Desafios da IA para o futuro

alucinações, “ilusão de pensamento” e caminhos de engenharia

Cesar de Oliveira F. Silva

Setembro / 2026

Roteiro

1. Contexto e motivação
2. Why Language Models Hallucinate
3. The Illusion of Thinking
4. How well do DeepSeek, ChatGPT, and Gemini respond to water science questions?
5. Engenharia aplicada e PINNs
6. Eficiência energética e linguagens
7. Conclusões

IA generativa hoje

- Modelos de linguagem de grande porte (LLMs) e Large Reasoning Models (LRMs) tornaram-se centrais em aplicações de IA.
- Ao mesmo tempo, problemas estruturais permanecem:
  - **Alucinações**: respostas plausíveis, porém factualmente incorretas.
  - **Limites de raciocínio**: colapso de desempenho em tarefas complexas.
  - **Custo computacional e energético** elevado.
- Pergunta: qual rumos da aplicação de **IA em projetos de engenharia ambiental** são mais promissores para um futuro mais confiável e sustentável?

Why Language Models Hallucinate

Adam Tauman Kalai* OpenAI
Ofir Nachum OpenAI
Santosh S. Vempala† Georgia Tech
Edwin Zhang OpenAI

September 4, 2025

Abstract

Like students facing hard exam questions, large language models sometimes guess when uncertain, producing plausible yet incorrect statements instead of admitting uncertainty. Such "hallucinations" persist even in state-of-the-art systems and undermine trust. We argue that language models hallucinate because the training and evaluation procedures reward guessing over acknowledging uncertainty, and we analyze the statistical causes of hallucinations in the modern training pipeline. Hallucinations need not be mysterious—they originate simply as errors in binary classification. If incorrect statements cannot be distinguished from facts, then hallucinations in pretrained language models will arise through natural statistical pressures. We then argue that hallucinations persist due to the way most evaluations are graded—language models are optimized to be good test-takers, and guessing when uncertain improves test performance. This "epidemic" of penalizing uncertain responses can only be addressed through a socio-technical mitigation: modifying the scoring of existing benchmarks that are misaligned but dominate leaderboards, rather than introducing additional hallucination evaluations. This change may steer the field toward more trustworthy AI systems.

Visão geral do artigo *Why Language Models Hallucinate*

- Kalai et al. (2025) analisam por que modelos de linguagem alucinam mesmo quando parecem sofisticados.
- Alucinação: geração de respostas plausíveis, mas erradas, muitas vezes com alta confiança.
- Ideia central: as alucinações surgem de **pressões estatísticas** no pré-treino e de **como avaliamos** os modelos.
- Os autores conectam geração de texto a um problema de classificação binária:
  - "Isto é uma saída válida ou um erro?"
  - Mostram que, mesmo com dados perfeitos, há **limites inferiores inevitáveis** para a taxa de erro.

Causas estatísticas das alucinações

- No pré-treino, o modelo aprende uma distribuição de linguagem a partir de um grande corpus.
- Fatos raros ou praticamente não repetidos levam a **incerteza epistemológica**:
  - Se um fato aparece uma única vez, o modelo não tem padrão forte para generalizar.
  - Em muitos casos, ele acaba “chutando” quando confrontado com perguntas sobre esses fatos.
- A análise dos autores mostra que parte das alucinações é **esperada estatisticamente**, e não apenas um “bug de engenharia” pontual.

Por que as alucinações persistem no pós-treino

- Pós-treino (RL, feedback humano, etc.) deveria reduzir alucinações, mas:
  - Benchmarks e leaderboards usam métricas 0–1 (acertou/errou) que **penalizam respostas do tipo "não sei"**.
  - Modelos que **sempre respondem**, mesmo inseguros, tendem a ter melhor score.
- Analogia com provas humanas:
  - Alunos que chutam em questões difíceis às vezes são recompensados.
  - Modelos de linguagem são otimizados para serem "bons de prova", não necessariamente honestos.
- Proposta dos autores:
  - Ajustar a forma de **pontuar avaliações principais**, de modo a não punir abstenções justificadas.
  - Encarar a mitigação de alucinações como um problema **socio-técnico**, não só algorítmico.

Implicações para o futuro da IA

- Mesmo com modelos maiores e mais dados, há **limites estatísticos** para a eliminação de alucinações.
- Sem mudar objetivos e avaliações, continuaremos a premiar modelos que:
  - soam confiantes, mas
  - não representam bem sua incerteza.
- Para aplicações críticas (saúde, direito, engenharia), precisamos:
  - Modelos que saibam dizer “não sei” ou encaminhar para ferramentas externas.
  - Arquiteturas que reduzam a exposição a alucinações em partes sensíveis do sistema.

The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity

Parshin Shojaee*† Iman Mirzadeh* Keivan Alizadeh
Maxwell Horton Samy Bengio Mehrdad Farajtabar

Apple

Abstract

Recent generations of frontier language models have introduced Large Reasoning Models (LRMs) that generate detailed thinking processes before providing answers. While these models demonstrate improved performance on reasoning benchmarks, their fundamental capabilities, scaling properties, and limitations remain insufficiently understood. Current evaluations primarily focus on established mathematical and coding benchmarks, emphasizing final answer accuracy. However, this evaluation paradigm often suffers from data contamination and does not provide insights into the reasoning traces' structure and quality. In this work, we systematically investigate these gaps with the help of controllable puzzle environments that allow precise manipulation of compositional complexity while maintaining consistent logical structures. This setup enables the analysis of not only final answers but also the internal reasoning traces, offering insights into how LRMs "think". Through extensive experimentation across diverse puzzles, we show that frontier LRMs face a complete accuracy collapse beyond certain complexities. Moreover, they exhibit a counterintuitive scaling limit: their reasoning effort increases with problem complexity up to a point, then declines despite having an adequate token budget. By comparing LRMs with their standard LLM counterparts under equivalent inference compute, we identify three performance regimes: (1) low-complexity tasks where standard models surprisingly outperform LRMs, (2) medium-complexity tasks where additional thinking in LRMs demonstrates advantage, and (3) high-complexity tasks where both models experience complete collapse. We found that LRMs have limitations in exact computation: they fail to use explicit algorithms and reason inconsistently across puzzles. We also investigate the reasoning traces in more depth, studying the patterns of explored solutions and analyzing the models' computational behavior, shedding light on their strengths, limitations, and ultimately raising crucial questions about their true reasoning capabilities.

The Illusion of Thinking: configuração experimental

- Shojaee et al. (2025) estudam **Large Reasoning Models (LRMs)**:
  - Modelos que geram longos traços de "pensamento" antes da resposta final.
- Em vez de só benchmarks de matemática/código, os autores usam:
  - Ambientes de **puzzles** com complexidade controlável (ex.: Torre de Hanói, River Crossing, Blocks World).
  - Capacidade de verificar tanto a **resposta final** quanto o **passo a passo do raciocínio**.
- Objetivo: entender como o desempenho e o "esforço de raciocínio" escalam com a complexidade do problema.

Três regimes de complexidade para LRMs

- Os autores identificam **três regimes** ao comparar LRMs com LLMs "normais" sob o mesmo custo de inferência:
  1. **Baixa complexidade**: LLMs padrão são mais eficientes; LRMs "pensam demais" sem ganho.
  2. **Complexidade média**: LRMs passam a ter vantagem; o raciocínio explícito ajuda.
  3. **Alta complexidade**: ambos colapsam para quase 0% de acerto.
- Ponto crítico:
  - Perto do colapso, LRMs **diminuem** o número de tokens de pensamento conforme a tarefa fica mais difícil, mesmo tendo orçamento de tokens disponível.

A “ilusão de pensamento”

- Analisando os traços internos de raciocínio, os autores observam:
  - **Overthinking**: em tarefas simples, o modelo encontra a solução correta cedo, mas continua gerando texto redundante.
  - **Fixação em erros**: em tarefas difíceis, o modelo fixa em uma estratégia errada cedo e não se corrige, desperdiciando tokens.
  - **Limitações algorítmicas**: mesmo quando recebe um algoritmo explícito, o modelo nem sempre o segue corretamente.
- Conclusão:
  - Mais “pensamento” textual não implica raciocínio algorítmico robusto.
  - Há **limites de escalabilidade** no raciocínio desses modelos, ligados à complexidade composicional das tarefas.

Desafios de raciocínio para o futuro da IA

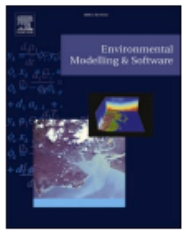
- Em combinação com o trabalho sobre alucinações, o quadro é:
  - Respostas superconfiantes, mesmo quando o modelo está estatisticamente condenado a errar.
  - Colapso de raciocínio em problemas com alta profundidade composicional.
- Isso limita o uso de LLMs/LRMs como **motores únicos** de raciocínio em domínios de engenharia complexos.
- Abre espaço para abordagens onde:
  - O LLM atua mais como **interface natural** e orquestrador,
  - enquanto o núcleo de cálculo e de garantias vem de modelos estruturados e de física/engenharia.

Environmental Modelling and Software 196 (2026) 106772

Contents lists available at ScienceDirect

Environmental Modelling and Software

journal homepage: www.elsevier.com/locate/envsoft



How well do DeepSeek, ChatGPT, and Gemini respond to water science questions?

Seyed Hossein Hosseini [a,*], Ali Pourzangbar [b]

[a] Department of Built Environment, School of Engineering, Aalto University, Espoo, Finland
[b] Institute for Water and River Basin Management, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

ARTICLE INFO

Keywords:
Large language model (LLM)
Hydrology
Water science
DeepSeek R1
ChatGPT-4o
Gemini 2

ABSTRACT

This study aims to evaluate the performance of three prominent LLMs, DeepSeek R1, ChatGPT-4o, and Gemini 2, in addressing key questions within four core fields of hydrology and water science: machine learning and optimization, remote sensing, flood modeling, and sediment transport. LLMs' responses are systematically compared to benchmark responses derived from review articles in the respective fields. To assess the LLMs' efficiency, a novel evaluation rubric is introduced in this study, incorporating four key criteria: relevancy, accuracy, authenticity, and novelty. Findings revealed that each model can address the core aspects of the benchmark questions. DeepSeek R1 achieved the highest overall scores in machine learning and optimization, flood modeling, and sediment transport, while ChatGPT-4o demonstrated superior performance in remote sensing. Notably, DeepSeek R1 and Gemini 2 exhibited the lowest response similarity in 95 % of the evaluated questions, whereas ChatGPT-4o and Gemini 2 showed the highest similarity in 70 % of cases.

Estudo de caso em hidrologia e ciência da água

- Estudo: **Hosseini & Pourzangbar (2026) – "How well do DeepSeek, ChatGPT, and Gemini respond to water science questions?"**
- Objetivo:
  - Avaliar **DeepSeek R1**, **ChatGPT-4o** e **Gemini 2** em quatro áreas de hidrologia:
    (i) machine learning e otimização; *(ii)* sensoriamento remoto; *(iii)* modelagem de cheias; *(iv)* transporte de sedimentos.
- Todos os modelos foram usados em suas **versões gratuitas** nas plataformas oficiais.
- Respostas dos LLMs foram comparadas com **respostas-referência** retiradas de artigos de revisão de alto impacto.

Rubrica de avaliação e desenho experimental

- Para cada um dos 4 tópicos:
  - Selecionam-se **artigos de revisão** representativos.
  - São criadas **5 questões conceituais** por tópico (enfatizando desafios, aplicações e soluções, não números específicos).
- Prompt em três partes:
  1. **Visão geral** (trecho de revisão, sem dar a resposta).
  2. **Prompt de papel**: “responder como especialista em hidrologia”.
  3. **Questão** propriamente dita.
- Rubrica com quatro critérios (Tabela 2–3, pág. 5):
  - **Relevância** (10%) – aderência ao tema.
  - **Acurácia** (35%) – presença dos elementos-chave da resposta-referência.
  - **Autenticidade** (35%) – correção científica.
  - **Novidade** (20%) – ideias novas além do benchmark.

Medição de acurácia, similaridade e velocidade

- **Acurácia**:
  - Algoritmo de **"exact and fuzzy word matching"** (Figura 3, pág. 6): tokeniza o texto, procura correspondências exatas e aproximações via distância de Levenshtein, com limiar definido por testes.
  - Acurácia = % de palavras-chave do benchmark presentes na resposta do LLM.
- **Similaridade entre modelos**:
  - Uso do `spaCy` para calcular similaridade semântica entre pares de respostas (heatmaps nas Figuras 4, 7, 10 e 13).
- **Velocidade de geração**:
  - Medem número de palavras geradas dividido pelo tempo de resposta (*word/s*); resultados nas Figuras 6, 9, 12 e 15.
- **Score global** para cada resposta:

$$\text{Score} = 0{,}10\,\text{Rel.} + 0{,}35\,\text{Acur.} + 0{,}35\,\text{Autent.} + 0{,}20\,\text{Novid.}$$

Resultados principais por área

- **Machine learning & otimização** (Tabela 4, Figura 5):
  - DeepSeek obteve a **maior soma de scores** (19,95) e acurácia média de $\sim 75{,}5\%$.
  - ChatGPT e Gemini vieram logo atrás ($\sim 73{,}6\%$ e $71{,}1\%$).
- **Sensoriamento remoto** (Tabela 5, Figura 8):
  - ChatGPT foi o melhor no score agregado (18,31) e na acurácia média ($\sim 62\%$).
- **Modelagem de cheias** (Tabela 6, Figuras 10–12):
  - DeepSeek liderou em score (19,60) e acurácia ($\sim 76{,}5\%$).
- **Transporte de sedimentos** (Tabela 7, Figuras 13–15):
  - DeepSeek teve maior **score global**; acurácia média similar entre os modelos, com leve vantagem para Gemini ($\sim 79{,}7\%$).

Similaridade entre modelos e velocidade de resposta

- **Similaridade das respostas** (heatmaps nas Figuras 4, 7, 10, 13):
  - Em $\approx$ 95% das questões, **DeepSeek** e **Gemini** foram o par com **menor similaridade** semântica.
  - Em $\approx$ 70% dos casos, **ChatGPT** e **Gemini** formaram o par **mais semelhante**.
- **Velocidade de geração** (Figuras 6, 9, 12, 15):
  - **Gemini** é claramente o mais rápido: $\sim$ 86 a 98 palavras/s, dependendo do tópico.
  - **ChatGPT** fica em faixa intermediária: $\sim$ 13 a 24 palavras/s.
  - **DeepSeek** é o mais lento ($\sim$ 6–9 palavras/s), em parte pelo foco em raciocínio detalhado.

Perfis dos modelos e implicações práticas

- **DeepSeek R1**
  - Respostas mais **técnicas**, estruturadas e matematicamente detalhadas, com alta **novidade** em vários tópicos.
  - Útil quando se busca **análises profundas** e há tempo de inferência disponível.
- **ChatGPT-4o**
  - Equilíbrio entre **profundidade técnica** e **estrutura clara**; desempenho destacado em **sensoriamento remoto**.
  - Bom para integração **interdisciplinar** e explicações bem organizadas.
- **Gemini 2**
  - Foco em **aplicação prática** em campo, sustentabilidade e stakeholders.
  - **Mais rápido** dos três — interessante para fluxos de trabalho em que a velocidade é crítica.

Lições para IA em engenharia hídrica

- Todos os modelos conseguem capturar os **pontos centrais** das respostas de referência em ciência da água, mas:
  - Há variação em acurácia, novidade e estilo de resposta.
  - Em tópicos não bem cobertos no treinamento, a confiabilidade ainda é uma preocupação.
- O estudo reforça que:
  - LLMs são ferramentas úteis para **síntese, brainstorming e apoio** à decisão em hidrologia.
  - Ainda é necessária **validação por especialistas** e integração com modelos físicos (hidrológicos, hidráulicos, PINNs etc.).
- Em linha com os desafios discutidos antes na apresentação:
  - O futuro da IA em água passa por combinar LLMs com modelos **fisicamente fundamentados** e algoritmos eficientes em C/CUDA, mantendo o humano no circuito.

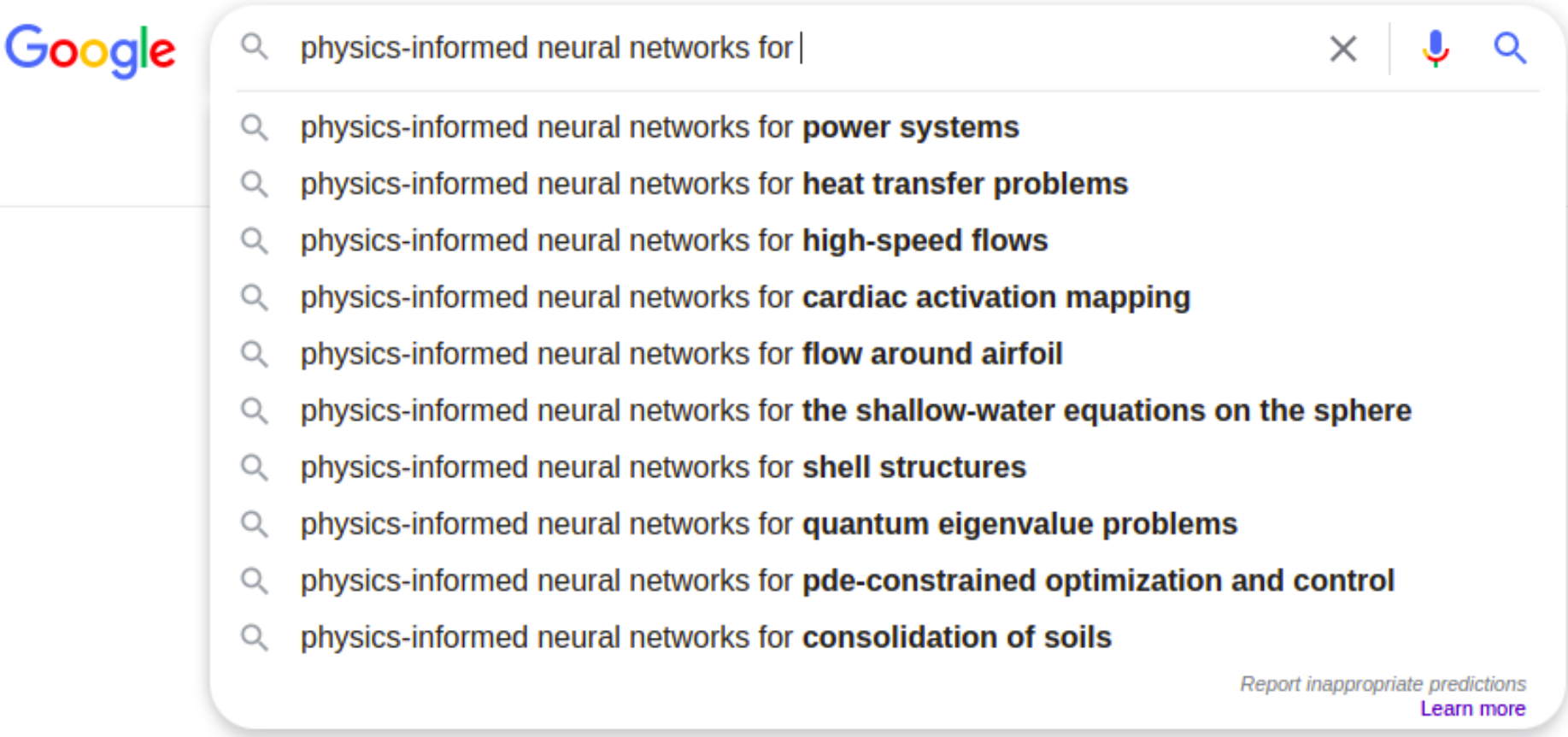
Por que apostar em engenharia aplicada

- Estes três artigos sugerem limitações **estruturais** dos LLMs/LRMs:
  - Alucinações alimentadas pelo próprio processo de treino e avaliação.
  - Dificuldade em manter raciocínios consistentes em alta complexidade.
- Em contraste, muitos problemas de engenharia:
  - são regidos por leis físicas bem conhecidas (equações diferenciais, conservação de massa/energia, etc.);
  - exigem **confiabilidade** e **interpretabilidade** maiores que um chatbot genérico.
- Estratégia: investir em modelos que **integram conhecimento de domínio** desde a formulação, como as **Physics-Informed Neural Networks (PINNs)**.

Physics-Informed Neural Networks (PINNs)

- PINNs são redes neurais que incorporam **leis físicas** (tipicamente equações diferenciais parciais):
  - A função de perda inclui termos que penalizam violações das equações e das condições de contorno.
- Vantagens para engenharia:
  - Podem aprender com **poucos dados**, usando as leis físicas como regularização.
  - Produzem soluções que respeitam **restrições de conservação** e outras propriedades físicas.
  - São úteis como **substitutos numéricos** (surrogate models) para simulações caras.
- Exemplos de aplicações:
  - Mecânica de fluidos, transferência de calor, elasticidade, problemas multi-físicos.
  - Controle preditivo, modelagem de materiais, problemas inversos em engenharia.
- Ideia-chave: ao restringir o espaço de soluções às fisicamente plausíveis, reduzimos o "espaço de alucinação" do modelo.

Aplicações das Redes Neurais Informadas pela Física (PINNs)



Linguagens de programação e energia

- Estudos experimentais com dezenas de linguagens mostram que:
  - **Linguagens compiladas** (C, C++, Rust, Fortran) tendem a ser as mais rápidas e energeticamente eficientes.
  - **Linguagens interpretadas** (Python, Perl, Ruby, Lua) estão entre as piores em energia e tempo.
- Em média, soluções em C consomem **ordens de grandeza menos energia** que equivalentes em Python para os mesmos algoritmos.
- Razões principais:
  - Código C é compilado para instruções nativas altamente otimizadas.
  - Menos overhead em tempo de execução (sem interpretador pesado a cada operação).
  - Tipos e estruturas resolvidos em tempo de compilação, com melhor uso de cache e menos objetos dinâmicos.

Por que Python é bem mais custoso que C

- **Interpretação e tipagem dinâmica**:
  - Cada operação envolve despacho dinâmico, criação/manipulação de objetos de alto nível.
- **Coletor de lixo** e gestão automática de memória:
  - Facilita o desenvolvimento, mas adiciona trabalho extra em tempo de execução.
- **Loops numéricos puros em Python**:
  - Podem ser dezenas ou centenas de vezes mais lentos que equivalentes em C, com consumo proporcionalmente maior de energia.
- Por isso, em computação científica:
  - Usamos Python como **camada de orquestração**,
  - enquanto o núcleo numérico fica em bibliotecas escritas em C, C++ ou Fortran (NumPy, BLAS, cuDNN, etc.).

GPUs, Python e a base em C

- Frameworks modernos de redes neurais:
  - **PyTorch**: API em Python, backend em C++ e CUDA (biblioteca `libtorch`).
  - **TensorFlow**: APIs em Python, C++ e C, com kernels de alto desempenho em C++/CUDA.
- O que acontece quando escrevemos em Python:
  - A chamada Python é um **invólucro leve** que agenda operações em tensores.
  - O cálculo pesado é feito por kernels otimizados em C/C++ e CUDA na CPU ou GPU.
- Consequências:
  - A “fachada” da IA é Python, mas o consumo real de tempo e energia está em código C/C++ compilado.
  - Para máxima eficiência (latência, energia, custo), queremos empurrar o máximo de lógica possível para essa camada nativa.

Conclusões

- **Alucinações**:
  - Decorrem de limitações estatísticas do treinamento e de avaliações que punem a incerteza.
- **Ilusão de raciocínio**:
  - LRMs mostram ganhos em alguns regimes, mas colapsam em alta complexidade e nem sempre seguem algoritmos de forma consistente.
- **Engenharia aplicada**:
  - Modelos como PINNs aproveitam leis físicas e conhecimento de domínio para aumentar confiabilidade e eficiência de dados.
- **Eficiência computacional**:
  - C e outras linguagens compiladas são muito mais eficientes energeticamente que Python.
  - A aceleração em GPU de redes neurais é, na prática, baseada em C/C++ e CUDA, com Python como camada de controle.
- Mensagem final:
  - O futuro da IA passa por combinar LLMs como interfaces flexíveis com **núcleos de engenharia** robustos, fisicamente informados e energeticamente eficientes.