Uma taxonomia para modelos de raciocínio de última geração

Onde estivemos e para onde estamos indo com RLVR*
(Raciocínio Ampliado por Recuperação, Linguagem e Verificação)

Nathan Lambert

A primeira geração de modelos de raciocínio nos trouxe escalabilidade no tempo de inferência e o fascínio de ver o que pode ser chamado de processo de raciocínio de um modelo de linguagem.

A segunda geração de modelos de raciocínio nos trará novos tipos de aplicações de modelagem de linguagem agêntica.

As características e habilidades necessárias para modelos agênticos são adicionais à primeira geração, mas não estão presentes por padrão. Algumas das novas habilidades necessárias podem ser inicializadas com prompts inteligentes, mas, para obter os melhores resultados, precisamos treinar nossos modelos de raciocínio diretamente para otimizar o planejamento.

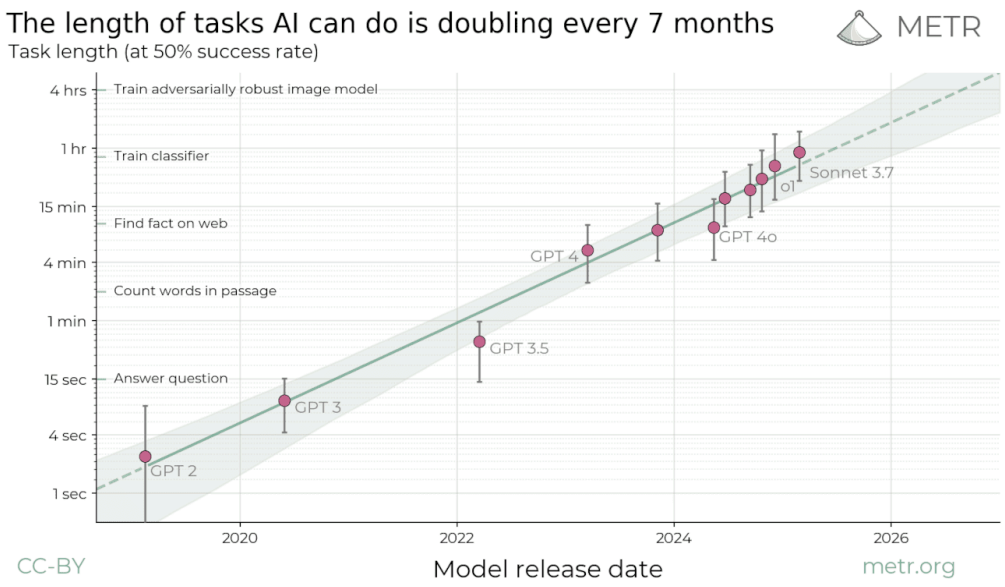
Nesta postagem, explicamos quatro aspectos principais dos modelos de raciocínio atuais e da próxima geração:

1. **Competências:** Capacidade de resolver problemas independentes.
2. **Calibração:** Capacidade de compreender a dificuldade de um problema e não pensar demais.
3. **Estratégia:** Capacidade de escolher o plano de alto nível certo.
4. **Abstração:** Capacidade de dividir uma estratégia em partes solucionáveis.

Elas são apresentadas na ordem em que devem ser resolvidas para formar um modelo de raciocínio progressivamente mais completo para tarefas complexas.

Habilidades, calibração, estratégia e abstração. As duas primeiras são habilidades nativas dos modelos em passagens de inferência únicas quando apresentados a um problema técnico, e as últimas são habilidades necessárias para construir agentes eficazes.

Para contextualizar, lembre-se do popular gráfico de "progressão do horizonte temporal" do METR:



Os modelos estavam ficando saturados em torno do GPT 4o em 2024. O desbloqueio das habilidades de raciocínio proporcionou um salto com o Claude Sonnet 3.7 em 2025. Um bom planejamento será a característica dos modelos que darão o salto de 1 para 4+ horas em 2026 e nos anos seguintes. Isso não é algo que se consegue de graça — os pesquisadores de IA precisam trabalhar muito para que isso aconteça.

Todo o entusiasmo em torno dos modelos de raciocínio explodiu quando foi demonstrado que o aprendizado por reforço escalonado com recompensas verificáveis (RLVR) permite que o modelo aprenda **habilidades** úteis para resolver uma variedade de tarefas downstream. A primeira confirmação pública disso foi com o DeepSeek R1, que mostrou como o tempo de treinamento RL compute se traduz em desempenho.

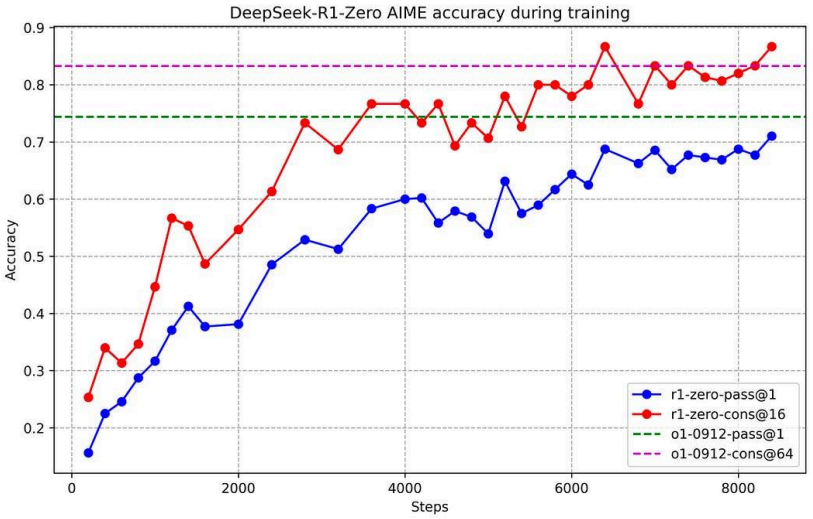


Figure 2 | AIME accuracy of DeepSeek-R1-Zero during training. For each question, we sample 16 responses and calculate the overall average accuracy to ensure a stable evaluation.

Interligado a isso, os modelos gerarão mais tokens por resposta enquanto descobrem essas habilidades. Em todos os modelos de raciocínio atuais, as habilidades listadas acima — habilidades, calibração, estratégia e abstração — podem ser ajustadas ainda mais com o aumento do gasto de tokens por componente.

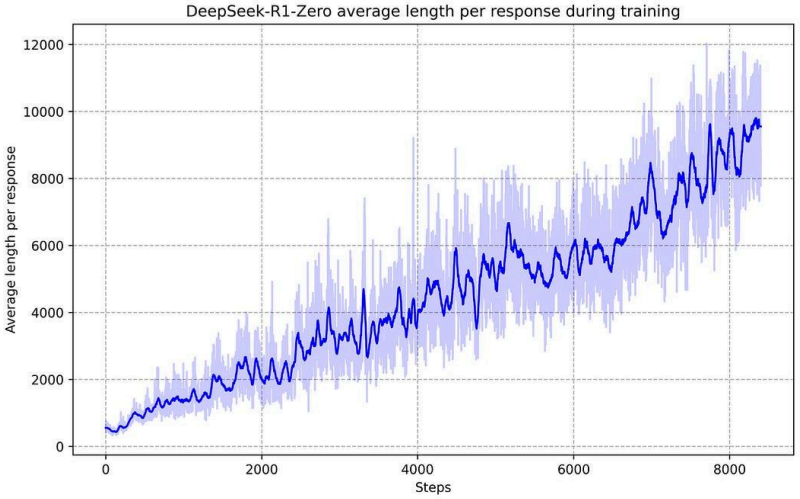


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

Este ano, todos os principais laboratórios de IA lançaram ou lançarão um modelo de raciocínio, pois esses modelos são melhores na aquisição de habilidades que

lhes permitem resolver os problemas mais difíceis na fronteira da IA — avaliações como Humanity's Last Exam, MATH, AIME, LiveCodeBench, Aider Polyglot, etc. observaram mudanças significativas no desempenho em relação à classe anterior de modelos. Essas habilidades são a base para todas as mudanças que estão ocorrendo no setor. Grande parte das discussões atuais sobre o dimensionamento do treinamento gira em torno de encontrar os problemas certos para permitir que os modelos se tornem mais robustos em uma variedade de cenários.

A corrida louca pela aquisição de habilidades nesses modelos aumentou um problema de segunda ordem dos modelos, que é pensar demais mesmo em problemas fáceis. Isso surge devido ao profundo acoplamento do treinamento RL e ao desbloqueio do dimensionamento do tempo de inferência. O objetivo final é claramente que os modelos dimensionem o tempo de computação da inferência por conta própria, proporcionalmente à dificuldade do problema. No curto prazo, quando a taxa de ganho de desempenho é tão alta, faz sentido priorizar as habilidades em detrimento da eficiência. À medida que as habilidades se saturarem, o desempenho e o custo serão ponderados de forma mais equilibrada.

No momento, a **calibração** da dificuldade do problema é transferida para o usuário na forma de seletores de modelo entre raciocinadores ou modelos de instrução tradicionais, botões para ativar/desativar o raciocínio, imposição de orçamento de raciocínio (1) e, em breve, seletores de esforço de raciocínio. No lado da pesquisa, foi demonstrado que as funções de perda RL são flexíveis o suficiente para permitir o controle mais preciso da duração — algo que funções de perda como ajuste de instrução ou preferência não conseguem fazer. Da mesma forma, os modelos treinados como raciocinadores expressam melhor sua confiança, o que em breve deve se traduzir em mitigações do excesso de raciocínio.

Calibrar a dificuldade do problema com o esforço da solução permitirá soluções muito mais práticas (e mais rápidas e agradáveis) para os usuários finais e

também soluções mais lucrativas. A calibração, embora seja uma característica de nível inferior dos modelos, não é um caminho tão crucial para a implementação de novos casos de uso com os modelos. Para isso, os criadores de IA vão recorrer a melhores capacidades de planejamento.

Para saber mais sobre as pesquisas atuais sobre calibração, clique na nota de rodapé a seguir (2).

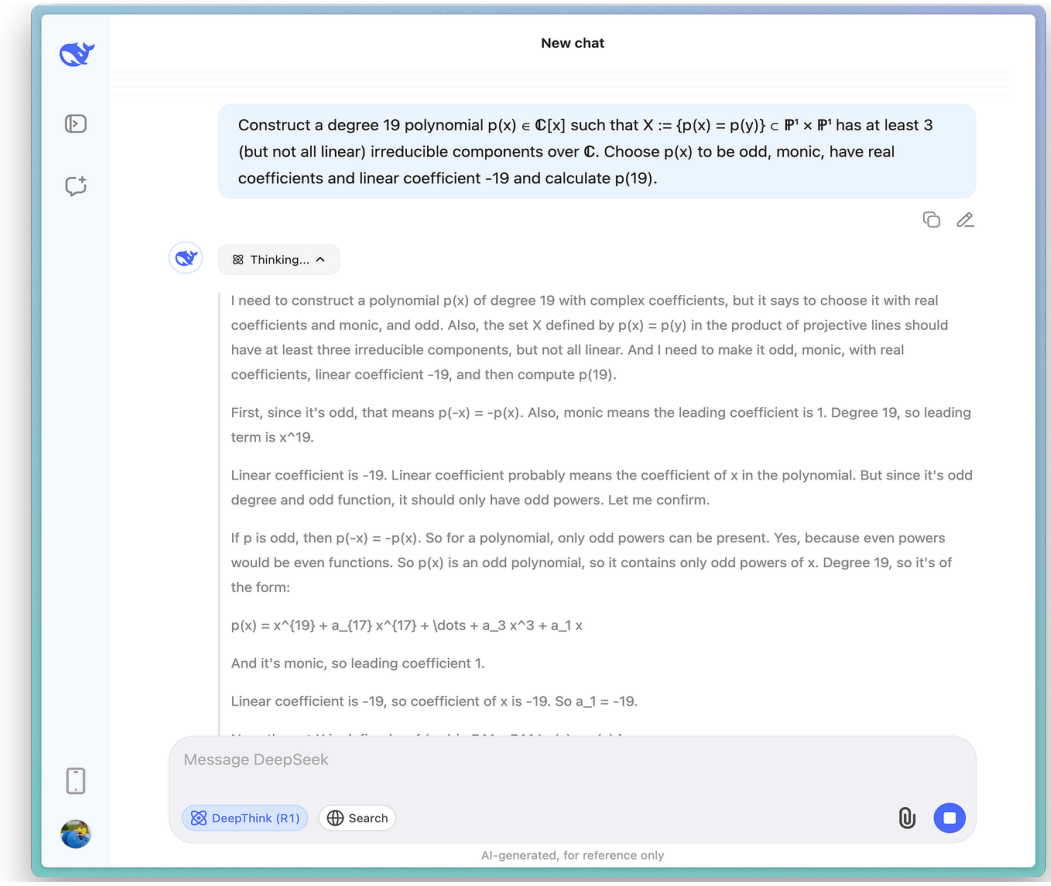
Antes de passarmos às capacidades de planejamento, que são frequentemente discutidas em detalhes na comunidade como sendo cruciais, sem fornecer uma maneira clara de compreendê-las, precisamos contextualizar **como a computação paralela e outros métodos de escalonamento do tempo de inferência afetarão o futuro dos modelos de raciocínio**. O método mais proeminente aqui é algum tipo de pesquisa misturada com modelos de consistência ou pontuação interna (por exemplo, modelos de recompensa) como o o1-pro. Por exemplo, na publicação sobre o lançamento do Claude 4, a Anthropic mencionou que utiliza "computação paralela em tempo de teste, através da amostragem de múltiplas sequências e da seleção da melhor através de um modelo de pontuação interno". O Google também anunciou, mas não lançou o Gemini Deep Think, que espelhará isso.

O uso desses métodos deixa claro que a computação paralela está fazendo algo muito diferente do que escalar o RL subjacente — é uma forma adicional de robustez ou qualidade nas respostas. O o1 pro, em meus testes, sempre foi o modelo mais consistente que experimentei. O escalonamento da computação aqui não ajuda diretamente o modelo a desbloquear mais habilidades, como a computação RL do tempo de treinamento, mas, na prática, parece semelhante porque uma melhor extração e formatação das respostas ajudam o modelo a parecer mais inteligente. A melhor maneira de resumir a direção um tanto ortogonal da computação paralela para escalonamento no tempo de inferência é que a qualidade geralmente é anti correlacionada com tokens raros quando uma métrica de classificação ou modelo de recompensa é implantado, pois tokens

raros serão suprimidos por métodos de votação majoritária ou modelos de recompensa que nunca os viram antes.

Quando se trata dos principais modelos de raciocínio do futuro, chamar a computação paralela ou apenas o pensamento linear estendido pode ser melhor considerado como uma ferramenta que o agente pode chamar. Eles serão flechas no arco de um modelo que planeja uma estratégia e sabe quais partes dela serão mais difíceis de superar.

No entanto, para chegar lá, os modelos precisam ser tratados de maneira muito diferente. Os modelos atuais fazem muito pouco planejamento em problemas difíceis, a menos que sejam solicitados a fazê-lo. Por exemplo, eis o que acontece quando o novo modelo R1 recebe um problema do Frontier Math (um dos benchmarks mais difíceis atualmente):



Com os modelos atuais, é razoável que eles façam um planejamento muito leve ou implícito — as habilidades que estamos tentando treinar permitirão que o modelo divida os problemas em etapas e os resolva. Implicitamente, os primeiros tokens que esses modelos recebem os levam a um determinado plano. Esses comportamentos serão menores em relação ao que surge em fluxos de trabalho agênicos — onde um plano é necessário a priori para restringir substancialmente o espaço de pesquisa.

Planejamento é o termo técnico usado para abranger as habilidades de longo prazo e de várias etapas dos modelos.

O planejamento abrange muitas sub habilidades e habilidades, mas a divisão de nível mais alto que importa na fronteira atual dos modelos de agentes é estratégia e abstração. **Estratégia** é a capacidade do modelo de se direcionar corretamente na direção de uma solução de alta qualidade. Com uma passagem auto-regressiva, apontar o fluxo de tokens na direção errada muitas vezes não é recuperável. Embora os agentes sejam um pouco melhores nisso por serem capazes de editar seu plano, eles ainda são altamente suscetíveis.

Abstração é como o modelo divide a estratégia em partes acessíveis. Mesmo com o modelo mais habilidoso, assumir uma sub tarefa muito difícil de uma só vez fará com que nenhum progresso seja feito no geral. Assumir poucas tarefas de cada vez fará com que o modelo expire. Atualmente, a abstração é um problema menor, pois o horizonte temporal é bastante curto, mas os modelos precisarão ser capazes de dividir tarefas de vários dias em subproblemas que possam ser resolvidos em etapas de inferência individuais de 1 a 2 minutos (ou seja, 10 a 100 mil tokens de inferência direta).

Uma habilidade intimamente relacionada é o gerenciamento de contexto, em que os modelos devem ser capazes de armazenar um resumo completo do que fizeram até o momento. As melhores formas de gerenciamento de contexto permitirão que o modelo pule tarefas nas quais acidentalmente voltou, mesmo que já tenham sido concluídas, ou tente uma nova estratégia após uma

abordagem mal sucedida. Essa é uma das muitas habilidades de baixo nível que surgirão para permitir habilidades de planejamento generalizadas (3).

O o3 é o modelo líder neste paradigma atualmente, com o maior espectro de habilidades em matemática, código e pesquisa, além de algumas habilidades de planejamento de ponta, como Deep Research. Quando o o3 encontra informações específicas para mim, atribuo muito pouco desse comportamento ao planejamento, mas sim à habilidade, ao uso de ferramentas de tentativas múltiplas (4), de saber continuar procurando até encontrar a resposta. Outros modelos têm qualidades que estão à frente em algumas regiões da fronteira de Pareto, como o planejamento do Claude 4 para tarefas de software (em essência, dizendo que o Claude Code é atualmente melhor do que o agente de codificação Codex da OpenAI).

o3 é melhor quando tem a tarefa de encontrar informações extremamente específicas que existem talvez em uma página da web. Ele falha quando solicitado a comparar todo o conteúdo que existe. Na taxonomia acima, o o3 quase resolveu a habilidade de pesquisa, mas a síntese em uma categoria ampla envolve um planejamento mais avançado das informações a serem obtidas e analisadas.

O planejamento não parece uma habilidade que eu esperaria que surgisse ao treinar em tarefas desafiadoras e com várias etapas, mas não ficaria surpreso se fosse um comportamento que pudesse ser refinado. Assim como a história do Q* foi, na verdade, um esforço inicial substancial de curadoria de dados pela OpenAI para criar alguns traços de raciocínio, eles provavelmente precisarão fazer o mesmo para semear comportamentos de planejamento de maior qualidade antes de continuar a treinar o modelo. As amostras de treinamento de alta qualidade aqui incluem estratégias de alto nível e detalhes sobre como abstrair o problema.

Assim como as habilidades específicas para raciocinar sobre problemas matemáticos ou de código únicos, como verificação ou checagem de trabalho, levará muito tempo até sabermos o equilíbrio entre o que surge do

pré-treinamento geral, do treinamento intermediário focado ou dos dados especializados de partida a frio.Independentemente do equilíbrio a longo prazo, em breve veremos uma corrida para adicionar essas capacidades de planejamento, de modo que os laboratórios possam começar com o pós-treinamento (dados SFT de partida a frio) que revelam tudo o que havia no pré-treinamento. Essa tarefa não será tão difícil quanto inicializar as cadeias de raciocínio em si, pois o planejamento tem mais a ver com resultados do que com o comportamento que os obtém (que deve ser parcialmente transferido de problemas matemáticos e de código difíceis).

A primeira coisa que os agentes atuais provavelmente fazem é escrever um plano de ataque para seu objetivo final. A fraqueza das habilidades atuais de planejamento é vista pela variação nos resultados, como Deep Research e Codex, onde oscila entre uma obra-prima e um fracasso. As capacidades de planejamento do Claude Code poderiam ser melhores por uma razão tão simples quanto o modelo ser ensinado a editar e revisitar o plano várias vezes enquanto está em execução. Esse tipo de escopo de distribuição de resultados, ou tempo que o modelo tentará, começa a vincular as capacidades de planejamento à calibração também.

Tudo isso traça um caminho bastante claro dos problemas que serão resolvidos nos próximos meses. As tarefas agênticas exigem mais daquilo que torna os modelos de raciocínio excelentes. Ao mesmo tempo, as tarefas são muito mais focadas em tarefas do mundo real do que em coisas representadas em benchmarks acadêmicos existentes. Os trabalhos acadêmicos atuais estão impulsionando fortemente a direção das habilidades para esses modelos, particularmente em matemática, e bastante em calibração (veja as notas de rodapé abaixo), mas não o suficiente nos subconjuntos de planejamento de que precisamos. O desafio é que essas capacidades só podem ser avaliadas no sistema mais amplo em que operam, o que muitas vezes será acompanhado por custos de inferência substanciais. A verdadeira corrida é para construir sistemas que as pessoas usem, seja com modelos abertos ou fechados, em vez de levar os

modelos ainda mais longe em habilidades que não mostram um valor claro, como problemas matemáticos quase impossíveis ou os escalões mais altos da programação competitiva.

Com os modelos atuais, devemos estar otimistas de que podemos resolver muitos dos problemas que surgirão. Temos algum trabalho manual de anotação de dados a fazer para iniciar as capacidades de planejamento e, em seguida, podemos tentar o objetivo final de treinar agentes de ponta a ponta com aprendizado por reforço em tarefas esparsas e de longo prazo.

Thanks to Ross Taylor for some feedback on an early form of this taxonomy and Sophie Alpert for helping crystallize some of my ideas around o3.

1 I.e. suppressing the end of thinking token </think> from generation and adding “wait” to get the model to think longer.

2 Calibration reading list.

Confidence & Calibration

- Language Models Prefer What They Know: Relative Confidence Estimation
- Uncertainty Quantification and Confidence Calibration in Large Language Models
- The Role of Calibration in Self-Improving Large Language Models
- Reasoning Models Better Express Their Confidence
- Uncertainty-Aware Decoding with Minimum Bayes Risk

Difficulty-Adaptive Reasoning

- When More is Less: Understanding Chain-of-Thought Length in LLMs
- DAST: Difficulty-Aware Self-Training on Large Language Models
- DAST-v2: Difficulty-Adaptive Slow-Thinking for Large Reasoning Models
- Thought Calibration: Efficient and Confident Test-Time Scaling

- GRPO-LEAD: A Difficulty-Aware Reinforcement Learning Approach for Mathematical Reasoning

Overthinking Mitigation

- THOUGHTTERMINATOR: Benchmarking, Calibrating and Mitigating Overthinking in LLMs
- Revisiting Overthinking in Long Chain-of-Thought from Self-Doubt

Planning & Abstraction

- Learning Adaptive Parallel Reasoning with Language Models
- Adaptive Deep Reasoning: Triggering Deep Thinking When Needed
- Adaptive Graph of Thoughts: Test-Time Adaptive Reasoning Unifying Chain, Tree and Graph

Source: https://chatgpt.com/share/683b8dcb-0d40-8005-ba05-bbd6c252d59e

3 E.g. Claude Code's context compression.

4 And lots of "effort" in tokens spent.

*RLVR significa **Retrieval-Augmented, Language-augmented, and Verifier-augmented Reasoning** — é uma proposta recente para categorizar e evoluir os modelos de raciocínio de IA de próxima geração, especialmente no contexto da pesquisa e desenvolvimento de agentes autônomos que combinam linguagem com ações e verificação.

A sigla **RLVR** representa um tipo de framework em camadas para pensar como os modelos de linguagem podem ser complementados com três capacidades:

1. **R (Retrieval-augmented)** Recuperação de informação – o modelo consulta bases de dados externas, documentos, código, ou conhecimento factual, em vez de depender apenas de sua memória treinada. Ex: RAG

(Retrieval-Augmented Generation).

2. **L (Language-augmented)** Linguagem como ferramenta de raciocínio – o raciocínio é mediado pela linguagem, permitindo planejar, refletir ou auto-explicar em etapas verbais. Ex: Chain of Thought, Tree of Thought, Reflexion.

3. **V (Verifier-augmented)** Verificação de respostas – há uma camada de verificação que checa se as respostas são válidas, consistentes ou corretas com base em critérios formais ou externos. Ex: o uso de verificadores simbólicos, motores de código, ou modelos auxiliares para detectar erros.

4. **R (Reasoning)** Raciocínio – o objetivo final do sistema é raciocinar com eficácia, seja para resolver problemas, tomar decisões ou executar planos complexos.

Esse acrônimo foi proposto em artigos e apresentações recentes da OpenAI e de pesquisadores ligados à DeepMind e Anthropic, como parte da tentativa de desenhar agentes de IA que **não apenas respondam, mas que saibam pensar, buscar, refletir e conferir**. (o3 GPT)