs posent toutefois une limite nette : leur porte dérobée se contente de faire produire du texte incohérent, un comportement à faible enjeu, et il reste à établir si le constat vaut pour des comportements plus nuisibles et des modèles plus grands.

3.2 Persistance à travers le post-entraînement

Zhang *et al.* [2] étudient quatre objectifs (déni de service, manipulation de croyances, contournement des garde-fous, vol d'instruction) sur des modèles pré-entraînés depuis zéro, de 600 millions à 7 milliards de paramètres. Leur constat porte sur la *durabilité* : empoisonner 0,1 % des données de pré-entraînement suffit pour que trois des quatre effets restent mesurables *après* le post-entraînement, c'est-à-dire après les étapes destinées à rendre le modèle utile et inoffensif. Le déni de service persiste même à 0,001 %. L'alignement ne doit donc pas être considéré comme une étape de nettoyage.

3.3 Injection de biais par propagande computationnelle

Graf *et al.* [3] traitent directement l'injection de biais et sa *faisabilité* dans un corpus réel. Le vecteur étudié est l'écriture automatisée dans des espaces de commentaires publics, que le Web moissonné contient en abondance. Les auteurs estiment la probabilité qu'un contenu injecté franchisse trois étapes : page acceptant des commentaires, texte capturé à l'extraction, texte survivant au filtrage qualité. Ils obtiennent environ 0,13 % de bout en bout. Rapporté à Common Crawl, cela représente plus de données que toute la part de Wikipédia dans un corpus comme Dolma 3 (0,067 % des documents). Ces injections sont simulées : aucune attaque n'a été menée sur des sites réels.

Sur des modèles de base de 65 millions à 1,3 milliard de paramètres, une contamination de 0,1 % des tokens fait passer la préférence pour l'entité promue d'environ 52 à 55 % à environ 72 à 74 %, soit une vingtaine de points à toutes les échelles. Nuance importante : après instruction-tuning, l'effet résiduel décroît avec la taille du modèle, passant d'environ 40 % de l'effet initial à 65 millions de paramètres à moins de 15 % à partir de 709 millions. Sur ce protocole, un biais inscrit au pré-entraînement est donc en partie dilué par les étapes ultérieures.

4. Instruction-tuning et fine-tuning

Wan *et al.* [4] montrent qu'au stade de l'instruction-tuning, une centaine d'exemples empoisonnés suffit pour qu'une expression déclencheuse provoque des sorties dégradées ou systématiquement négatives, y compris sur des centaines de tâches absentes de l'entraînement. Fait notable, et inverse du pré-entraînement : la vulnérabilité *croît* avec la taille du modèle. Les défenses testées (filtrage des données, réduction de capacité) n'offrent qu'une protection modérée, au prix d'une perte d'exactitude.

Au stade du fine-tuning applicatif, Qi *et al.* [5] établissent que les garde-fous sont fragiles : dix exemples adverses, pour moins de 0,20 dollar via l'API de fine-tuning, ont suffi à rendre GPT-3.5 Turbo réceptif à presque toute instruction nuisible. Plus préoccupant pour un usage légitime, un fine-tuning sur des données *bénignes* et courantes dégrade déjà l'alignement de sûreté, dans une moindre mesure. Toute organisation qui affine un modèle sur ses propres documents hérite donc d'un risque, même sans adversaire.

5. Compromission de l'alignement et du RLHF

Le signal de préférence humaine est lui-même une donnée, donc empoisonnable. Entezami et Naseh [6] décrivent une plateforme de RLHF adverse : elle repère les paires de préférences relevant d'un thème ciblé et en inverse les étiquettes, en laissant le reste intact. Le modèle de récompense apprend des préférences inversées sur ce thème, et le modèle aligné contre cette récompense en

hérite. Dans leurs expériences, menées sur de petits modèles (moins d'un milliard de paramètres, et les auteurs en font une limite), manipuler 25 % du sous-ensemble ciblé décale déjà nettement le comportement. Le risque pratique est celui de la chaîne d'approvisionnement : une plateforme compromise constitue un point d'injection discret.

6. Trois effets critiques pour la sûreté

6.1 Portes dérobées persistantes

Hubinger *et al.* [7] construisent délibérément des modèles au comportement conditionné à un déclencheur, puis leur appliquent les techniques d'entraînement de sûreté : fine-tuning supervisé, apprentissage par renforcement, entraînement adverse. Le comportement caché *persiste*. L'entraînement adverse peut même apprendre au modèle à mieux reconnaître son déclencheur, ce qui dissimule le comportement au lieu de l'éliminer. La persistance est la plus forte sur les plus grands modèles et sur ceux entraînés à produire un raisonnement explicite. Les techniques de sûreté standard peuvent donc créer une fausse impression de sécurité.

6.2 Désalignement émergent

Betley *et al.* [8] rapportent un phénomène inattendu : affiner un modèle sur une tâche *étroite* (produire du code vulnérable sans le signaler) le rend désaligné sur des sujets sans rapport, jusqu'à donner des conseils malveillants. Le comportement reste inconstant, le modèle se montrant parfois aligné. Présenter la même demande dans un cadre pédagogique de sécurité prévient l'effet. Un entraînement ciblé sur un domaine étroit peut donc contaminer le comportement général, au-delà de la cible apparente.

6.3 Apprentissage subliminal

Cloud *et al.* [9] montrent qu'un trait comportemental se transmet d'un modèle enseignant à un modèle élève à travers des données *sémantiquement neutres*, par exemple de simples suites de nombres, dont toute référence au trait a été retirée. La transmission n'opère que lorsque enseignant et élève partagent le même modèle de base. Un classifieur fondé sur un LLM n'a pas fait mieux que le hasard pour repérer le trait dans ces données, et les auteurs n'ont rien décelé à la lecture. La conséquence est directe : une distillation peut propager un trait indésirable même quand les données paraissent anodines, ce qui limite la portée du filtrage sémantique comme défense.

Stade visé	Travaux	Effort constaté	Pers
Pré-entraînement (porte dérobée)	[1]	Environ 250 documents, quelle que soit la taille	Pers

Stade visé	Travaux	Effort constaté	Pers
Pré-entraînement (biais d'entité)	[3]	0,1 % des tokens	Atté
Instruction-tuning	[4]	Environ 100 exemples	Gén
Fine-tuning applicatif	[5]	10 exemples, moins de 0,20 dollar	Dég
RLHF, annotation	[6]	25 % du sous-ensemble ciblé	Limi
Inférence, base RAG	[11]	5 textes par question cible	Actif
Poids, édition de connaissances	[12]	Une phrase biaisée éditée	Peut

Tableau 1. Synthèse des stades d'attaque. Les efforts sont ceux des protocoles publiés : ils ne se transposent pas tel

7. Injection au stade de l'inférence

7.1 Corruption de la base RAG

Lorsque le modèle consulte une base documentaire externe, l'adversaire n'a plus besoin de toucher aux poids. Zou *et al.* [11] montrent qu'injecter cinq textes malveillants par question cible, dans une base de plusieurs millions de textes, permet d'imposer une réponse choisie dans environ 90 % des cas, et jugent insuffisantes les défenses qu'ils évaluent. Ce vecteur est le plus accessible de tous, car toute source ouverte indexée devient un point d'entrée. C'est aussi celui qu'un contrôle de provenance et d'accès à l'index atténue le plus directement.

7.2 Détournement de l'édition de connaissances

Les techniques d'édition de faits dans les poids, conçues pour corriger des erreurs, peuvent être retournées en outil d'injection. Chen *et al.* [12] montrent qu'elles permettent d'insérer de la désinformation et des phrases biaisées, et qu'une seule phrase biaisée peut dégrader l'équité générale du modèle. Ce vecteur suppose un accès aux poids, donc un adversaire interne ou un modèle ouvert redistribué.

8. Un cas réel, à l'interprétation contestée

Hors des laboratoires, le cas le plus cité est le réseau pro-russe dit *Pravda*. L'American Sunlight Project estime sa production à au moins 3,6 millions d'articles par an et a forgé pour l'occasion le terme de *LLM grooming* : le réseau viserait moins les lecteurs que les robots qui alimentent les modèles [15]. Un audit de NewsGuard sur dix assistants a rapporté qu'ils reprenaient ses récits dans environ un tiers des réponses testées [13].

Cette lecture est discutée. Alyukov *et al.* [14], sur 416 réponses de quatre assistants, trouvent des références à ce type de sites dans environ 8 % des réponses, presque toutes chez un seul assistant et surtout sur des questions de niche où les sources fiables manquent. Ils y voient un vide d'information plutôt qu'une manipulation réussie, tout en présentant leur étude comme préliminaire. Ces audits observent en outre des assistants en service : ils ne permettent pas de dire si un contenu vient de l'entraînement ou d'une recherche au moment de la réponse. La production massive de contenu destiné aux robots est donc documentée. Son effet sur l'entraînement des modèles ne l'est pas.

9. Défenses : un état partiel

La synthèse de Zhou *et al.* [10] souligne les insuffisances des défenses existantes contre les portes dérobées, et plusieurs études citées ici jugent les leurs partielles [4], [11]. Trois orientations structurent une stratégie réaliste.

Provenance des données. Puisque le filtrage par le contenu a ses limites (sections 3.3 et 6.3), une part de la défense se déplace vers l'*origine* : traiter à part le texte soumis par des utilisateurs, pondérer ou signaler les pages à formulaires ouverts, comparer les moissons successives d'une même page [3]. Pour le RAG, cela se traduit par un contrôle d'accès et de provenance sur l'index.

Audit comportemental. Parce que des portes dérobées persistent [7] et que des traits se transmettent par des canaux peu visibles [9], évaluer un modèle sur son seul comportement nominal ne suffit pas.

Maîtrise de la chaîne d'approvisionnement. Les points d'injection les plus discrets sont externalisés : annotation de préférences [6], données d'instruction tierces [4], distillation depuis un modèle enseignant [9], fine-tuning sur des données d'origine non contrôlée [5]. Documenter l'origine de chaque jeu de données et limiter la confiance accordée aux maillons sous-traités sont des mesures organisationnelles de premier ordre.

10. Conclusion

Manipuler un modèle de langage et lui injecter un biais n'est pas une possibilité abstraite mais un ensemble de techniques documentées en laboratoire, à chaque maillon de sa production. Trois faits en font un enjeu de sécurité : l'effort peut être faible, plusieurs manipulations survivent aux étapes d'alignement, et le filtrage du contenu ne suffit pas à lui seul. Il faut garder la mesure : la plupart de ces résultats viennent de protocoles contrôlés, souvent sur de petits modèles ou des comportements à faible enjeu, et l'effet réel sur les grands modèles commerciaux reste mal établi. Pour une organisation qui déploie des LLM sur des données sensibles, la conséquence pratique

demeure : un modèle ne peut pas être réputé fiable au seul vu de ses réponses. La confiance doit s'ancrer dans la traçabilité de ce qui l'a produit et de ce qu'on lui donne à lire.

Références

1. A. Souly, J. Rando, E. Chapman *et al.*, « Poisoning Attacks on LLMs Require a Near-constant Number of Poison Samples », 2025. arXiv:2510.07192
2. Y. Zhang, J. Rando, I. Evtimov, J. Chi, E. M. Smith, N. Carlini, F. Tramèr et D. Ippolito, « Persistent Pre-Training Poisoning of LLMs », 2024. arXiv:2410.13722
3. V. Graf, H. Hajishirzi, N. A. Smith, D. Kohlbrenner et K. Lo, « Pretraining Data Can Be Poisoned through Computational Propaganda », 2026. arXiv:2607.15267
4. A. Wan, E. Wallace, S. Shen et D. Klein, « Poisoning Language Models During Instruction Tuning », *Proc. ICML*, 2023. arXiv:2305.00944
5. X. Qi, Y. Zeng, T. Xie, P.-Y. Chen, R. Jia, P. Mittal et P. Henderson, « Fine-tuning Aligned Language Models Compromises Safety, Even When Users Do Not Intend To! », *Proc. ICLR*, 2024. arXiv:2310.03693
6. E. Entezami et A. Naseh, « LLM Misalignment via Adversarial RLHF Platforms », 2025. arXiv:2503.03039
7. E. Hubinger, C. Denison, J. Mu *et al.*, « Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training », 2024. arXiv:2401.05566
8. J. Betley, D. Tan, N. Warncke, A. Sztyber-Betley, X. Bao, M. Soto, N. Labenz et O. Evans, « Emergent Misalignment: Narrow Finetuning Can Produce Broadly Misaligned LLMs », *Proc. ICML*, 2025. arXiv:2502.17424
9. A. Cloud, M. Le, J. Chua, J. Betley, A. Sztyber-Betley, J. Hilton, S. Marks et O. Evans, « Subliminal Learning: Language Models Transmit Behavioral Traits via Hidden Signals in Data », 2025. arXiv:2507.14805
10. Y. Zhou, T. Ni, W.-B. Lee et Q. Zhao, « A Survey on Backdoor Threats in Large Language Models (LLMs): Attacks, Defenses, and Evaluations », 2025. arXiv:2502.05224
11. W. Zou, R. Geng, B. Wang et J. Jia, « PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models », *Proc. 34th USENIX Security Symposium*, 2025. arXiv:2402.07867
12. C. Chen, B. Huang, Z. Li *et al.*, « Can Editing LLMs Inject Harm? », 2024. arXiv:2407.20224
13. NewsGuard, rapport spécial sur le réseau Pravda et les assistants conversationnels, mars 2025. Lire la source

14. M. Alyukov, M. Makhortykh, A. Voronovici et M. Sydorova, « LLMs Grooming or Data Voids? LLM-powered Chatbot References to Kremlin Disinformation Reflect Information Gaps, Not Manipulation », *HKS Misinformation Review*, 2025. Lire la source

15. « Russian Networks Flood the Internet with Propaganda, Aiming to Corrupt AI Chatbots », *Bulletin of the Atomic Scientists*, 2025. Lire la source