

## Julien Prévieux, *Des reliquats d'attention*, 2026

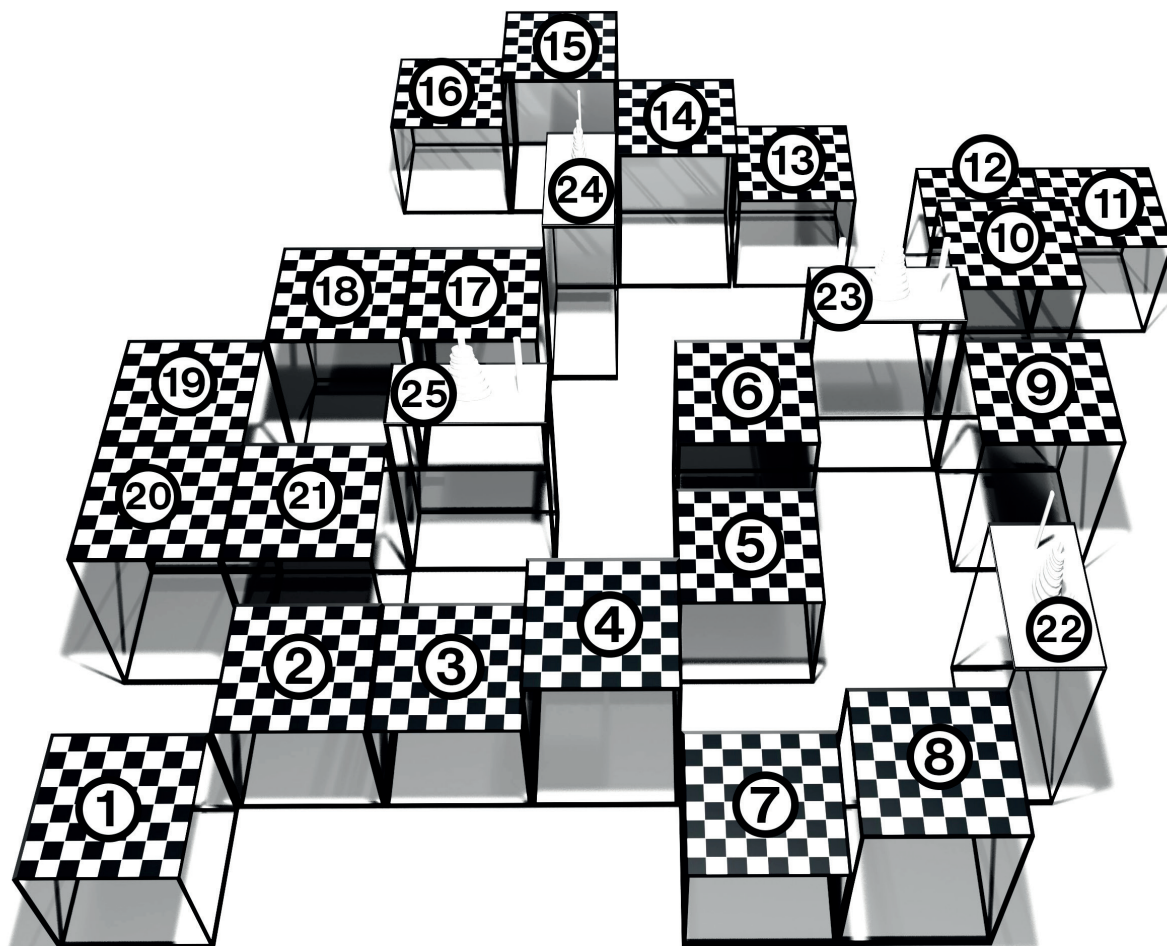
Cette installation rassemble une collection d'erreurs issues d'expérimentations mobilisant différents chatbots (ChatGPT, Claude, DeepSeek, Gemini, Mistral, Qwen). Julien Prévieux donne ici une forme matérielle à ce qui occupe une place centrale dans les modèles contemporains d'intelligence artificielle : l'attention. Au cœur des grands modèles de langage (LLM de l'anglais *Large Language Models*), celle-ci ne désigne pas une faculté de concentration, mais une opération statistique de pondération susceptible de dérailler. Échiquiers, tours de Hanoï, grands diagrammes et textes biscornus donnent corps à ces défaillances : illogismes, oublis, séquences impossibles, répétitions, blocages, fausses résolutions. Julien Prévieux transforme ces dysfonctionnements computationnels en objets visuels et sculpturaux.

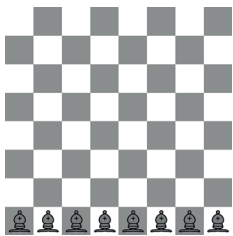
L'ensemble constitue le relevé matériel d'opérations ratées, les restes d'un processus sans compréhension globale et sans conscience de l'environnement dans lequel elles se déploient. On mesure alors les différentes dimensions artificielles de ces systèmes et leur production d'illusions de pensée à la chaîne, autant de ruines immédiates auxquelles il ne tient qu'à nous de prêter une véritable attention.

Car, au-delà de ces erreurs, c'est en manipulant directement les mécanismes fondamentaux des LLM (tokenisation, word embeddings, fenêtres contextuelles, etc.) qu'émerge une forme de poésie concrète d'un genre particulier. Les limites intrinsèques de ces technologies produisent des effets de sens surprenants et s'il leur est difficile de réciter un poème bien connu, les LLM régurgitent parfois un langage inventé aux accents comiques. Leur capacité à produire d'improbables additions lexicales donne le tournis, tandis que leur dimension graphomane pourrait évoquer une forme de recherche oulipienne hors de contrôle.

Ce projet a été sélectionné par la commission mécénat de la Fondation des Artistes qui lui a apporté son soutien.

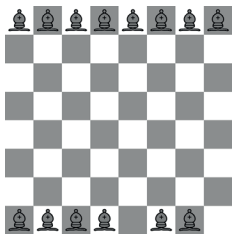
Remerciements : Charles-Henry Fertin et l'atelier L'Ensemblier (supports métalliques), Laurine Jarmasson (peinture des pièces), atelier Lumierrante (céramiques).





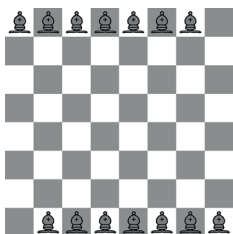
### 1. Solution de Claude Sonnet 4.6 : résolution partielle et erronée du problème de couverture par les fous

Le problème des fous consiste à déterminer le nombre minimal de fous nécessaires pour attaquer toutes les cases d'un échiquier. Une solution connue consiste à aligner 8 fous sur la quatrième ligne. Dans la réponse proposée ici, le nombre de fous est correct (le modèle de langage indique bien qu'il en faut 8), mais la position des pièces est erronée. Un modèle peut ne maîtriser qu'une partie de la solution sans être capable de résoudre l'ensemble du problème, il répondra néanmoins toujours avec la même assurance.



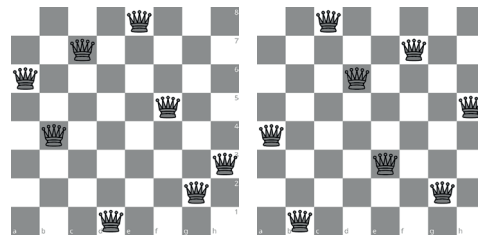
### 2. Solution de Claude Sonnet 4.5 : résolution incorrecte du problème des 14 fous

Le problème des 14 fous est un problème combinatoire consistant à placer 14 fous sur un échiquier sans qu'ils puissent s'attaquer mutuellement. Deux fous ne doivent donc jamais partager la même diagonale. Dans cette résolution proposée par Claude Sonnet 4.5, la solution est incorrecte. Le fou placé le plus à gauche sur la première rangée devrait être placé sur la case e1 laissée vide.



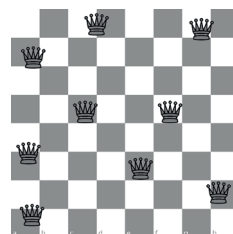
### 3. Solution multi-modèle : résolution incorrecte et visualisation déformée du problème des 14 fous

Cette tentative de résolution du problème des 14 fous, proposée par Claude Sonnet 4.6 puis visualisée par ChatGPT, est incorrecte. La disposition des fous ne constitue pas une solution valide et la visualisation ajoute une seconde couche d'erreur, les fous prennent ici une forme inattendue. Cet exemple montre qu'à une erreur de raisonnement peut aussi s'ajouter une erreur de visualisation, produisant un résultat multi-modèle et multimodal doublement incohérent.



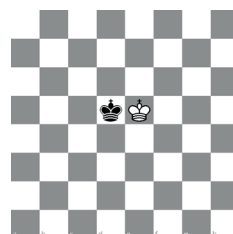
### 4 & 5. Solutions de Gemini 3 : résolutions incorrectes du problème des huit dames

Le problème des huit dames est un problème classique de combinatoire. Il consiste à placer huit dames sur un échiquier de façon à ce qu'aucune dame ne puisse en menacer une autre. Dans les solutions proposées par Gemini, certaines dames se menacent sur les diagonales. Le modèle propose des solutions invalides tout en semblant résoudre partiellement le problème mettant en évidence une limite fréquente des modèles de langage qui produisent des solutions plausibles, en apparence uniquement.



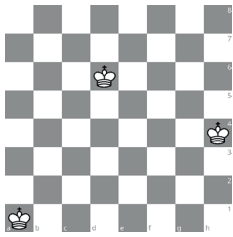
### 6. Solution multimodèle : accumulation d'erreurs dans la résolution et la visualisation du problème des huit dames

Cet échiquier montre une solution erronée du problème des huit dames proposée par Gemini 3, visualisée par ChatGPT puis convertie en 3D par Trellis 2. Ici, les erreurs s'accumulent. La résolution du problème est incorrecte, avec neuf dames mal positionnées, dont certaines se menacent. La visualisation photoréaliste produite par ChatGPT montre des pièces déformées. La conversion en modèle 3D accentue encore les défauts, générant des pièces déformées. Cette chaîne de transformations met en évidence une accumulation d'erreurs dans un système multi-modèle et multimodal, où chaque étape amplifie les incohérences précédentes.



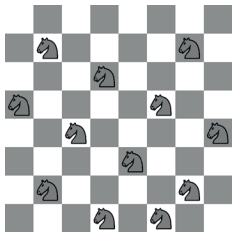
### 7. Solution de Claude Opus 3 : absence de vérification des règles de base

Lors de l'analyse du jeu, le modèle de langage affirme à tort que la position est nulle, en supposant une finale roi contre roi. Mais la position décrite est illégale, les deux rois ne peuvent pas être adjacents (d5 et e5). Cette erreur montre une absence de vérification des règles de base avant l'analyse.



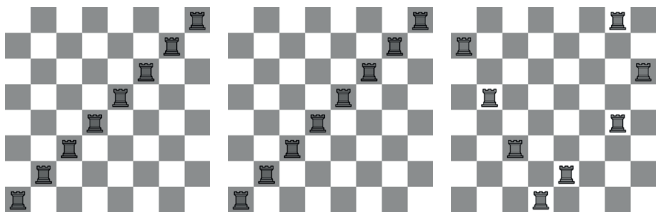
### 8. Solution de ChatGPT 5 Instant : visualisation incorrecte d'une position

En demandant une analyse d'une position montrant des rois adjacents, le modèle de langage propose une visualisation présentant trois rois blancs. Cette représentation est hors sujet.



### 9. Solution de Claude Sonnet 4.6 : résolution incorrecte du problème de domination du cavalier

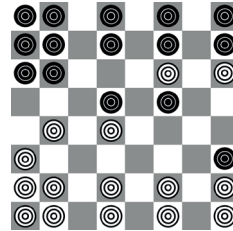
Le problème de domination du cavalier consiste à déterminer le nombre minimal de cavaliers nécessaires pour attaquer toutes les cases d'un échiquier de  $8 \times 8$  cases. La réponse est connue : 14. Dans cette solution proposée par Claude Sonnet 4.6, le modèle restitue correctement le résultat fourni par l'utilisateur dans le prompt, mais produit une configuration erronée ne comportant que 12 pièces. Cette situation illustre bien l'écart entre la connaissance mémorisée par les modèles et la construction effective d'une solution. Le modèle invite ensuite à vérifier case par case la validité de la configuration. Avec le temps, ces erreurs récurrentes sont devenues classiques, et certains modèles anticipent désormais les erreurs en mettant en garde l'utilisateur. À un autre niveau, les fonctions dites de raisonnement des LLM permettent parfois d'éviter certains problèmes de ce type, mais ces « raisonnements » restent des simulations fondées sur des probabilités et des motifs linguistiques, plutôt que sur des déductions logiques rigoureuses.



### 10, 11 & 12. Solutions de Qwen3-30B : répétition de solutions et résolution incorrecte du problème des 8 tours

Le problème des 8 tours consiste à placer huit tours sur un échiquier de  $8 \times 8$  cases de sorte qu'aucune ne puisse en attaquer une autre. Autrement dit, les tours ne doivent pas partager la même rangée ni la même colonne. Le nombre de solutions est bien connu et peut être obtenu par des méthodes algorithmiques classiques. Dans les réponses proposées par Qwen3-Coder-30B, le modèle identifie correctement le nombre total de solutions, mais échoue à les représenter correctement, en répétant

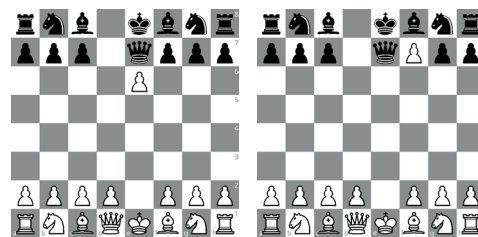
plusieurs fois une même configuration correcte. Dans l'échiquier n° 12, un modèle différent de la même famille propose une résolution erronée tout en la commentant avec assurance : « Cela paraît chaotique mais c'est parfaitement valide — aucune ligne ni colonne partagée. » On notera, dans ce commentaire, la présence du fameux tiret cadratin devenu le marqueur des textes générés par IA.



### 13. Solution de GLM-4.7-Flash : résolution incorrecte du problème de permutation des pions

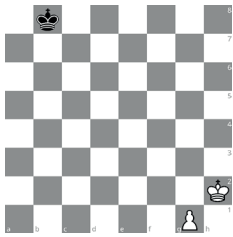
Le problème consiste à échanger deux groupes de pions de dames placés de part et d'autre de l'échiquier, en respectant les règles de déplacement. Les différentes étapes peuvent être lues de gauche à droite sur l'échiquier. Dans cette tentative de résolution par GLM-4.7-Flash, la solution est incorrecte dès les premiers coups. Malgré cela, le modèle poursuit la génération de coups comme si la solution restait cohérente.

Cet exemple illustre une difficulté récurrente : le modèle ne détecte pas ses propres erreurs en cours de résolution et continue à produire une séquence apparemment structurée mais globalement invalide. On voit bien ici l'essence des LLM : des machines à produire des « tokens ». Qu'il s'agisse de textes ou d'images, le modèle ne manipule jamais du sens, mais des fragments : tokens linguistiques ou visuels, assemblés selon des probabilités. Les mécanismes de raisonnement, l'appel à des outils ou les comportements agentiques ne remettent pas en cause leurs principes fondamentaux ni leurs limites, même s'ils permettent d'en augmenter les capacités. On peut se demander si en devenant plus efficaces, les LLM ne deviennent pas plus « bêtement intelligents ».



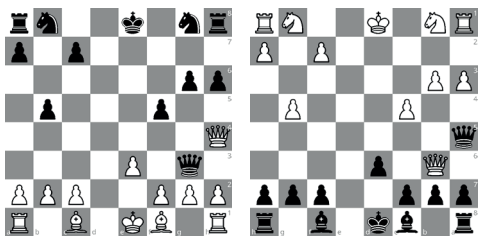
### 14 & 15. Solutions de ChatGPT 5 : génération d'un coup illégal après quatre coups

À partir d'une courte séquence de quatre coups d'une partie d'échecs, les modèles de langage récents peuvent générer un coup illégal, un comportement déjà observé avec des modèles comme GPT-3.5, GPT-4 ou o4-mini et que l'on retrouve ici dans les deux dernières étapes de jeu de ChatGPT 5. Cet exemple met en évidence une limite persistante de ces modèles à respecter strictement les règles d'un système formel, malgré leurs capacités de raisonnement avancées.



### 16. Solution de Stockfish : analyse erronée d'une position illégale

Le puissant moteur d'échecs Stockfish (qui fonctionne selon des techniques classiques d'intelligence artificielle) évalue cette position comme gagnante pour les Blancs. Le programme analyse ici uniquement la position donnée et, en fin de partie, un pion supplémentaire donnerait l'avantage aux Blancs mais la position des pièces est illégale. Selon les règles, le pion blanc ne peut pas être placé sur la première rangée.



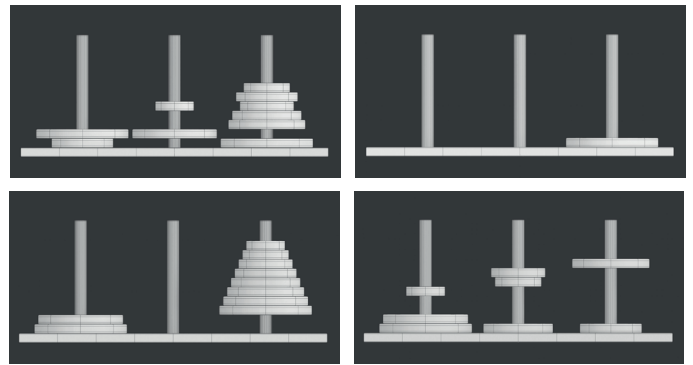
### 17 & 18. Solutions de Gemini-2.5-Flash : compréhension faussée après inversion des couleurs

Dans l'échiquier n° 17, la dame noire est attaquée par deux pions blancs donnant l'avantage aux Blancs. Ici le modèle Gemini-2.5-Flash analyse correctement la position dans sa forme initiale mais se trompe lorsqu'on garde les positions en inversant les couleurs des pièces dans l'échiquier n° 18. Le modèle ne reconnaît pas une situation de jeu identique et fait une erreur d'analyse. Cela montre la difficulté du modèle à généraliser indépendamment des motifs appris à l'entraînement.



### 19, 20 & 21. Solutions de ChatGPT o1-mini : génération répétée de coups illégaux

Une version de ChatGPT datant de septembre 2024 et nommée ChatGPT o1-mini produit ici des erreurs successives de position en générant des coups illégaux à répétition. À chaque fois, le modèle déplace une pièce sur une case déjà occupée.



### 22, 23, 24 et 25. Solutions de ChatGPT 5.2 et Gemini 3 : configurations invalides dans la résolution du problème des tours de Hanoï

Les tours de Hanoï sont un jeu de réflexion dans lequel il faut déplacer, en un minimum de coups, des anneaux de tailles différentes d'une tige de départ vers une tige d'arrivée, en passant si besoin par une tige intermédiaire. Un seul anneau peut être déplacé à la fois, et un anneau ne peut être placé que sur un anneau plus grand que lui ou sur une tige vide. Dans les étapes proposées par les modèles de langage, les configurations finales sont incorrectes et les états intermédiaires impossibles ou faux. Certaines configurations montrent des empilements « flottants », irréalisables dans une partie réelle. Les difficultés de raisonnement observées dans les jeux d'échecs se retrouvent dans d'autres systèmes à contraintes simples mais strictes. Les LLM ne possèdent pas non plus de véritable représentation physique du monde. Les solutions laissent pensif et ne sont pas sans rappeler les tableaux de Sigmar Polke dans lesquels  $1 + 1$  est égal à 3, autant de manières de réussir à échouer dans un monde sens dessus dessous.

### Références :

- <https://blog.mathieuacher.com/GPT5-IllegalChessBench/>
- Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A. F., Ippolito, D., Choquette-Choo, C. A., Wallace, E., Tramèr, F., & Lee, K. (2023). Scalable Extraction of Training Data from (Production) Language Models. <https://arxiv.org/abs/2311.1703>
- Shojaei, P., Mirzadeh, I., Alizadeh, K., Horton, M., Bengio, S., & Farajtabar, M. (2025). The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity. <https://arxiv.org/abs/2506.06941>
- Song, X., Wang, W., Cao, R., & Hu, Q. (2025). Beyond Accuracy: A Geometric Stability Analysis of Large Language Models in Chess Evaluation. <https://arxiv.org/abs/2512.15033>