

画期的な発明を創造するのは、ChatGPT-5.2 Pro と Gemini3 Deep Think で比較するとどちらと考えられるか？

ChatGPT 5.2 Pro

「画期的な発明（＝新規性が高く、なおかつ実装・検証まで辿り着く発明）」という観点だと、どちらが“上”かを一発で断言するよりも、発明プロセスのどこを AI に担わせたいかで勝ち筋が変わります。結論だけ先に言うと：

- 発想の探索（多様な仮説を並列に出す／戦略的に枝分かれ探索する）に寄せるなら：Gemini 3 Deep Think がハマりやすい
→ Deep Think は「複数仮説を同時に探索する並列推論」や「創造性・戦略・段階的改善」に強いと説明されています。 (blog.google)
- 実装・検証まで含めた“発明を形にする”一気通貫（設計書、試験計画、コード、データ解析、長い仕様を抱える）に寄せるなら：ChatGPT-5.2 Pro がハマりやすい
→ GPT-5.2 は「長文脈」「エージェント的ツール実行」「視覚」「複雑タスクの end-to-end 遂行」を強化したとされています。 ([OpenAI](https://openai.com))

そして、**“画期的”の確率を最大化するなら、併用（相互査読）**がいちばん堅いです。理由は、発明はだいたい

①発散（新規アイデア） → ②収束（実現可能性・検証） → ③実装（プロトタイプ） → ④再発散（改良）

のループで進み、モデルごとに得意な局面が違うからです。

Gemini 3 Deep Think が「発明の発想」に効きやすいポイント

- Google は Deep Think を、Gemini 3 の推論・マルチモーダル理解を押し上げる**“強化推論モード”**として位置づけ、「より複雑な問題」を解くためのモードだと説明しています。 (blog.google)
- 仕組みとしても「複数仮説を同時に探索する並列推論」を明言しています。 (blog.google)
- さらに Google は、Deep Think が Humanity’s Last Exam / GPQA Diamond / ARC-AGI-2 などの難関ベンチで高い成績を出すと述べています（※条件差はありうるので“参考指標”として）。 (blog.google)
- 一方で、Deep Think は実験的機能で中断・停止されうるとも案内されています。 ([Google Help](https://google.com))

発明で言うと：

「問題設定をいくつかの見立てに分解して、並列に“あり得る機構”を出す」「戦略的に改良案を積む」みたいな局面で効きやすいです。

ChatGPT-5.2 Pro が「発明を形にする」に効きやすいポイント

- GPT-5.2 は、**長文脈理解・エージェント的ツール呼び出し・視覚**などを強化し、「複雑な現実タスクを end-to-end で実行する」方向を明確に打ち出しています。
([OpenAI](#))
- GPT-5.2 Pro は「難しい問題向け」で、推論強度 (reasoning effort) を上げられ、**400,000 のコンテキスト**などの仕様が公開されています。([OpenAI Platform](#))
- 科学・数学の文脈では、GPT-5.2 Pro が**統計学習理論の未解決問題に関するケーススタディ** (人間が検証しながら進めた) を OpenAI 自身が紹介しています。ここは「新規な発明 (研究的発明) に寄与しうる」根拠として強いです。([OpenAI](#))
- ただし OpenAI も「モデルは独立した研究者ではない」「検証と専門家判断が重要」と明確に釘を刺しています。([OpenAI](#))

発明で言うと：

「要求仕様が長い」「制約 (コスト・安全・規格) が多い」「検証設計・実験計画・実装・文書化まで落とす」局面で効きやすいです。

じゃあ結局どっち？ (意思決定の目安)

あなたが“発明で詰まりがち”なのがどこかで選ぶのが現実的です。

Gemini 3 Deep Think を優先しやすい人

- そもそも「何を作るべきか」の着想が枯れがち
- 1つの案に早く収束しすぎて、別ルート探索が足りない
- 機構・戦略・改良案を、並列に比較検討したい

ChatGPT-5.2 Pro を優先しやすい人

- アイデアはあるが、設計・検証・実装に落ちない
- 仕様書・論文・ログ・実験ノートなど、長い材料を抱えて整合させたい
- プロトタイプ (コード、計算、手順書、試験計画、ドキュメント) までを一気通貫で作りたい

“画期的”の確率を上げる最適解

- 併用 (発散 = Gemini、収束 = GPT、相互批判 = 両方)
→ 片方が作った案をもう片方に「壊してもら (欠陥・前提・既存技術・安全性・コスト)」のが効きます。
-

すぐ試せる「併用ワークフロー」テンプレ

1. **Gemini 3 Deep Think** に投げる (発散)
 - 「この課題に対して、原理が異なる解決策を 10 個。各案は“作用機構”を 1 行で」
 - 「既存の前提を 3 つ疑って、前提を変えた世界線の解法を 5 つ」
2. **ChatGPT-5.2 Pro** で絞る (収束)
 - 「上位 3 案を、性能・コスト・リスク・検証容易性でスコアリングして、最短で PoC

できる順に並べて」

- 「最有望案について、実験計画（測るべき指標、失敗条件、必要機材、想定ノイズ）を書いて」
 - 「プロトタイプの擬似コード／回路ブロック図／BOM 案／安全対策／規格リスクを列挙して」
3. **Gemini** に“敵役レビュー”させる
 - 「この案が現実には失敗する理由を 10 個（物理・製造・人間工学・法規・市場）」
 4. **ChatGPT** で統合改訂
 - 「致命傷トップ 3 を潰す設計変更を提案し、再度テスト計画を更新」

-
- [Reuters](#)
 - [Android Central](#)
 - [axios.com](#)