ChatGPT

NVIDIA AVOのARC-AGI-3「100%」達成は何を意味するのか

エグゼクティブサマリー

NVIDIAは2026年8月21日、長期自律エージェント基盤 **AVO（Agentic Variation Operators）** にClaude Opus 5を組み合わせ、ARC-AGI-3の**25環境・183レベルからなるPublic Setで100.00 RHAE**を達成したと報告した。全183レベルを **6,624 environment actions** で完了しており、同じClaude Opus 5を用いたVISTAの7,542 actionsより約12%少ない。AVOの中心は新しい基盤モデルではなく、**永続メモリ、長期コンテキスト管理、ツール実行、計画、実世界フィードバック、停滞を検知して介入するsupervisor**を一体化したエージェント・ハーネスである。NVIDIAは同じ基本アーキテクチャを、それ以前にGPUカーネル最適化へ7日間連続で適用していた。 1

しかし、この結果を「ARC-AGI-3を完全攻略した」「AGIに到達した」と解釈するのは不適切である。**100%なのは公開済みPublic Demo Setだけ**であり、ARC Prize自身が、公開セットは過適合や事前露出の影響を受け得るため、公式な「benchmark solved」の判定にはsemi-private / fully-privateの初見環境を用いる方針を明示している。Technical ReportではPublic Demoが25環境、Semi-Privateが55、Fully Privateが55環境で構成され、さらにPublic Setの成績を公式リーダーボードの成績とはみなさないとしている。 2

さらに、**AVOはPublic Setで最初の100点システムでもない**。現在公開されている一次資料では、VISTA+Claude Opus 5が100.00、Tycho+Claude Opus 5およびTycho+GPT-5.6 Solも100.00を報告している。Tycho+Opus 5は6,641 actionsであり、AVOの6,624は比較した公開100点ランの中でわずか17 actions少ない。一方、Tychoは明示的な実行可能world modelを構築し、VISTAは画像ベースのlossless visual memoryを使うのに対し、AVOは**明示的なプログラム世界モデルを持たず、64×64の正確なテキストgridだけで100点に達した**点が技術的に特徴的である。 3

最も重要な意味は、**「基盤モデルの能力」から「エージェント・システム全体の能力」へ評価単位が移りつつあることを強く示した点**にある。Claude Opus 5単体のARC Prize評価は30.16%だったのに対し、AVOシステムでは100%となった。ただしNVIDIA自身が認める通り、推論設定、観測表現、メモリ、実行ループなどがすべて異なるため、これはAVO単独の寄与を測ったcontrolled ablationではない。OpenAIもGPT-5.6 Solについて、モデルを変えずにreasoning保持とcontext compactionだけでPublic Setを13.3%から38.3%へ改善し、出力tokenも約6分の1にした。Twinでも同じGPT-5.6 Solについて、通常Codex harness 61.1からworld-model harness 93.3への改善が観測されている。したがって「harness matters」はAVOだけに依存しない再現性の高い傾向である。 4

一方、AVOのARC実験については、**token使用量、API呼び出し数、wall-clock時間、総推論費用、seed間分散、失敗trajectory、memory/supervisorの個別ablation、ARC用実装・完全prompt**がNVIDIAの公開資料では明らかにされていない。このため「6,624 actionsという環境効率」と「計算効率」は区別しなければならない。ARC-AGI-3は内部推論やツール呼び出しをaction costに数えないからである。Twinはこの盲点を定量化しており、93.3点に2.60B processed tokens・91.4時間を使い、61.1点のno-Twin ablationより多くの内部計算を投入して環境actionを減らしている。 5

本稿の総合評価は次の通りである。AVOの100%は、AGI達成の証拠としては弱いが、**長期エージェント設計におけるmemory・supervision・state continuity・closed-loop feedbackの重要性を示す非常に強いシステム工学上の結果**である。ただし、Public Set、学習データ汚染の可能性、hidden compute、再現性不足

という四つの大きな留保がある。特にClaude Opus 5のknowledge/training cutoffは2026年5月であり、ARC-AGI-3 Public Demoの公開後であるため、公開課題が学習データに含まれていなかったことは、Anthropicの公開情報からは保証できない。 [6]

ARC-AGI-3とは何を測っているのか

ARC-AGI-3は従来の静的なARC問題とは異なり、エージェントが**説明のない未知のインタラクティブ環境へ入り、行動し、結果を観察し、ルールと目的を推測し、より難しい後続レベルへ知識を持ち越す**能力を測るベンチマークである。ARC Prizeはこれをexploration、world modeling、goal discovery、planning/executionを要求するinteractive reasoning benchmarkとして設計している。人間には解ける一方、初期のfrontier modelはsemi-private環境で1%未満に留まった。 [7]

各環境は最大64×64セル、16色の抽象grid worldであり、使用可能な操作は方向キー系、選択操作、Undoなどに限定される。ゲーム固有のルール、ゴール、自然言語による攻略説明は与えられない。これは画像認識そのものより、**未知の因果関係を少数のinteractionから推論すること**を評価する設計である。環境はcustom engine上で高速に実行され、難易度は個々の知覚問題よりも、複数のmechanics、長期依存、希薄な成功feedback、未知のgoalの合成から生じる。 [8]

スコアリングと「100%」の正確な意味

現在のRHAE（Relative Human Action Efficiency）では、レベルごとに、人間の初見プレイヤーが要したaction数 h とAIのaction数 a を比較して、おおむね

$$s_\ell = \min\left(1.15, \left(\frac{h_\ell}{a_\ell}\right)^2\right)$$

という効率点を計算する。後半レベルにはより大きな重みが与えられ、ゲーム全体の点数はcompletionによる上限を受け、最終的に25ゲームの平均が0–100点になる。内部CoT、Python実行、LLM呼び出し、read-only inspectionなどはenvironment actionに含まれない。人間基準は初見プレイ記録から作られる。 [9]

ここには重要な細部がある。**RHAE 100は「183レベルすべてでAIが人間より少ないactionだった」という意味ではない。** 個別level scoreは最大1.15まで上振れできるため、あるレベルで人間より遅くても、別のレベルで人間を大きく上回ればgame score 100になり得る。実際VISTAの公開表には、人間48 actionsに対しagent 50 actions、あるいは人間33に対しagent 56でも、他レベルの高効率によってゲーム全体100になっている例がある。100が保証する重要な事実は、全ゲームでcompletion capを満たすため**全183レベルを完了したこと**である。 [10]

公開・非公開セットの違い

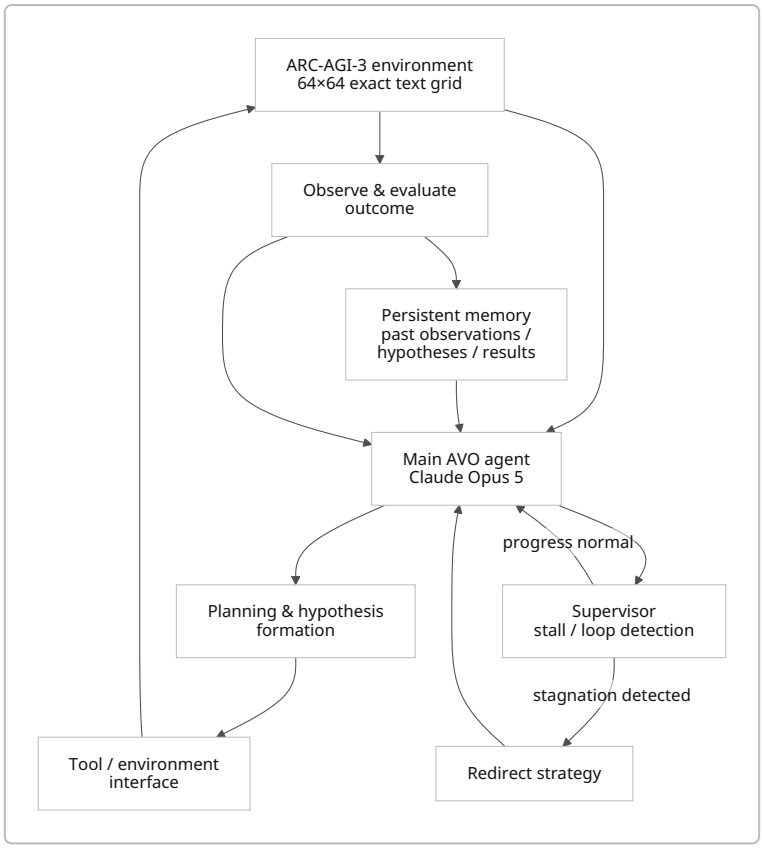
評価集合	環境数	公開性	AVO結果	解釈
Public Demo	25	課題・環境が公開	**100.00**	強いが過適合・汚染を排除できない
Semi-Private	55	非公開評価	未報告	公式な一般化評価に近い
Fully Private	55	完全非公開	未報告	最も強い初見一般化テスト

ARC Prize Technical Reportは、Public Setは開発・デバッグ・研究比較用であり、公開環境に特化したharnessを作れるため、**public scoreをofficial leaderboard scoreにしない**と明記している。また公開セットの方がprivate setより実質的に易しいとしている。したがって「AVOがARC-AGI-3で100%」という見出しは、厳密には**「ARC-AGI-3 Public Demoで100%」**と読む必要がある。 [2]

この区別こそが100%の「凄さ」と「限界」を同時に説明する。183レベル全完走は長期適応性能として顕著だが、ARC Prizeが本来測ろうとしている「訓練時にも開発時にも見ていない新環境へのskill-acquisition efficiency」を確定するにはprivate評価が必要である。[11]

AVOアーキテクチャと100%を可能にした技術

AVOは、パラメータを持つ新しいfoundation modelではない。NVIDIAが開発した**general-purpose coding/agent harness**であり、その中核推論器としてfrontier LLMを使う。元論文では、進化的探索の固定的な「sample → generate → evaluate」を、履歴、knowledge base、評価関数を直接見ながら自律的に候補を作る **Agentic Variation Operator** に置き換える。ARC-AGI-3では、この「variation operator」という進化的最適化をそのままゲームに適用したのではなく、同じgeneral-purpose agent loopを別のenvironment interfaceへ接続した、とNVIDIAは説明している。[12]



この図はNVIDIAが公開したAVO構成を抽象化したものだ。main agentはcontextを調査し、計画し、行動し、結果を評価するループを回し、supervisorがより大域的なtrajectoryを監視する。[13]

構成要素と公開情報

項目	AVO / ARC-AGI-3で判明していること	未公開・不明
基盤LLM	Claude Opus 5	**パラメータ数非公開**
AVO自体の「model size」	神経モデルではなくagent systemなので該当しない	harness自体のコード規模
入力	正確な64×64 text grid、画像token なし	grid serializationやpromptの完全仕様

項目	AVO / ARC-AGI-3で判明していること	未公開・不明
perception	symbolic/textual、VLM視覚認識を使わない	noisy perceptionでの性能
planning	main agentによるhypothesis → plan → action → revise	planning algorithmの詳細・探索幅
memory	過去の状態、評価、reasoning等を持続的に保持	storage形式、compression、reset boundary、最大容量
supervisor	stall・反復的失敗を検知し別方向へ誘導	trigger閾値、supervisor model、具体prompt
world model	Tychoのような明示的Python simulatorなし	内部自然言語表現の構造
RL / IL	**ARC実験用の追加RL/ILは報告なし**	基盤Opus 5の詳細post-training recipe
ARC用weight update	報告なし。test-time interaction中心	非公開内部調整の有無を独立検証不可
training dataset	AVO固有training setはなしとみられる	Claude Opus 5の正確なdataset量・token数

NVIDIAの元論文ではpersistent memoryはconversation historyを通して、過去のcode edits、compiler outputs、profiler results、reasoningを保持すると説明され、git commitによって探索状態も永続化する。停滞や無益なedit loopをself-supervisionが検出すると、trajectory全体を見て新たな候補方向へredirectする。ARC版でもNVIDIAはpersistent memoryとsupervisionを中心要素として挙げるが、ARC固有のmemory実装詳細までは公開していない。 14

基盤モデルの学習データという重要な留保

AnthropicはClaude Opus 5について、公開Internet情報、public/private datasets、user data、synthetic dataのproprietary mixtureでpretrainingし、その後substantial post-trainingを行ったと説明している。一方で正確なparameter count、training tokens、個々のデータセット一覧は公開していない。knowledge/training cutoffは**2026年5月**とされる。 15

これはARC評価では特に重要である。ARC-AGI-3 Public Demoは2026年3〜4月には公開されており、Opus 5のcutoffはその後になる。したがって、**Public Demoのゲーム、replay、解説、コード等がOpus 5のtraining mixtureに絶対に含まれていないとは公開情報から証明できない。** これはAVOに不正があったという意味ではないが、Public Set 100%を「真の初見一般化」と同一視できない理由になる。ARC Prizeがprivate setを重視する設計思想とも整合する。 6

技術的に何が新しいのか

AVOで最も重要なのは、新規のRL algorithmや画像encoderではなく、**agentic state management**である。具体的には、①長期trajectoryを捨てないpersistent memory、②ローカルなmain-agent推論と大域的なsupervisorを分離、③ツール・実行結果をground truth feedbackとして使う、④失敗を終端ではなく次の仮説へのdata pointとして蓄積、⑤同じloopをCUDA最適化から未知ゲームへ移植したこと、の組み合わせである。 16

特に**multimodalityが100%の原因ではない**点は重要である。Claude Opus 5自体は画像入力可能だが、AVO ARC runでは画像を一切送り込まず、正確なtext gridのみを用いた。VISTAが512×512 PNGとlossless visual

memoryを主構成とするのとは対照的である。したがって今回証明されたのはvisual perceptionの強さというより、**記号化済み観測上での因果推論・state retention・long-horizon control**である。[17]

RL/ILとの関係についても区別が必要である。公開されたARC版AVOに、environment rewardからweightsを更新するonline RLや、demonstrationを用いたimitation learningは報告されていない。「learning」は主に**test-time memory/context内の状態更新**であり、weight learningではない。元のAVO論文の「evolutionary search」もLLMの重みを進化させるものではなく、CUDA実装候補を探索・評価・commitする仕組みである。[18]

実験結果、アブレーション、計算量と再現性

ARC-AGI-3でのheadline resultは明快である。Claude Opus 5 + AVOはPublic Setの25/25 games、183/183 levelsを完了し、100.00 RHAE、6,624 environment actionsを記録した。VISTA + Opus 5も同じ183レベルを完了したが7,542 actionsだったため、AVOは約12%少なかった。ただしNVIDIA自身が、観測形式、backend、memory、context管理、実装が異なるため**controlled ablationではない**と明記している。[19]

公開されている実証結果

実験	主指標	結果	計算情報
AVO + Opus 5 / ARC Public	RHAE	**100.00、183/183、6,624 actions**	tokens、calls、wall time、費用は未公表
VISTA + Opus 5	RHAE	**100.00、183/183、7,542 actions**	完全replayあり、総token未記載
Tycho + Opus 5	RHAE	**100.00、183/183、6,641 actions**	API-equivalent約**$2.99k**
Twin + GPT-5.6 Sol	RHAE	93.3、179/183、11,614 actions	**2.60B processed tokens、91.4 h**
AVO / attention kernel	throughput	cuDNN比最大+3.5%、FA4比最大+10.5%	B200、連続7日、>500方向、40 commits

AVO・VISTA・TychoのARC値は各一次資料から、Twinのcomputeは論文のrun artifacts集計から、AVO kernel値はNVIDIA/AVO論文から採った。[20]

AVOとTychoはほぼ同じenvironment-action frontierに位置しており、6,624対6,641の差はわずか17 actions、約0.26%である。この差自体からarchitectureの優劣を結論づけるべきではない。両者はworld-model policy、prompt、内部compute、モデル呼び出し、context designが異なるからである。比較としてより意味が大きいのは、VISTAとの差そのものより、**明示的world modelありのTychoと、直接interaction型AVOが別々の設計で100へ到達したこと**である。[21]

AVOで存在するアブレーションと、存在しないアブレーション

AVO論文にはGPU kernel側の具体的なablationがある。例えばbranchless accumulator rescalingの導入は一部non-causal設定で約+8.1%、correction/MMA overlapは追加の改善、register rebalancingも特定条件で改善をもたらした。元研究は最終結果だけでなく、40世代近いcommit lineageを通じて複数の最適化が積み上がることを示している。[22]

しかし、**ARC結果については同レベルのcomponent ablationが公開されていない**。したがって現時点では、

$$100 = f(\text{Opus 5},\ \text{memory},\ \text{supervisor},\ \text{text grid},\ \text{prompt},\ \text{reasoning effort},\ \text{tools},\ \text{context policy})$$

の各要素の寄与を分解できない。特に「persistent memoryが何点を上げたか」「supervisorを外すと何点落ちるか」「画像入力ならどうなるか」「通常contextだけでは何点か」は未回答である。NVIDIA自身もVISTA比較はmemoryの効果をisolateしていないと明示している。[23]

対照的にTwinは、同じbase agent/modelを保ったままworld-model harnessを除去する比較で**93.3 → 61.1**を報告している。さらにOpenAIはGPT-5.6 Solについてreasoning retentionとcompactionだけを変えて**13.3 → 38.3**、かつoutput tokens約6分の1を報告している。この二つはAVOの「システム構成がモデルの潜在能力を大きく変える」という主張を独立に支持するが、AVO固有componentの因果効果までは証明しない。[24]

Computeと費用をどう読むべきか

ARCのRHAEは内部computeを課金しないため、**action efficiency ≠ compute efficiency**である。Twinではworld modelありの方がenvironment actionsを半減方向へ押し下げる一方、no-Twinより総processed tokensが1.06Bから2.60Bへ増えている。論文自身が、action efficiencyを買うためにtest-time computeを前倒ししていると結論づけている。[25]

AVOのARC runについては総token量が公開されていないため、信頼できる総額推計はできない。2026年8月23日時点のClaude Opus 5標準API価格はfresh input **$5 / MTok**、output **$25 / MTok**であり、cache利用やFast Modeでは別料金となる。したがって、AVOの基本API費用は概念的には

$$C \approx 5I + 25O + C_{\text{cache}} + C_{\text{runtime}}$$

（I, O は百万token単位）となるが、NVIDIAが I, O を公表していない以上、ドル額を置くのは推測になる。[26]

比較用にTychoはOpus 5の公開token accountingから**約$2.99k/25 games**のAPI-equivalent estimateを報告している。GPT-5.6 Sol版は当時の価格換算で約$4.47kだった。これはAVOの費用を意味するものではないが、「Public Set 100」を数千ドル規模のAPI inferenceで実現できる例が既にあることは示す。[27]

システム	Environment actions	内部compute	公開費用情報	比較可能性
AVO / Opus 5	**6,624**	**不明**	**不明**	actionのみ比較可能
Tycho / Opus 5	6,641	token telemetry公開	約$2.99k	比較的高い
VISTA / Opus 5	7,542	完全trajectory公開、総token不明	不明	action比較可能
Twin / GPT-5.6	11,614	2.60B tokens / 91.4 h	cache構成依存	modelが異なる

またGPU-kernel研究では「B200 GPUsで7日間」という情報はあるものの、論文からは総GPU台数を確定できないため、**GPU-hoursや電力費を精密に推定することもできない**。[28]

再現性については差がある。TychoはApache-2.0のopen-source implementationを公開し、VISTAはlevel別trajectory、agent notes、replaysを公開している。Twinもreleased run artifactsからtoken/action集計を再構成している。一方、確認できたNVIDIAブログとAVO論文では、ARC版AVOの完全ソース、system prompt、supervisor prompt、token logs、全trajectoryへの公開リンクは示されていない。そのため**スコアを第三者が同じAVOとして再現するための情報は現状不足している**。[29]

競合、先行SOTA、そして研究の流れ

AVOの意義を理解するには、単一モデルのleaderboardとagent harnessのleaderboardを分離する必要がある。

基盤モデルとエージェントを分けた比較

システム / モデル	ARC-AGI-3設定	Score	中心アイデア
Claude Opus 5 High	ARC Prize model evaluation/Public Demo	**30.16%**	汎用frontier reasoning model
GPT-5.6 Sol + OpenAI memory settings	Public Demo	**38.3%**	retained reasoning + compaction
Continual Harness + Gemini 3.1 Pro	Public Demo	**20.54%**	persistent/self-improving harness
OPINE-World + Opus 4.8	Public Demo	78.4	object-centric executable model
Twin + GPT-5.6 Sol	Public Demo	93.3	validated digital twin
Rodionov verification + GPT-5.6	Public Demo	98.97	executable WM + hard verification
VISTA + Opus 5	Public Demo	**100.00**	visual memory + free-form reasoning
Tycho + Opus 5	Public Demo	**100.00**	programmatic world model
Tycho + GPT-5.6 Sol	Public Demo	**100.00**	同上
AVO + Opus 5	Public Demo	**100.00**	persistent memory + supervisor + direct interaction

Opus 5の30.16はARC Prizeのverified model result、OpenAIの38.3はOpenAI独自Responses API harness、その他のacademic/community systemsは各公開artifactに基づくため、**同一条件の単純なleaderboardではない**。それ自体が現在のARC-AGI-3研究の重要な特徴であり、性能が「model × harness × observation × memory × compute budget」の積として現れるようになっている。[30]

Google DeepMind系については、ARC-AGI-3リリース時の公式semi-private evaluationではGemini 3.1 Pro Previewが0.40%だった。一方、後のcommunity harnessで同モデルを用いたContinual HarnessはPublic Demoで20.54%に達している。ただしprivateとpublic、公式generic harnessとdomain-specific harnessが違うため、これらは直接比較できない。[31]

Anthropic自身のOpus 5は2026年7月24日にARC Prize上で30.16%を記録し、それまでのmodel-only frontierを大きく更新した。その同じmodel familyが、VISTA、Tycho、AVOという異なるagent systemでは100に達

している。この差は「Opus 5が30%の能力しか持たない」のではなく、「generic benchmark harnessが引き出す能力」と「specialized/persistent agent harnessが引き出す能力」が大きく異なることを示す。[32]

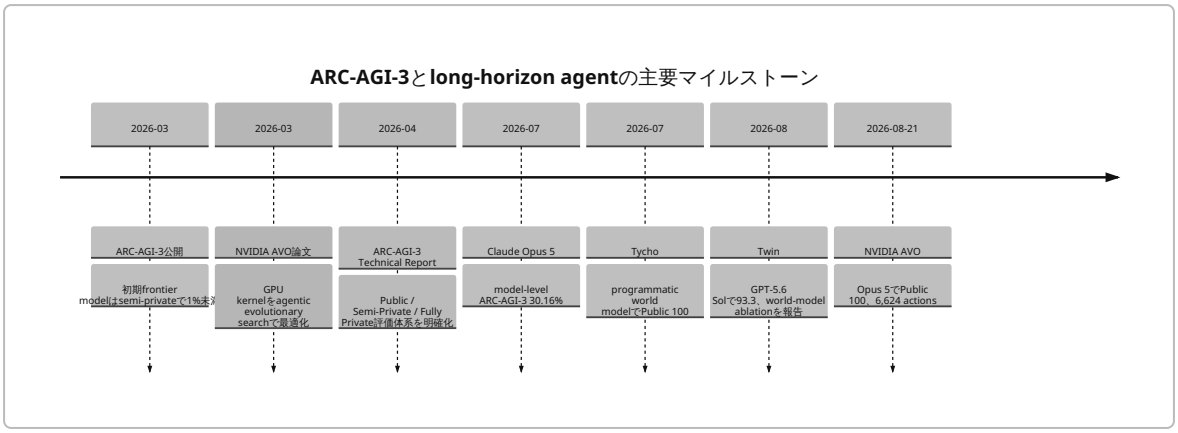
OpenAIの結果はこの解釈をさらに強める。GPT-5.6 Solではreasoningを各turnで捨て、古い履歴をrolling truncationするgeneric harnessが長期学習を阻害していた。reasoning retentionとcompactionを有効にしただけで13.3から38.3へ上昇した。つまり長期agent benchmarkでは、**「モデルに何を考えさせるか」だけでなく「一度考えたことを次のturnでも持たせるか」が一次的な性能変数**になっている。[33]

AVOと主要アーキテクチャの違い

特徴	AVO	VISTA	Tycho	Twin
主観測	**64×64 text grid**	主に512×512 PNG	grid / executable evidence	grid
明示的world model	なし	なし、自然言語 notes	**あり、Python**	**あり、Python digital twin**
長期memory	persistent memory	lossless frames + GUIDE/WORKING	persistent evidence/ workspace	code + replay history
hard model verification	なし/非公開	なし	あり	**全action前に replay validation**
supervisor	**あり**	基本single-agent	actor/builder orchestration	validate/explore/ plan loop
planning	LLM direct	free-form language	simulator search	validated simulator search
Public score	**100**	**100**	**100**	93.3
Opus 5 actions	**6,624**	7,542	6,641	—

VISTAは「世界を正確なプログラムとして再構築しなくても、圧縮された自然言語理解で十分な場合がある」という立場を取り、全frameをlossless visual memoryとして保存する。Tycho/Twinは逆に、agentがtest timeにPython simulatorを構築し、過去transitionで検証したうえで将来状態をsimulationする。AVOは前者に近いdirect-interaction型だが、長期memoryとsupervisorをより一般目的のagent infrastructureとして強調する。[34]

このためAVOの研究上の新規性は「100という数値」そのものより、**CUDA kernel optimizationで使った同じlong-horizon control patternがARCにも移ったこと**にある。専用ARC solverを作らずに、異なるfeedback channelへgeneral-purpose architectureを接続した、というtransfer claimがNVIDIAの中心主張である。もっとも、ARC interface自体はVISTAの設計から影響を受けており、完全にtask-agnosticなzero-engineering transferではない。[19]



このタイムラインは、わずか数か月で研究の焦点が「frontier LLMをそのまま行動させる」段階から、memory、compaction、explicit world model、verification、supervisionを組み合わせる**agent systems competition**へ移ったことを示している。 35

実世界への含意、リスク、安全性

ロボティクスと自律システムへの含意

AVOからロボティクスへ直接「100%が移る」と考えるべきではない。ARC-AGI-3は64×64の離散・抽象・turn-based worldであり、知覚ノイズ、連続制御、接触力学、sensor drift、通信遅延、物体遮蔽、不可逆な身体的損傷といったreal-world roboticsの問題を意図的にほぼ除外している。AVO自身もARCではexact text gridを受け取っており、視覚groundingを解いたわけではない。 36

それでも、移植可能性のある設計原理は強い。ロボットやautonomous vehicleにも、「現在の観測だけで毎回推論をリセットしない」「過去の失敗と成功をpersistent stateに保存する」「環境feedbackで仮説を更新する」「local controllerが停滞したときmeta-controllerが戦略を切り替える」という構造はそのまま重要になる。これはNVIDIAがGPU kernelとARCという大きく異なる領域で共通loopとして観測した部分である。 37

ただしreal robotではARCと違って探索actionそのものが危険になり得る。ARCでは「試しに押して何が起きるかを見る」が合理的でも、産業ロボット、車両、医療機器では同じactive explorationを許せない。したがってAVO型システムを物理系へ展開する場合、**探索policyとsafety envelopeを分離し、agentが許可されていないactionを実行できないhard runtime enforcement**が必要になる。

NVIDIA自身もagent stackのsecurityについて、model/harnessだけに規則遵守を期待せず、OpenShellのようなruntimeでagentが「何を見られるか・何に触れられるか・何を実行できるか」を制限する考え方を提示している。AVOのsupervisorは公開説明上、主に**stagnationやunproductive loopを解消するcapability mechanism**であり、独立したsecurity monitorとして説明されているわけではない。この二つは区別する必要がある。 38

Alignment上の含意

AVOは「賢い一回の回答」より、数百・数千stepにわたりgoal pursuitを継続できるシステムを目指している。これは生産性の面では大きな利点だが、alignmentでは**誤った目的・誤った仮説・prompt injection・悪意あるtool outputを長期間保持してしまう**可能性も増やす。persistent memoryは正しい学習を累積させる装置であると同時に、誤ったstateを持続させる装置にもなり得る。

さらに、main agentとsupervisorが同種のfoundation modelや同じ文脈に依存するなら、両者のerrorは独立ではない可能性がある。したがって「LLMにLLMを監督させる」だけでなく、実行権限、ネットワークアクセス、filesystem access、secret access、deployment permissionをruntimeで分離し、重要なactionには deterministic policy checkerやhuman approvalを挟む方が安全設計として妥当である。NVIDIA自身もagentの権限はruntime側でcontain・recordすべきだというstack-level securityの方向性を示している。 39

AnthropicもOpus 5を公開する際、より強い安全保護の対象として扱い、autonomyに関するthreat modelingを行っている。AVOはそのモデルに長期memoryとtool loopを追加するため、**model safetyと agent-system safetyは同じではない**というNVIDIA自身の主張は、能力だけでなくリスクにもそのまま当てはまる。 40

最大の評価上のfailure modes

最も重要なリスクは四つある。

Public-set contamination / benchmark overfitting。 Opus 5のtraining cutoffがPublic Demo公開後であり、学習データ内容が完全公開されていない以上、完全な初見性は保証できない。ARC Prize自身がこの理由を含めpublic scoreをofficial benchmark victoryに使わない。 6

Hidden-compute inflation。 RHAEはLLM token、simulation、Python、subagent callsを数えないため、「environment actionsを減らすために大量の内部computeを使う」システムを高く評価できる。Twinが2.60B tokensを使っていることはこのtrade-offの実例である。 25

False-model persistence。 長期memoryは一度形成した誤ったworld modelを後続levelへ持ち越す可能性がある。Twinが20.1%のactionでworld modelとのprediction mismatchを観測し、counterexampleから修復していることは、世界モデルが常に正しいわけではないことを示す。AVOのdirect reasoningには同等の機械的replay verifierが公開されていない。 41

Metric saturation。 Public Setで複数アーキテクチャが100に到達した時点で、RHAE headline scoreだけではAVO、VISTA、Tychoの優劣を十分に分離できない。今後はprivate generalization、token efficiency、wall-clock latency、cost、variance、robustness、unsafe-action rateなど別軸が必要になる。Tychoも100点同士を区別するためhuman-replay midrankやraw action countsを導入している。 42

未解決問題、研究提案、最終評価

AVOの次に必要なのは、より高いPublic scoreではない。Public Setは既に100で飽和しているため、次の研究価値は**「なぜ100になったのか」「初見環境でも再現するか」「いくらの計算で実現したか」「安全に制御できるか」**を切り分けることにある。 43

第一に、最優先は**blind semi-private / fully-private evaluation**である。AVO、VISTA、Tychoを同じmodel version、reasoning level、token budget、time budget、初見private gamesで比較すれば、harness generalityをかなり厳密に測れる。特にevaluation tasksが各モデルのtraining cutoff後に生成されればcontamination問題を大幅に弱められる。これはARC Prizeがprivate benchmarkを用意した目的と一致する。 44

第二に、AVO自身の**factorial ablation**が必要である。同一Opus 5、同一text-grid interface、同一promptを固定し、`memory off/on × supervisor off/on × compaction off/on × reasoning effort`を比較するべきだ。各条件でRHAEだけでなく、levels completed、environment actions、LLM calls、fresh/cache/output tokens、wall-clock、API cost、retry count、stall count、seed varianceを報告すれば、「memoryが効いた」という仮説を因果的に評価できる。NVIDIAの現在のVISTA比較ではそこまでは分離できない。 23

第三に、**world-model vs direct-reasoning**を同一backendで比較する価値が大きい。Tycho/Twinはexplicit executable modelを、AVO/VISTAはより柔軟なimplicit/free-form modelを支持する。この二者は、精密検証と計算コストのtrade-offを持つ。Twinはhard replay validationが長いゲームで特に有効と報告する一方、AVOは明示的world modelなしでさらに少ないenvironment actionsを実現している。両方を同一model、同一compute budgetで比較すれば、「いつコード化world modelが必要か」を明らかにできる。 [45]

第四に、benchmark側は**compute-aware score**を併記すべきである。例えばRHAEに加え、

$$\text{Utility} = \frac{\text{task completion} \;/\; \text{action efficiency}}{\alpha\,\text{tokens} + \beta\,\text{wall time} + \gamma\,\text{dollar cost}}$$

のように環境効率と内部計算を別々に公開すれば、1 actionあたり数十万token考えるシステムと、低computeで多少多くactionするシステムを区別できる。Twinの計測は、この必要性を具体的に示している。 [25]

第五に、roboticsやautonomous systemsへ進む前には、**noisy / partial / continuous environments**での検証が必要である。text gridをcamera/video、uncertain state estimation、continuous controlへ置き換え、さらに危険な探索をsandbox/simulator内に閉じ込める必要がある。実世界deploymentではsupervisorとは別に、runtime permission、least privilege、audit log、human checkpoint、rollback、network isolationが必要になる。NVIDIAのOpenShellを含むtrusted-agent-stackの方向性は、この層をAVOのcapability layerから分離する考え方として重要である。 [46]

最終判断

研究的重要度：高い。AGI証拠としての強度：限定的。agent-systems engineeringの証拠としての強度：かなり高い。

AVOの100%が本当に示したのは、「Claude Opus 5が突然100%の抽象知能を獲得した」ことではない。むしろ、**30%級に見えるfoundation modelでも、memory、context、tool use、supervision、feedback loopを適切に設計すると、同じ公開タスク群に対するeffective agent capabilityが劇的に変わる**ということである。これはOpenAIの13.3→38.3、Twinの61.1→93.3、VISTA/Tychoの100という独立した結果とも方向が一致する。 [47]

その意味で、AVOは「次のスケーリング軸」がparameter countだけではなく、**test-time state、memory、verification、supervision、tooling、meta-controlへ移っている**ことを象徴する結果である。一方で、AVOのPublic 100を「ARC-AGI-3 solved」と呼ぶにはprivate evaluationが欠け、Opus 5のtraining cutoffはPublic Set公開後であり、ARC版のcompute accountingとcomponent ablationも欠けている。したがって現段階で最も厳密な表現は、**「AVOは公開済みARC-AGI-3環境を極めて高いaction efficiencyで完全攻略し、長期エージェントのシステム設計がfoundation-model単体性能と同程度かそれ以上に重要になり得ることを示した。ただし未知環境への一般知能を証明したわけではない」**である。 [48]

主要一次資料とURL

資料	URL
NVIDIA Technical Blog — AVO ARC-AGI-3	https://developer.nvidia.com/blog/nvidia-avo-reaches-100-on-arc-agi-3-demonstrating-a-frontier-level-general-purpose-architecture-for-long-horizon-autonomous-agents/

資料	URL
AVO論文 — Agentic Variation Operators for Autonomous Evolutionary Search	https://arxiv.org/abs/2603.24517
ARC-AGI-3 Technical Report	https://arcprize.org/media/ARC_AGI_3_Technical_Report.pdf
ARC-AGI-3 benchmark	https://arcprize.org/arc-agi/3/
ARC-AGI-3 scoring methodology	https://docs.arcprize.org/
ARC Prize — Claude Opus 5 results	https://arcprize.org/results/anthropic-claude-opus-5
VISTA project	https://vista-research.github.io/
Tycho論文	https://arxiv.org/abs/2607.28287
Tycho code	https://github.com/NIMI-research/Tycho
Twin論文	https://arxiv.org/abs/2608.14490
OpenAI — ARC-AGI-3 harness investigation	https://openai.com/index/how-two-settings-tripled-our-arc-agi-3-scores/
Anthropic Transparency Hub — Opus 5 training/cutoff	https://www.anthropic.com/transparency
Anthropic Claude pricing	https://docs.anthropic.com/en/docs/about-claude/pricing
NVIDIA — AI agent stack security	https://developer.nvidia.com/blog/where-security-fits-in-an-ai-agent-stack/
NVIDIA OpenShell / autonomous-agent security	https://developer.nvidia.com/blog/run-autonomous-self-evolving-agents-more-safely-with-nvidia-openshell/

1 13 16 17 19 20 23 37 43 48 NVIDIA AVO Reaches 100% on ARC-AGI-3, Demonstrating a Frontier-Level General-Purpose Architecture for Long-Horizon Autonomous Agents | NVIDIA Technical Blog
https://developer.nvidia.com/blog/nvidia-avo-reaches-100-on-arc-agi-3-demonstrating-a-frontier-level-general-purpose-architecture-for-long-horizon-autonomous-agents/

2 44 arcprize.org
https://arcprize.org/media/ARC_AGI_3_Technical_Report.pdf

3 10 34 VISTA: A Visual Harness for Reasoning in an Interactive World
https://vista-research.github.io/

4 30 32 47 Claude Opus 5 - ARC-AGI Results
https://arcprize.org/results/anthropic-claude-opus-5

5 9 ARC-AGI-3 Scoring Methodology - ARC-AGI-3 Docs
https://docs.arcprize.org/methodology

6 15 40 Anthropic’s Transparency Hub \ Anthropic
https://www.anthropic.com/transparency

7 11 ARC-AGI-3
https://arcprize.org/arc-agi/3

8 36 ARC-AGI-3: A New Challenge for Frontier Agentic Intelligence
https://arxiv.org/html/2603.24621v1

12 14 18 22 28 AVO: Agentic Variation Operators for Autonomous Evolutionary Search
https://arxiv.org/html/2603.24517v1

21 27 29 42 Tycho: Active Abstraction with Programmatic World Models for ARC-AGI-3
https://arxiv.org/html/2607.28287

24 25 41 45 Twin: Playing an Unknown Game with a Test-Time Digital Twin
https://arxiv.org/html/2608.14490v1

26 Pricing - Claude Platform Docs
https://docs.anthropic.com/en/docs/about-claude/pricing?utm_source=chatgpt.com

31 35 ARC-AGI-3: A New Challenge for Frontier Agentic ...
https://arcprize.org/media/ARC_AGI_3_Technical_Report.pdf?utm_source=chatgpt.com

33 How enabling two settings tripled our scores on the ARC-AGI-3 benchmark | OpenAI
https://openai.com/index/how-two-settings-tripled-our-arc-agi-3-scores/

38 39 46 Where Security Fits in an AI Agent Stack | NVIDIA Technical Blog
https://developer.nvidia.com/blog/where-security-fits-in-an-ai-agent-stack/?utm_source=chatgpt.com