

72.9%

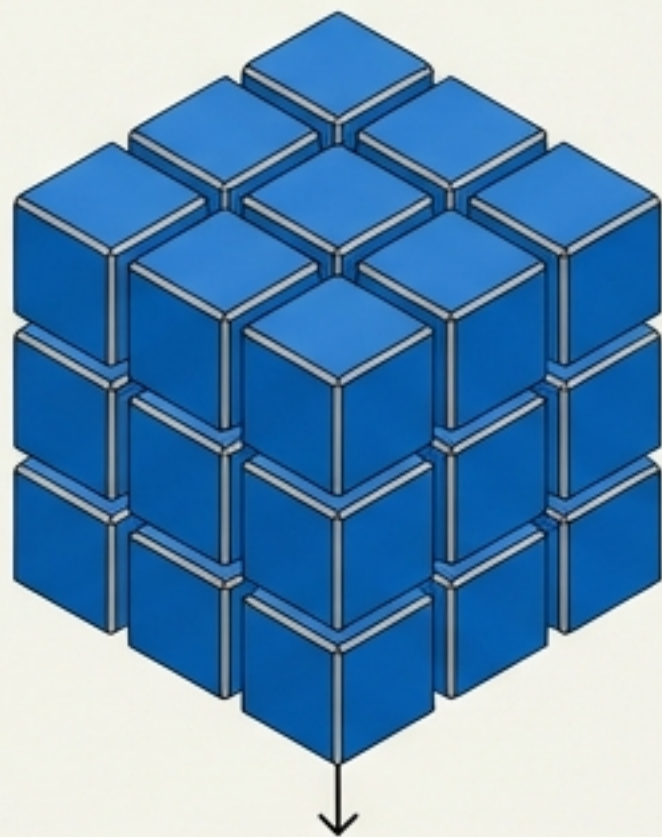
汎用人工知能への飛躍：ARC-AGI-2における「人間超え」の衝撃

Multi-Model Reflective Reasoningと 推論時計算が切り拓く新時代

REPORT: JOHAN LAND BREAKTHROUGH
DATE: 2026

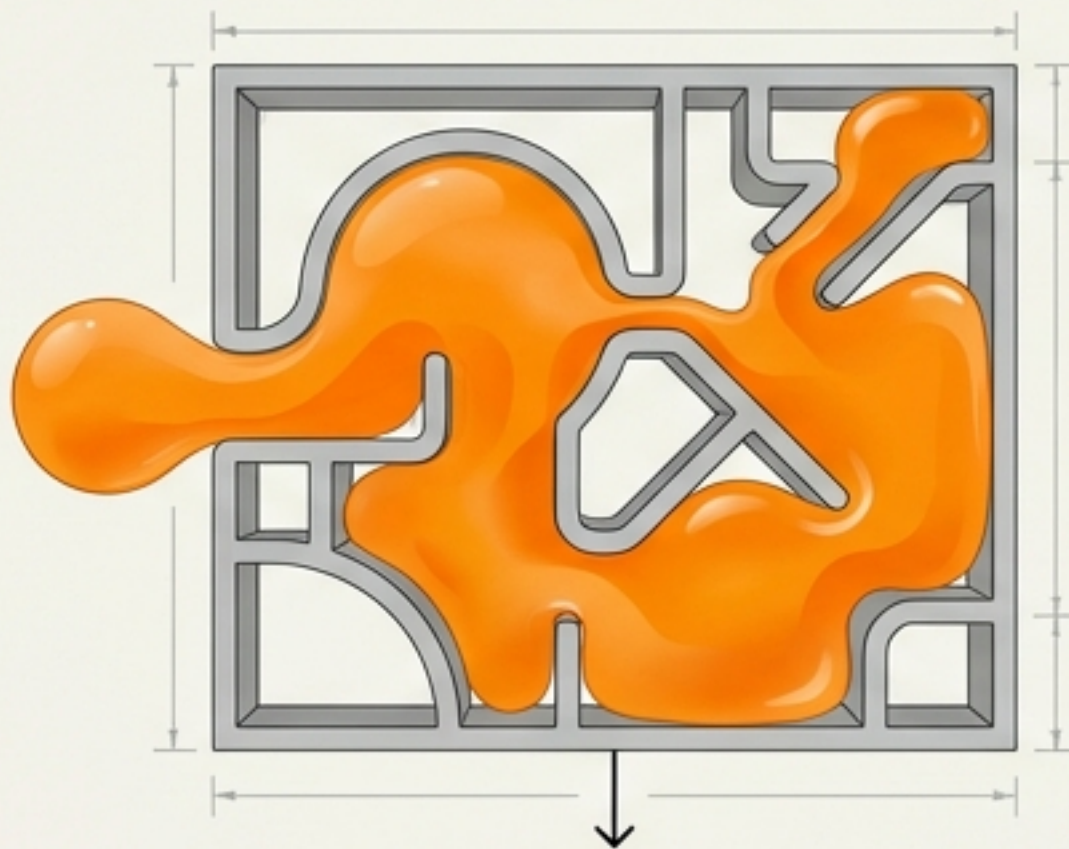
ARC-AGI-2：AIのための「最終障壁」と流動性知能

Crystallized Skills (結晶化されたスキル)



記憶・パターン認識
(Chess, Coding)

Fluid Intelligence (流動性知能)



未知への適応
(ARC-AGI-2)

ベンチマークの哲学

François Chollet氏の定義に基づき、膨大な訓練データの暗記ではなく、未知の問題への適応能力を測定。

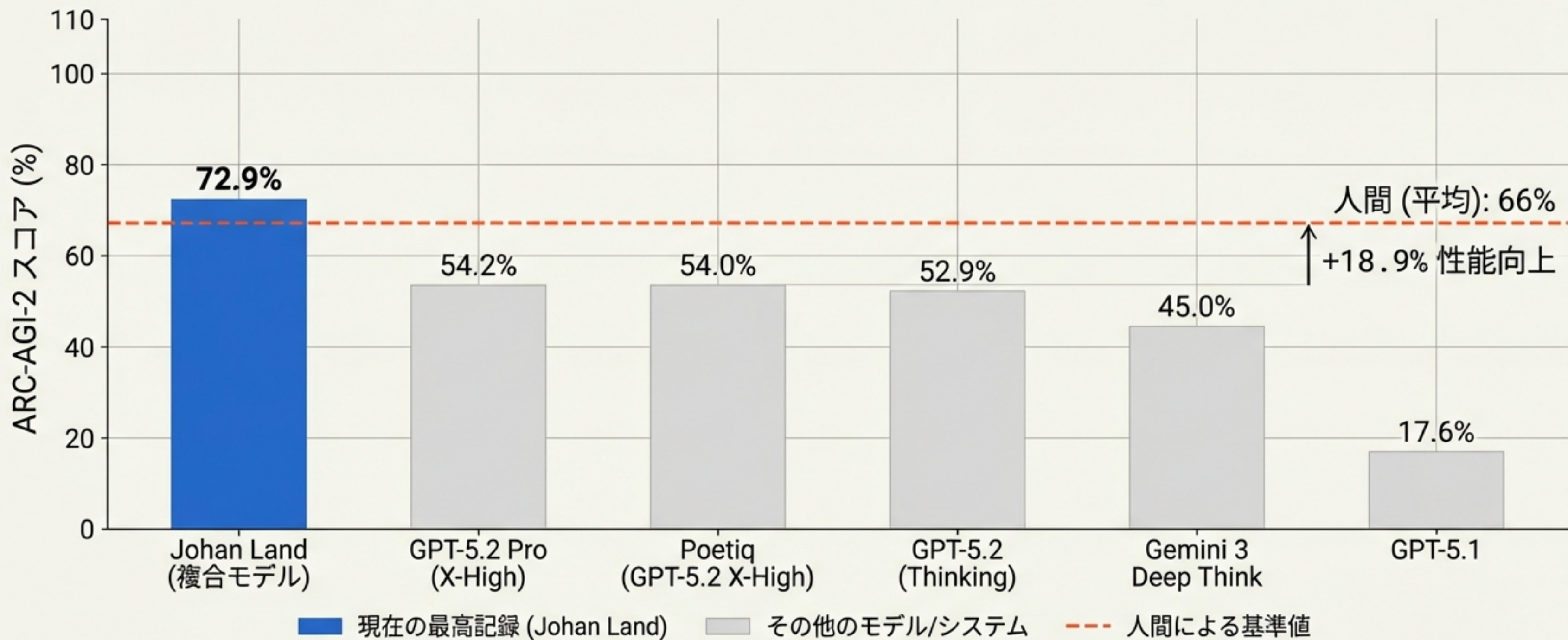
前提知識の制限

人間が先天的に持つ「コア知識 (Core Knowledge Priors)」— 物体認識、数概念、幾何学、トポロジー、エージェント性—のみを前提とする。

ARC-AGI-2の難化

記号の「意味論的解釈 (Symbolic Interpretation)」、ルールの「構成的推論 (Compositional Reasoning)」、「文脈的適用」が強化され、単純なパターンマッチングを無効化。

歴史的な転換点：人間ベースラインの突破



従来のLLMは「記憶と補間」に依存していたため、未知への適応 (Extrapolation) が困難だったが、この +18.9%の飛躍は質的な能力の変化を示している。

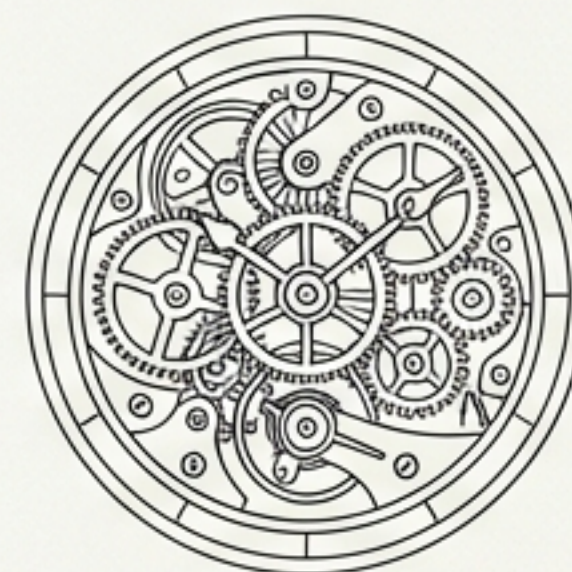
パラダイムシフト：「System 1」から「System 2」への移行

Traditional LLM (System 1)



即座に確率的なトークンを出力
(Fast Thinking)。直感・反射。

Land's Architecture (System 2)

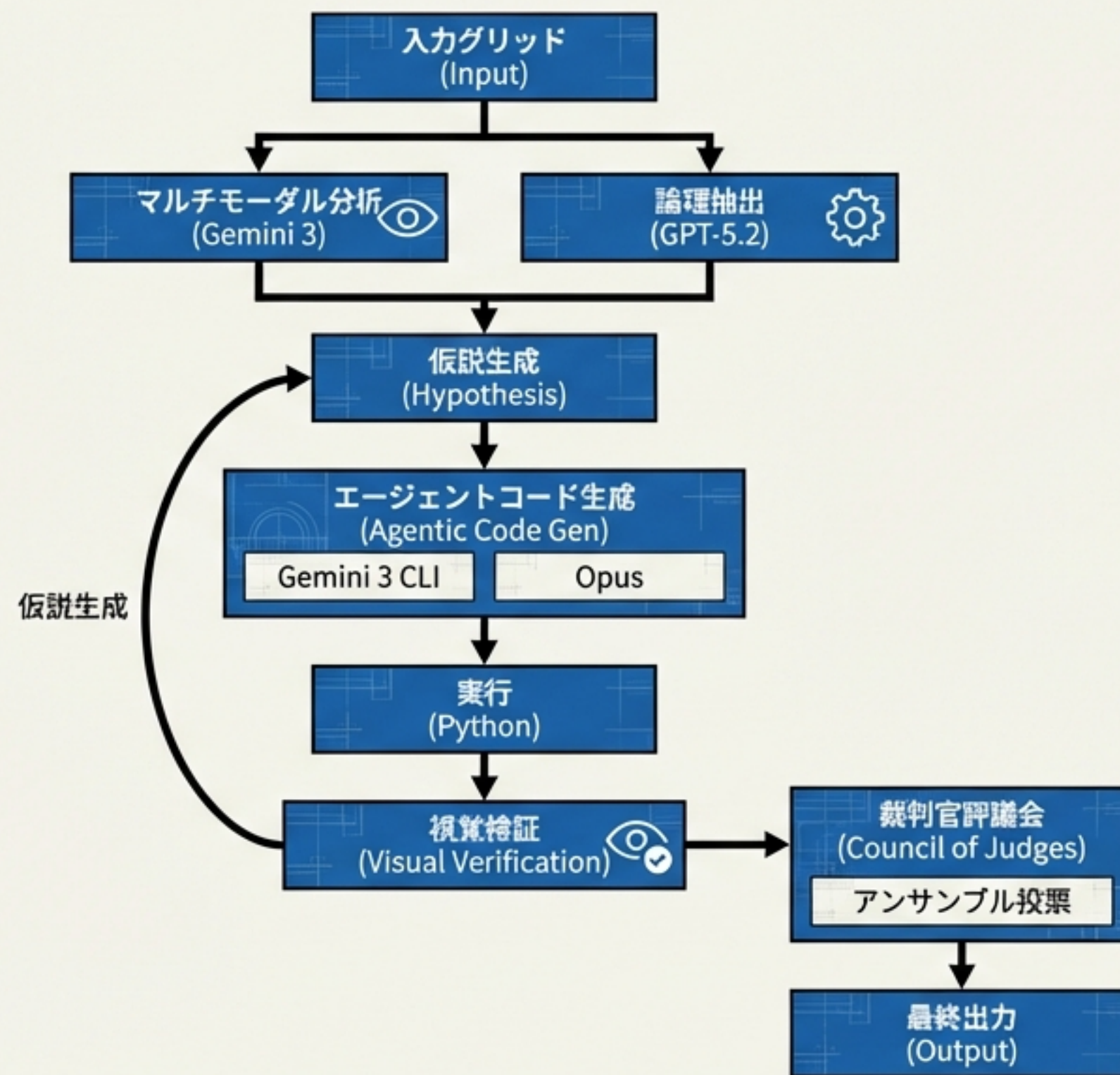


1問あたり平均 **6時間**の熟考
(Slow Thinking)。熟慮・検証。

Multi-Model Reflective Reasoning（マルチモデル再帰的推論）

単一の巨大モデルではなく、複数のフロンティアモデルを動的に連携させる高度なアンサンブルシステム。
推論プロセスを外部化し、試行錯誤のループを回すことで正解に到達する。

アーキテクチャ概観：推論プロセスの外部化



最適な認知資源の配置：モデル・アンサンブル戦略

THE ARCHITECT

GPT-5.2 (Logic Core)



システムの論理的バックボーン。
「Thinking」モードやX-High推論
を活用し、複雑な文脈保持とマルチ
ステップの論理構築、オーケスト
レーションを担当。

THE VISIONARY

Gemini 3 (Visual Cortex)



ネイティブ・マルチモーダル能力
で、対称性、包含関係、接続性な
どの視覚的パターンを「直感」的
に把握。

THE CRITIC

Claude Opus 4.5 (Coder/Lateral)



長文脈での安定性とコーディング
能力。他のモデルが見落とした
エッジケースの指摘や、異なる視
点からの解法提案（Lateral
Thinking）を行う。

Agentic Code Generation : 思考の実践的検証



Hypothesis to Code

AIは「頭の中」だけで考えるのではなく、仮説を検証するためのPythonスクリプトを記述する。



Active Inference (能動的推論)

プログラムを実行し、その結果（生成されたグリッド）を見てコードを修正する「試行錯誤」のループを超高速で回転させる。



Result

モデルのハルシネーション（幻覚）を排除し、実行可能なロジックのみを抽出。

100,000+

Python Executions per Task

```
def execute_code(code):
    code_executor = wp.execute()
    params, results = {}
    def execute_code():
        for it, in in code_lines(244):
            code_executor = A.execute(code_executor)
            print(f"Execution: {code_executor} {code_executor}")

    for __null__ in bues():
        for execute_code: _ferny, tzm12 < wp.execute_code():
            n = doc.execute_code_executor(8))
            execute_code_executor(8))

    return results

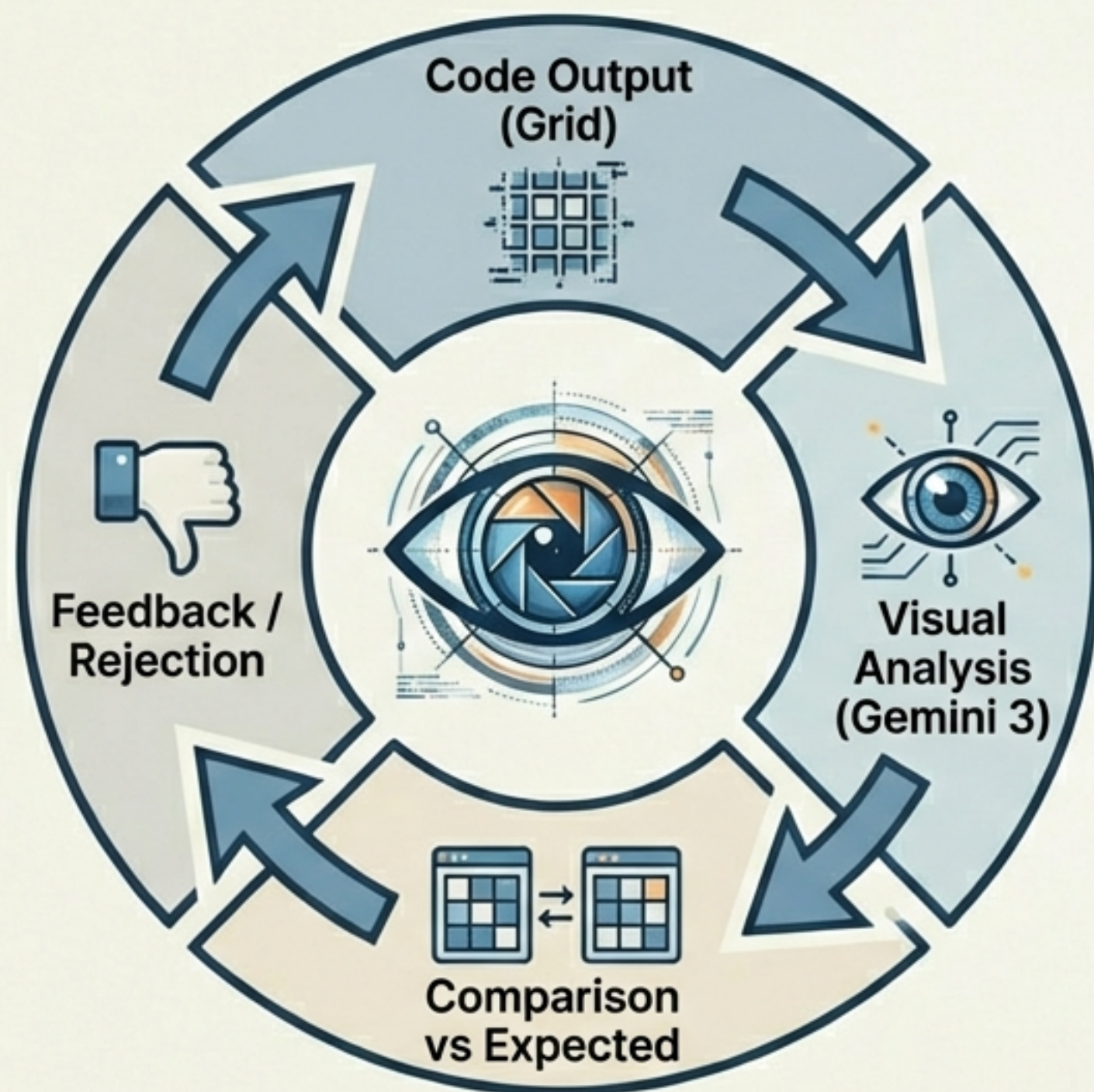
else:
    print(f"Check: {code_executor} {code_executor}")
    return results

if code in execution:
    code_executor = Python.execute_code_executor()
    return execute_code_executor(8))
else:
    print(f"Execution: {code_executor} {code_executor}")

code_executor = execute_code_executor()

execute_code_executor(8))
print(f"Execution: {code_executor} {code_executor}")
```


視覚的検証と再帰的ループ



Visual Reasoning

コード実行によって生成された出力グリッドを、Gemini 3が再度視覚的に評価。

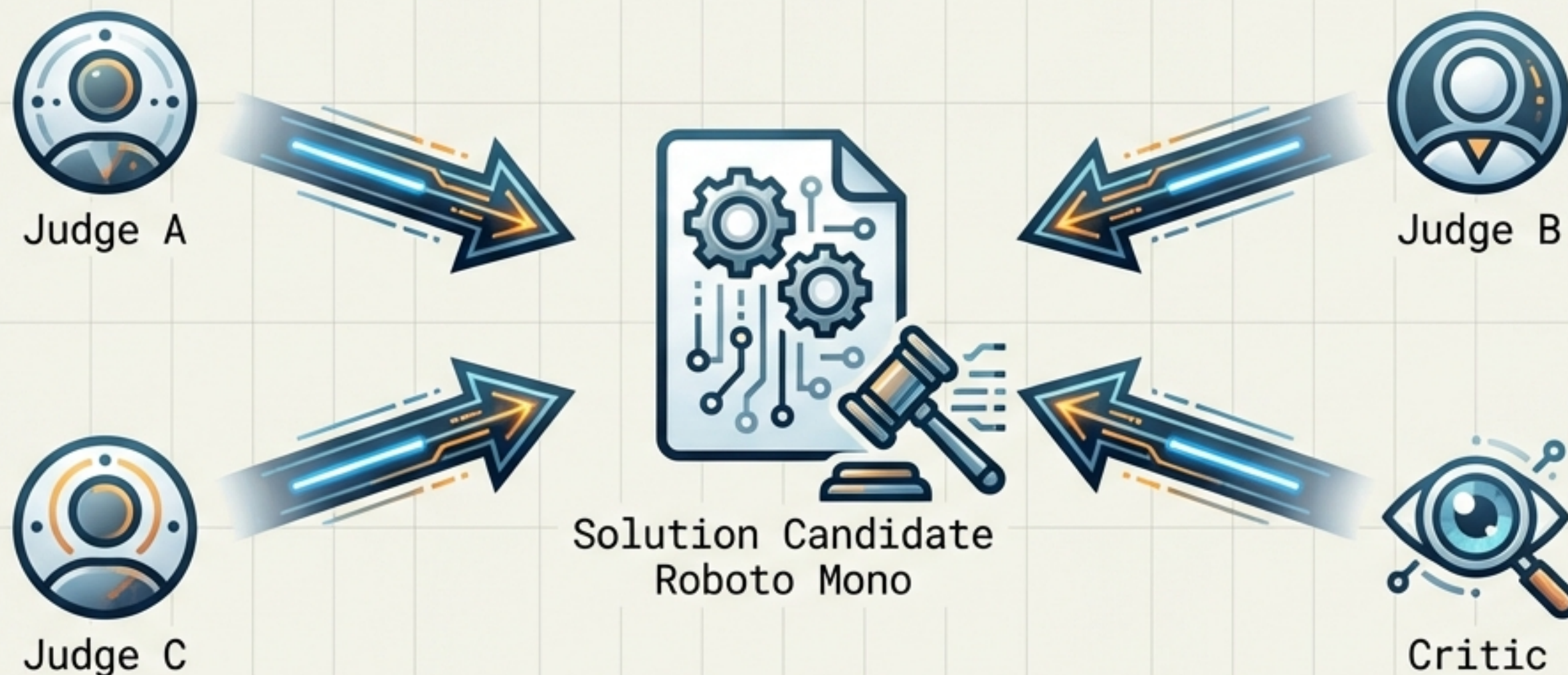
Verification Criteria

期待される出力と視覚的に一致しているか？
新たな予期せぬパターンが出現していないか？

Process

テキスト（コード）と視覚（グリッド）の往復により、人間がパズルを解く際の「目で見えて確認する」プロセスを模倣。

Council of Judges：合議による品質保証



審査員エージェント

複数のエージェントが生成した解法に対し、別のエージェント群がその妥当性を評価・投票する。

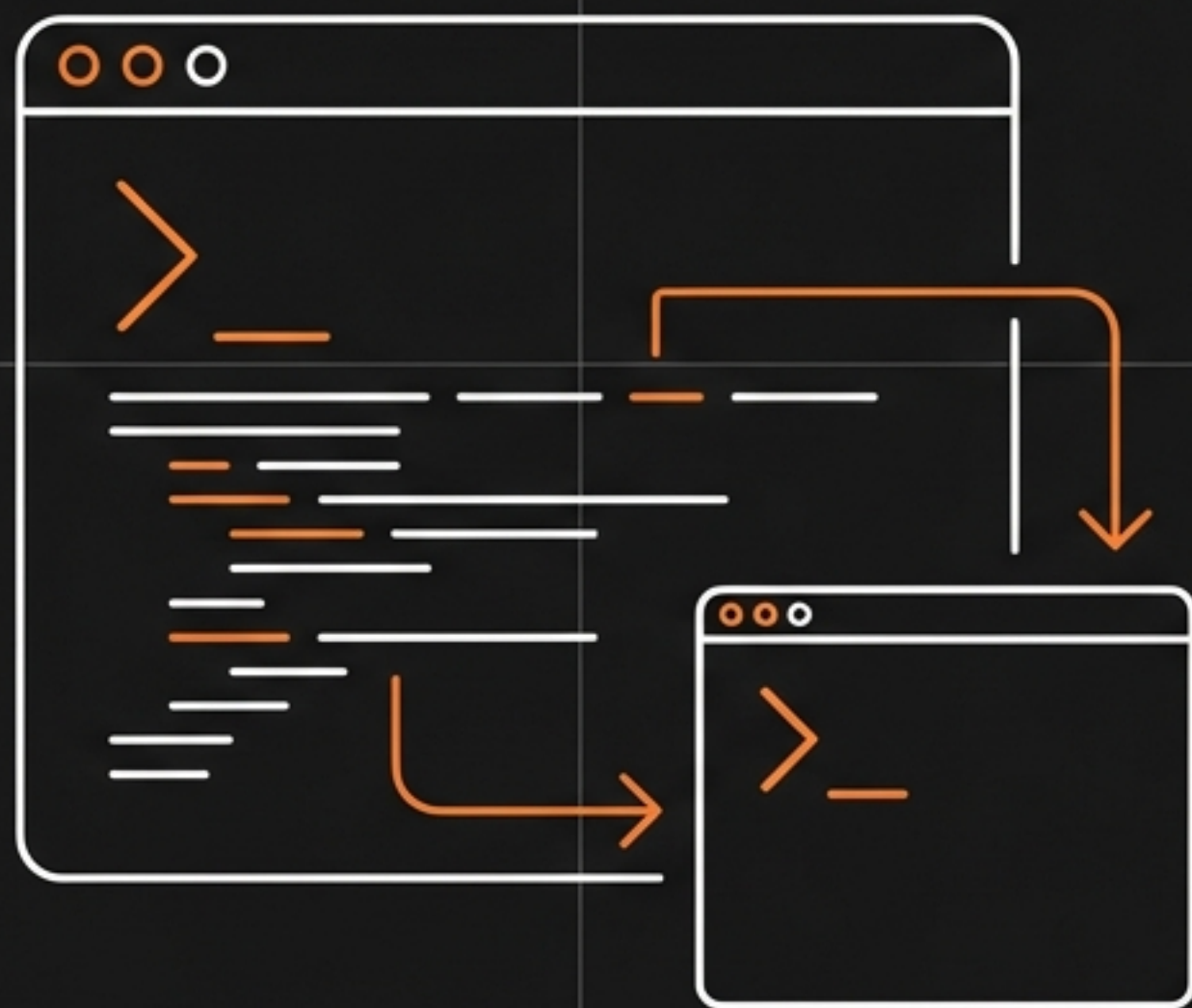
ハルシネーションの抑制

単一モデルの「もっともらしい嘘」を、異なるアーキテクチャやプロンプトを持つモデルが批判的に検証 (Debate with Judge)。

多様性の確保

全会一致で採択された解法は、特定のモデルのバイアスに依存しない普遍的なロジックである可能性が高い。

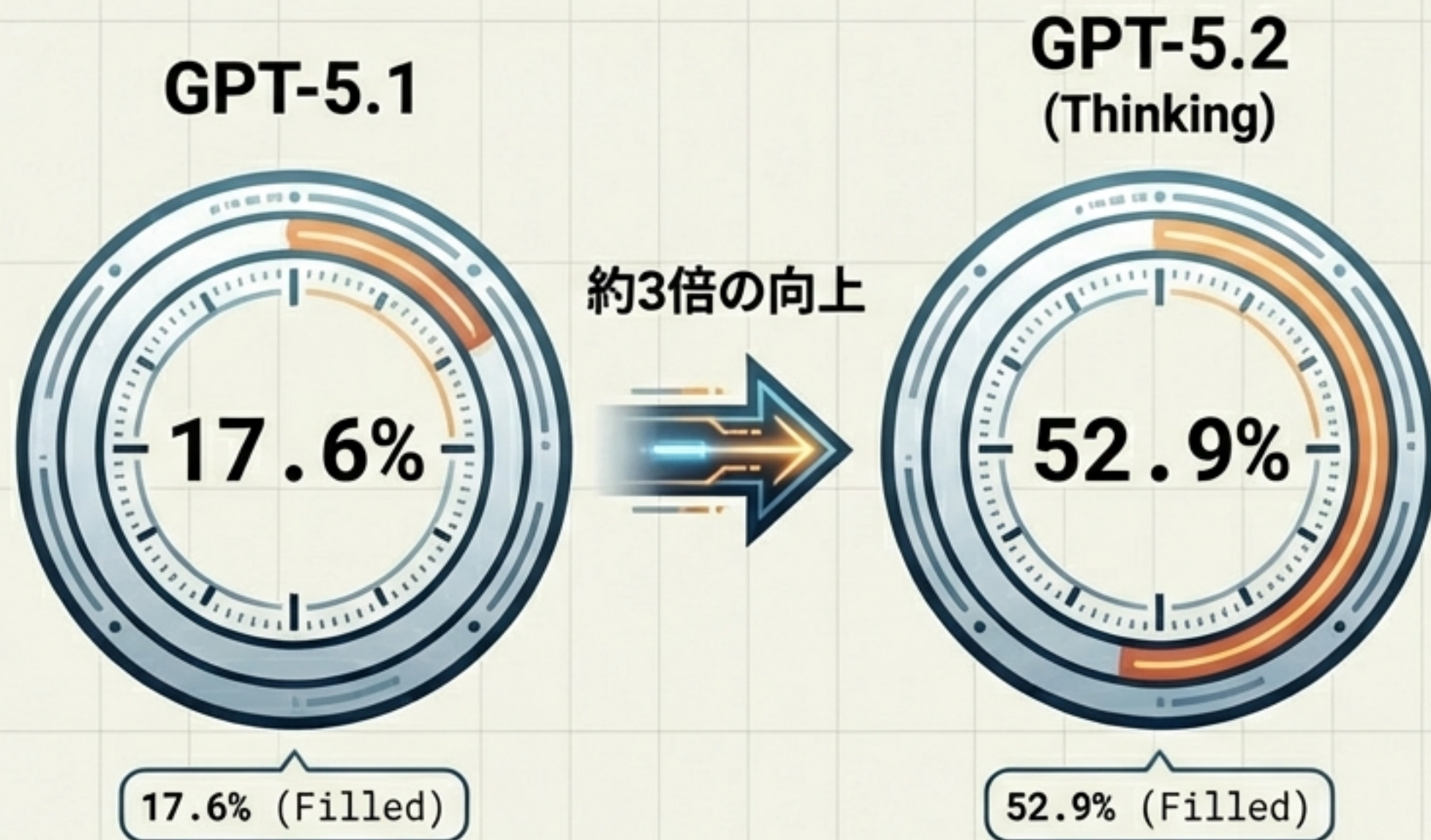
再帰的自己改善の萌芽：AIによるAIの開発



Gemini-3-CLI as the Coder

- Land氏の報告によれば、ソルバー自体のコードはGemini-3-CLIによって記述された（メタプログラミング）。
- ARC-AGI-2を解くためのシステム自体を人間が手書きしたのではなく、AIがコーディングエージェントとして構築。
- これは「AIがより優れた問題解決システムを生み出す」という再帰的自己改善の初期的な事例であり、SOTA達成のプロセス自体がAGI的性質を帯びている。

基盤モデルの進化：GPT-5.2の貢献



Thinking Mode (X-High)

ユーザーが推論の深さを調整可能。回答前に内部で膨大な「思考トークン」を消費し、自己検証を行う。

SWE-bench Verified 80.0%

高いコーディング能力が、システム内のPython生成プロセスの成功率を担保した。

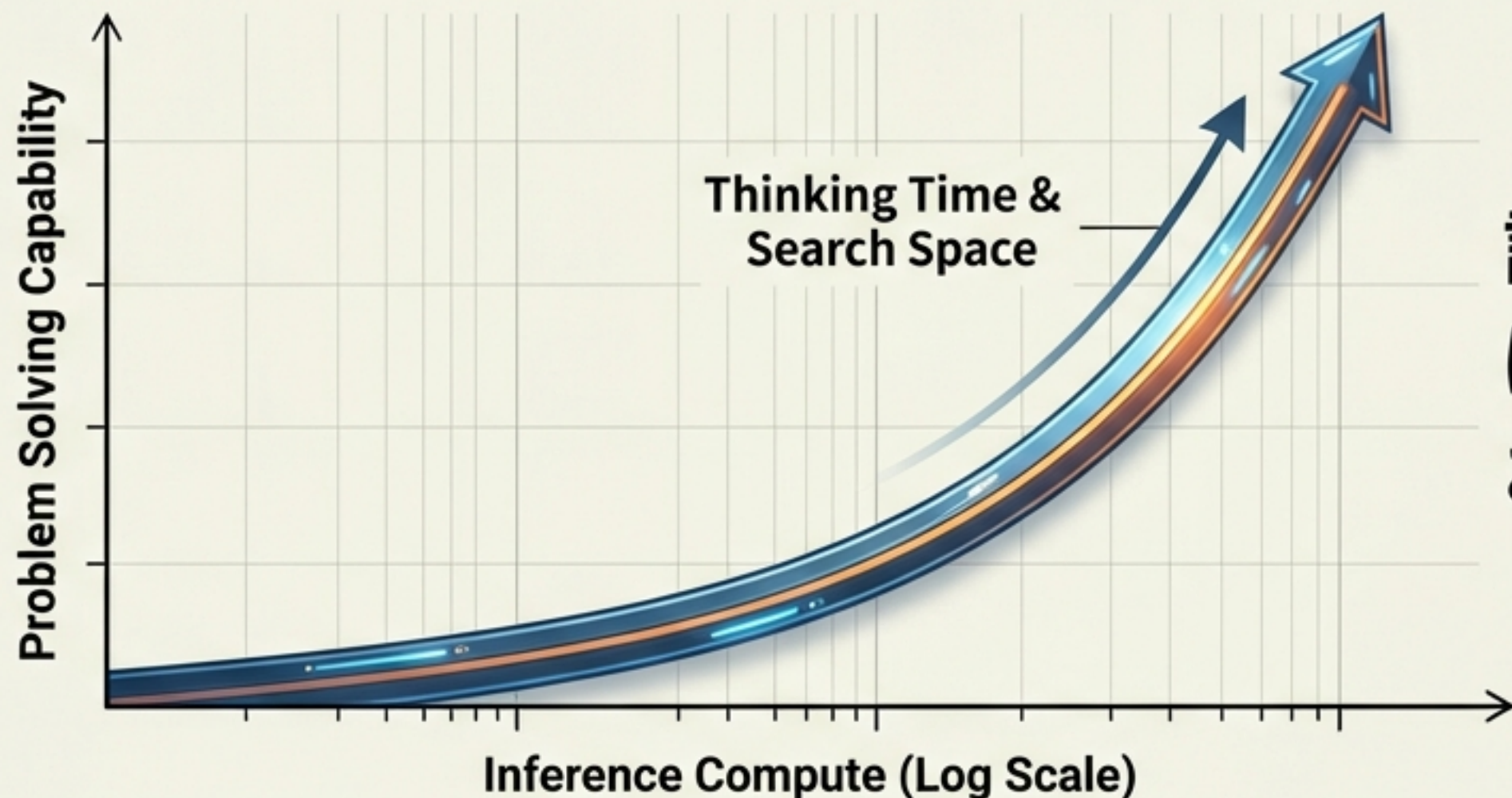
「知能」のコスト：精度と効率のトレードオフ

System	Score	Cost per Task	Notes
Johan Land (v2)	72.9%	\$38.90	10万回のコード実行
Poetiq (GPT-5.2)	54.0%	\$8.0 – \$30.0	メタシステムによる最適化
GPT-5.2 Thinking	52.9%	~\$1.90	Thinking Mode
NVIDIA (NVARC)	27.6%	\$0.20	軽量・高効率

Land氏のアプローチはNVIDIAの軽量手法の約200倍のコスト。「富豪的推論」による力技。現状では「精度」を購入するために指数関数的な計算リソースを投入している。

推論時計算のスケーリング則

Inference Scaling Laws



計算リソースさえ十分に投入すれば、AIは未知の抽象推論タスクにおいて人間を超えられる。

Implication:

これまでは「汎化では「汎化できない（解けない）」と思われていたタスクも、推論時の探索空間（Inference-time Search）を拡大することで解決可能であることが証明された。事前学習（Pre-training）の規模拡大だけでなく、思考時間（Thinking Time）の拡大が新たな性能向上の軸となる。

ARC-AGI-2突破が示す3つの示唆

01.

「システム」としての勝利

単体モデルではなく、Python実行環境やマルチエージェントを連携させた「Compound AI Systems」が超人的な成果を出す。

02.

推論スケーリングの確立

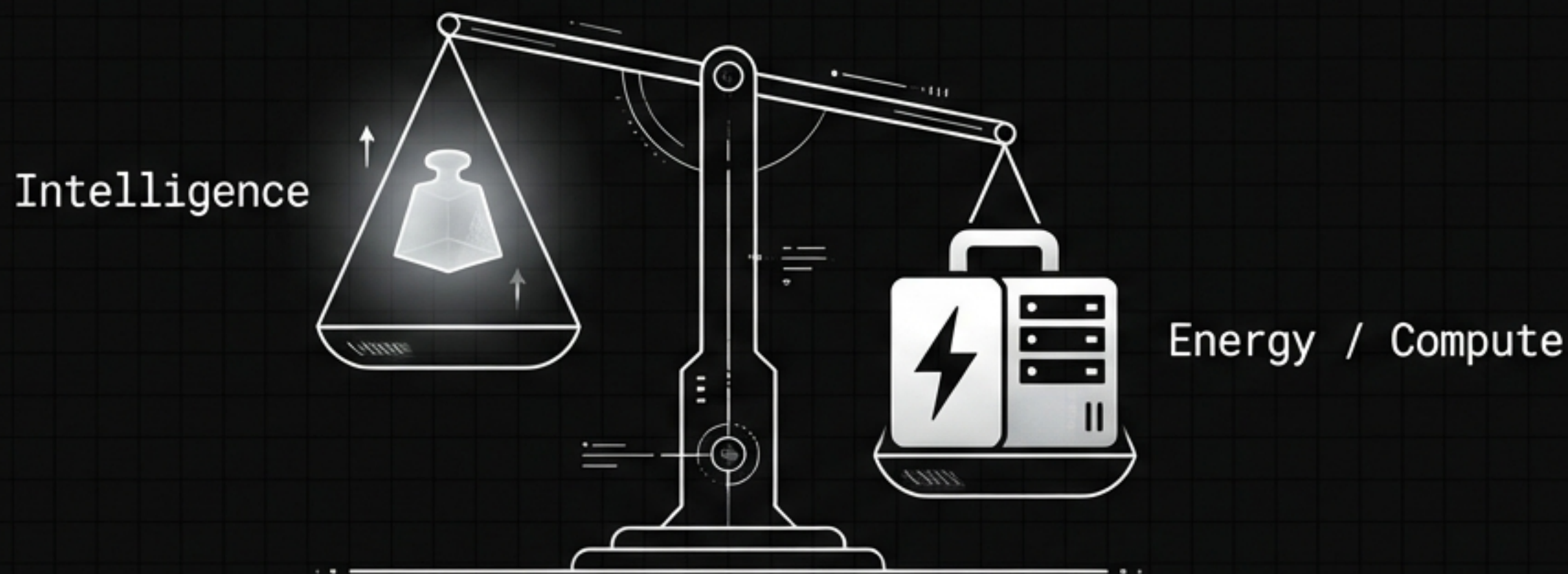
System 1（直感）からSystem 2（熟考）への移行により、AIは自身の限界を超えた難問を解決できる。

03.

新たなボトルネック

「知能の壁」は崩れたが、「効率性の壁」が残る。今後は高スコアを維持しつつコストを圧縮する「蒸留（Distillation）」フェーズへ。

結論：AGIは科学からエンジニアリングへ



François Chollet氏が「AIには解けない」として設計したARC-AGI-2の壁は、力技と知恵の融合によって破られた。

Land氏の成果は、AGIの実現がもはや「未知の魔法」ではなく、「エネルギーと計算資源の最適化問題」に移行したことを示している。

The barrier is no longer intelligence, but energy.