DeepAgent: 自律的エージェントの次なる飛躍

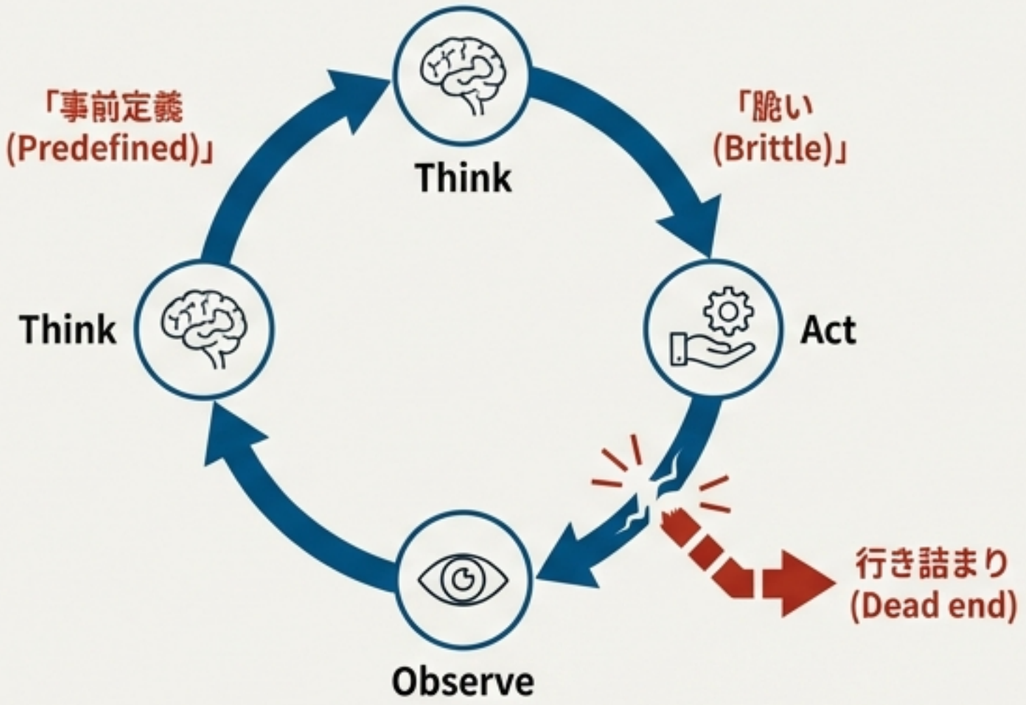
硬直したワークフローとコンテキストの爆発を超えて

RUC-NLPIR
中国人民大学 高瓴人工智能学院 自然语言处理研究所

小红书

現在のエージェントが直面する「二つの壁」

硬直したワークフロー



従来の手法は「思考→行動→観察」という固定的なループを強制する。予測不可能な実世界タスクでは、この硬直性が探索の行き詰まり（Dead end）を引き起こす。

コンテキスト長の爆発

長期タスクでは履歴が全て生のテキストとして蓄積され、3つの致命的な問題が発生する：

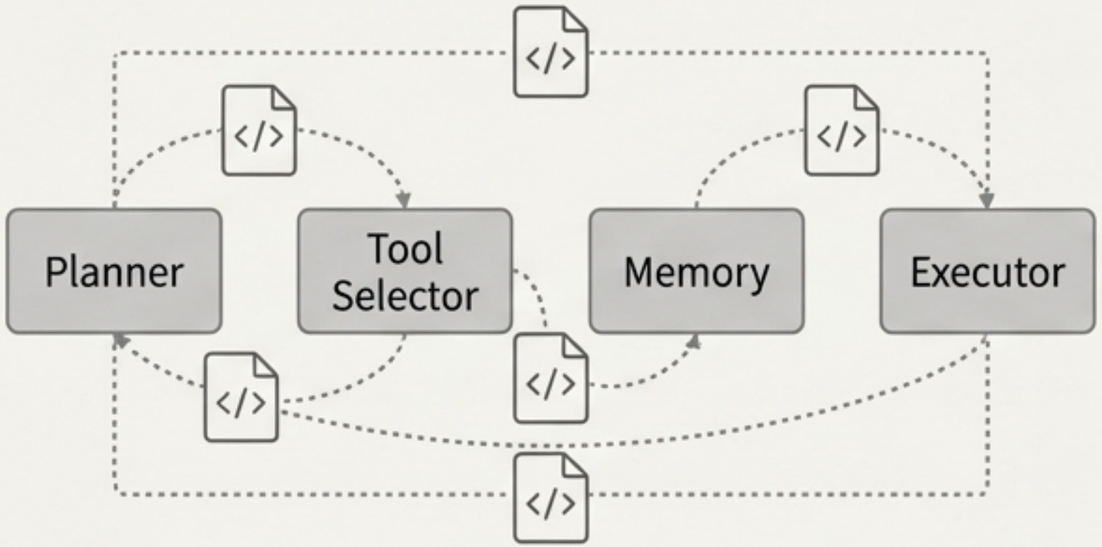
1. **情報の埋没**: 初期の重要情報が膨大なログに埋もれ、注意機構が機能不全に陥る。
2. **コストとレイテンシの増大**: 計算コストが指数関数的に増加する。
3. **推論精度の低下**: 無関係なログがノイズとなり、幻覚（Hallucination）を引き起こす。

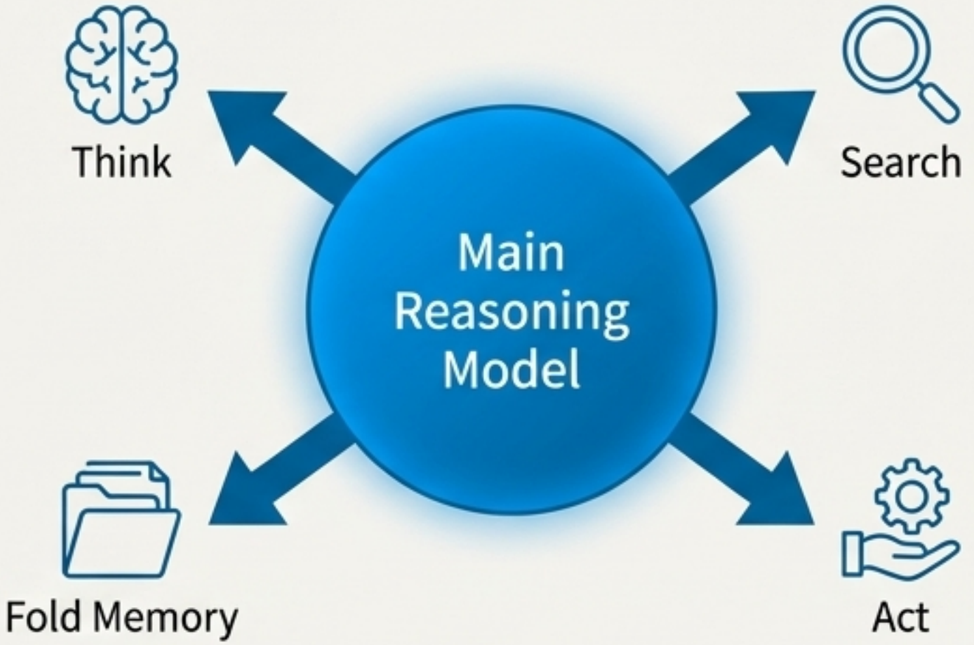
パラダイムシフト：外部制御から「統合的エージェント推論」へ

従来型アーキテクチャ

Before vs. After

DeepAgent



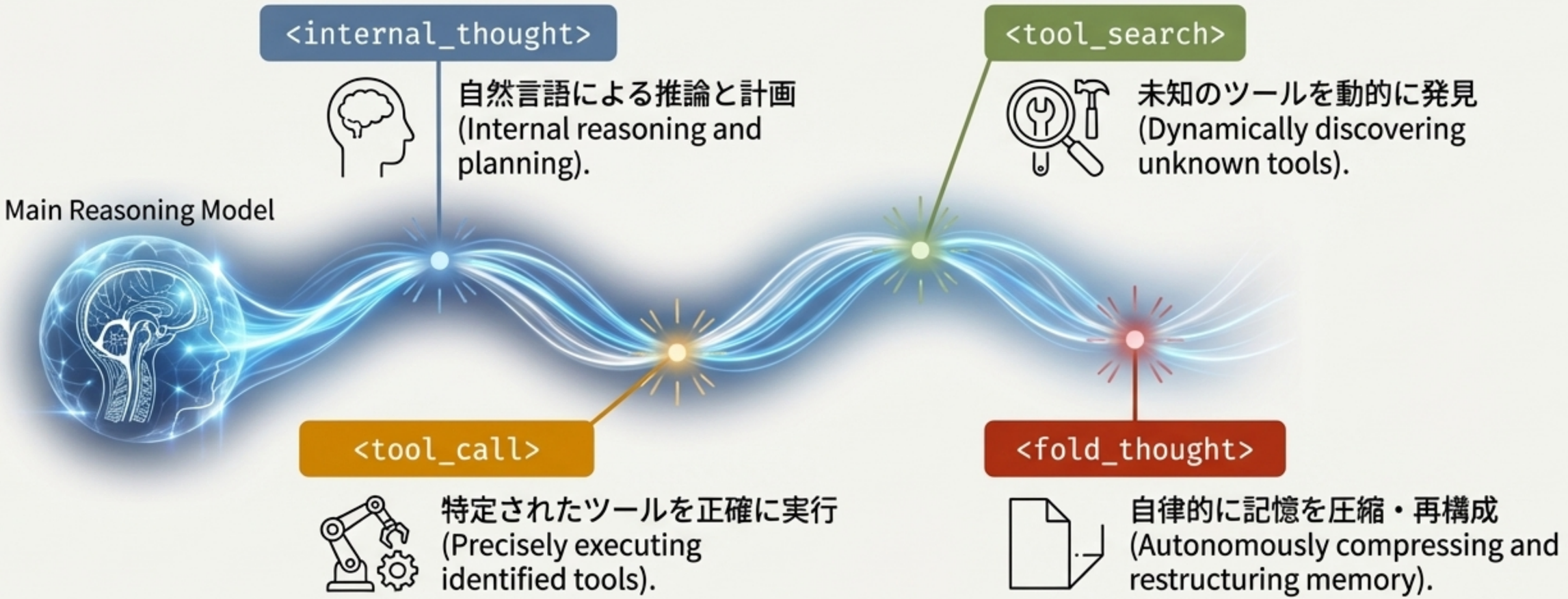


「モジュール型・外部制御 (Modular / Externally Controlled) 」

「統合的エージェント推論 (Unified Agentic Reasoning) 」

DeepAgentは、ツール使用やメモリ管理を外部スクリプトに委ねるのではなく、
LLMの推論プロセスそのものに内包させる。

DeepAgentアーキテクチャ：思考の流れにすべてを統合



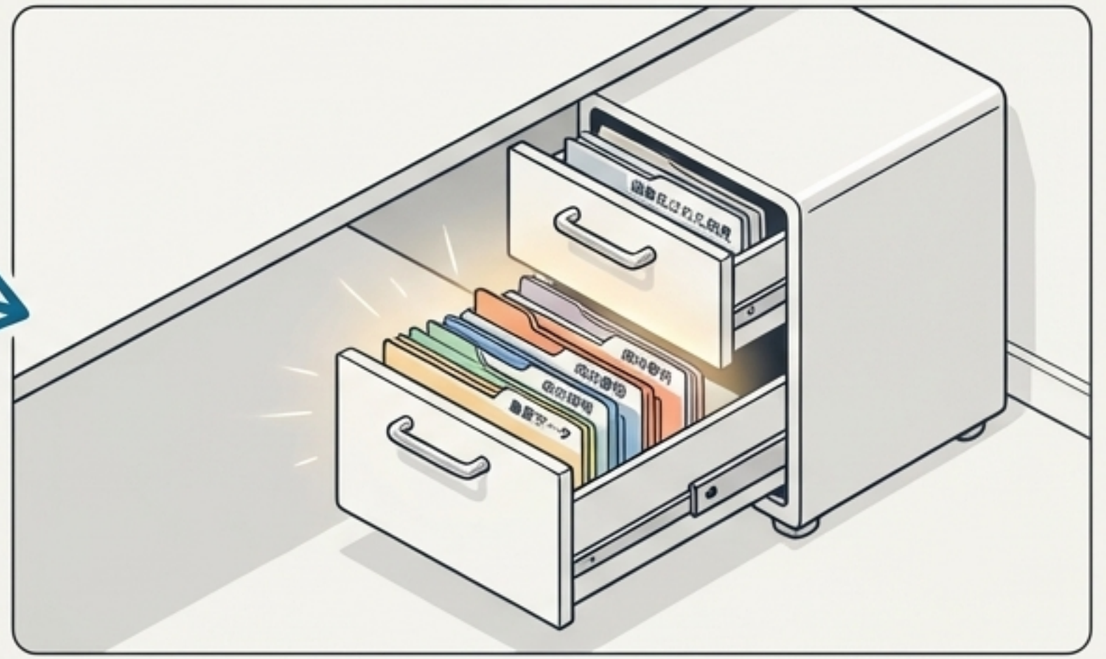
この4つのアクションタイプを文脈に応じて動的に生成することで、人間のような柔軟な認知戦略を実現する。

中核技術：自律的メモリ畳み込み — 「構造化された忘却」の実現

これは単なるデータ圧縮ではなく、人間の認知プロセスに触発されたメカニズムである。

Unmanaged Context
管理されていないコンテキスト

Folded Memory
畳み込まれたメモリ



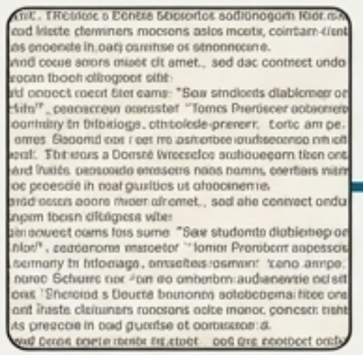
人間がワーキングメモリ（短期記憶）の情報を整理し、長期記憶に格納するプロセスを工学的に模倣。不要な詳細（失敗した試行など）を意図的に忘却し、抽象化された知見を保持する。

トリガー条件

- エージェントは「コンテキストの飽和」「サブタスクの完了」「探索の行き詰まり」を検知すると、自律的に **<fold_thought>** トークンを生成する。

脳模倣型メモリ・スキーマ：3層構造による高密度な記憶

自由形式の要約ではなく、厳格に定義されたJSON構造で記憶を保存することで、情報の欠落や幻覚を防ぎ、解釈の曖昧性を排除する。



`s_t
(Raw Interaction History)

Auxiliary LLM

{

エピソード記憶（Episodic Memory）

タスク全体のハイレベルな進行状況（完了したサブタスク、重要な意思決定）を記録する長期記憶。「なぜその判断をしたか」という因果関係を保持する。

ワーキングメモリ（Working Memory）

現在の焦点（直近のサブゴール、保留中の疑問点）を維持する短期記憶。「今、何をしようとしていたか」を保持する。

ツール記憶（Tool Memory）

ツール使用に関する経験的知識（有効だった引数、発生したエラー）を蓄積。「このAPIはYYYY-MM-DD形式を要求する」といった具体的な知見を記録し、同じ失敗を防ぐ。

}

スケーラブルなツールセット：未知の道具を発見し、使いこなす

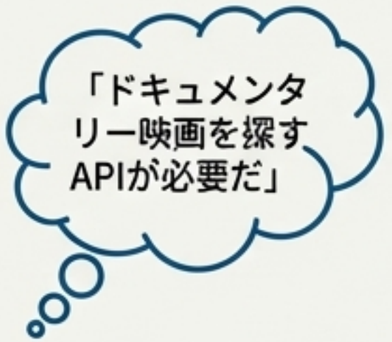
従来のエージェント: クローズドセット / 数十のツール

DeepAgent: オープンセット / 16,000+ API

従来の研究ではツールが数十個に限定されていたが、現実のウェブ環境には数万のAPIが存在する。DeepAgentは、事前に定義されていない未知の道具を自ら発見し、使用法を学習する「オープンセット」環境で機能するよう設計されている。

動的ツール検索：「思考」が「検索」になるメカニズム

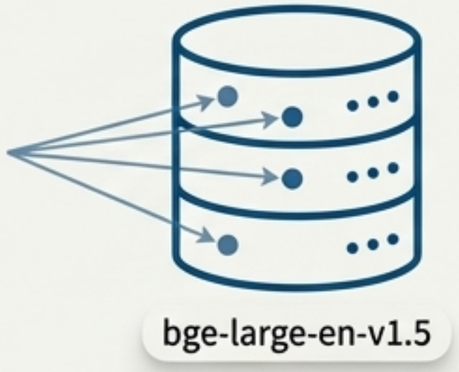
1 クエリ生成 (Generate Query)



<tool_search>
「ドキュメンタリー映画を探すAPIが必要だ」

モデルが考え、<tool_search>トークンに続いて自然言語の検索クエリを生成する。

2 高密度検索 (Dense Retrieval)



生成されたクエリをベクトル化し、ツールAPIのドキュメントが格納されたベクトルデータベースに対してコサイン類似度検索を実行する。

3 コンテキストへの動的注入 (Inject Context)

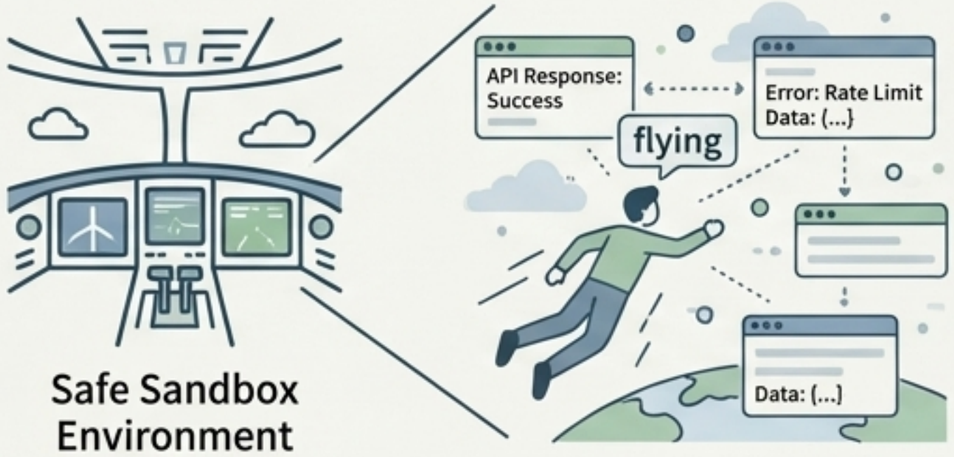
関連性が最も高いトップK個のツール定義（APIドキュメント）を即座にLLMのコンテキストに挿入。エージェントは次の瞬間からそのツールを使用可能になる。

新学習手法「ToolPO」：安全な環境で、賢く学ぶ

従来のRL手法は、報酬のスパース性（何が成功に繋がったか不明）と、実API利用時のコスト・不安定さという課題を抱えていた。

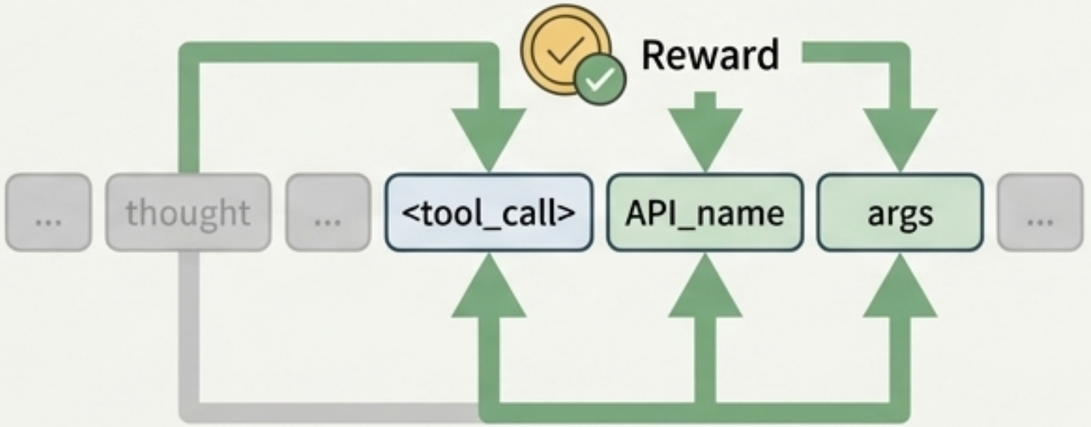
ToolPO’s Two Innovations

1. LLMベースのツールシミュレータ



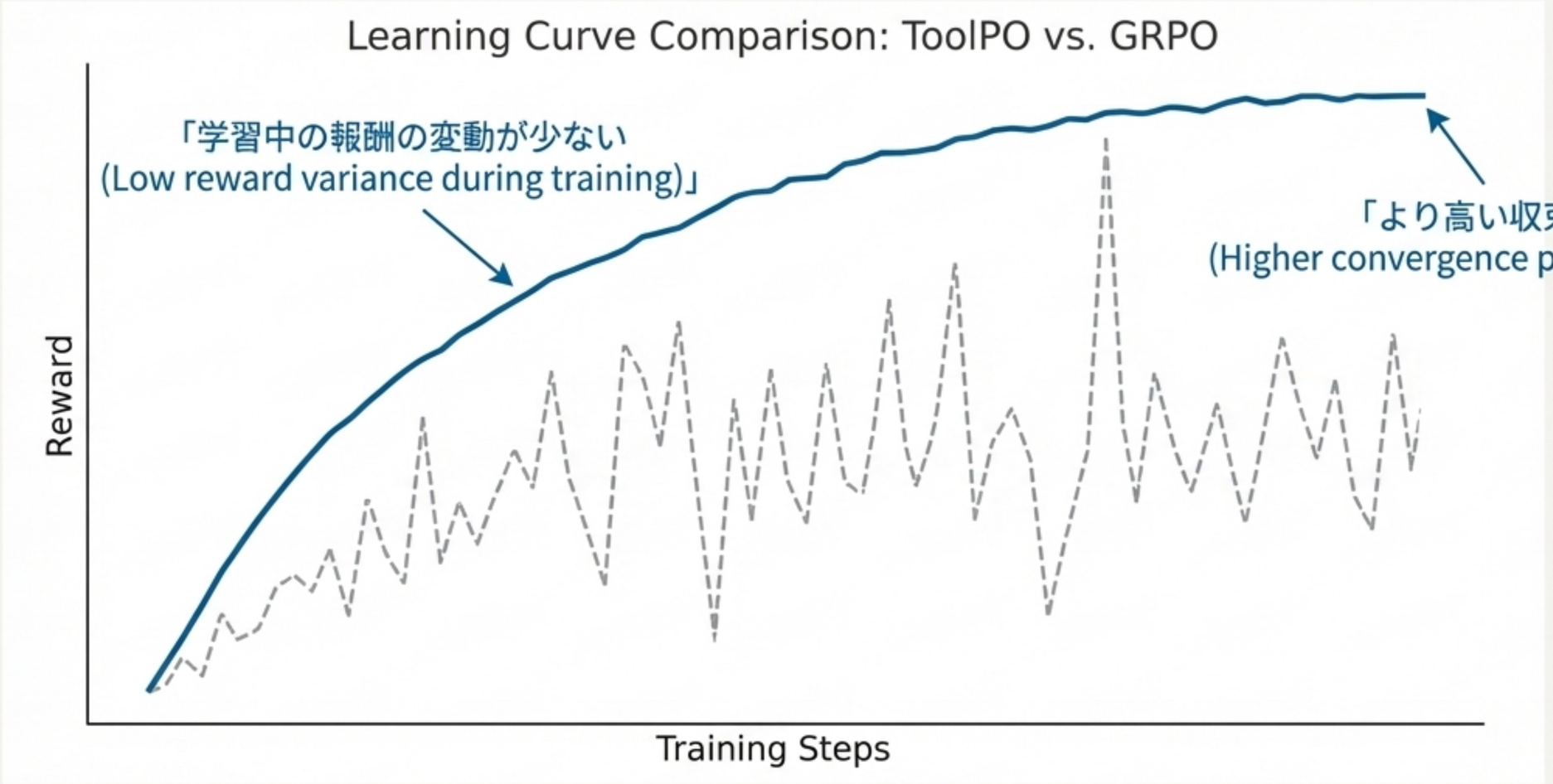
実際のAPIを叩く代わりに、補助LLMがAPIの応答（成功データやエラーメッセージ）を予測して返す。これにより、コストや不安定さを排除し、安全な環境で大規模な試行錯誤が可能になる。

2. 微細なアドバンテージ帰属



行動系列全体ではなく、「ツール呼び出し」に関連する特定のトークン群に直接報酬をフィードバック。これにより、「この場面での検索ツールの使い方は正しかった」といった部分的な成功を効率的に学習できる。

ToolPOの効果：安定かつ効率的なポリシー最適化



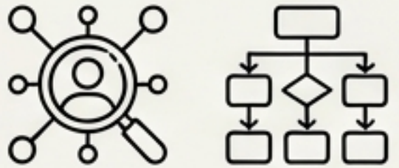
実験結果は、ToolPOがGRPO等の従来手法と比較して、学習の安定性と最終的な性能の両方で優れていることを示している。

ベンチマーク評価①：汎用ツール使用タスクで他を圧倒

表1：一般ツール使用タスクにおけるPass@1成功率（%）の比較

手法	モデル	ToolBench	ToolHop	TMDB	Spotify
DeepAgent-32B-RL	**QwQ-32B**	**64.0**	**40.6**	**75.4**	**92.0**
DeepAgent-32B-Base	QwQ-32B	63.0	49.1	70.2	89.3
ReAct	QwQ-32B	55.0	36.2*	22.8	45.5
CodeAct	QwQ-32B	51.0	29.0	24.6	49.6
Plan-and-Solve	QwQ-32B	54.0	-	19.6	18.0

Key Analysis Points



- **オープンセットでの優位性**：数千のツールから選択するToolBench/ToolHopでReAct等を大きく凌駕。動的ツール検索の有効性を証明。
- **複雑なクエリへの対応**：複数APIの組み合わせが必要なSpotifyタスクで92.0%の成功率を達成、ReAct (45.5%) を圧倒。

ベンチマーク評価②：長期的・複雑な実世界タスクも制覇

表2：ダウンストリーム・アプリケーションにおける成功率比較

手法	GAIA	ALFWorld	WebShop
DeepAgent-32B-RL	**53.3**	**91.8**	**34.4**
DeepAgent-32B-Base	46.7	88.1	32.0
HiRA (Previous SOTA)	42.5	84.3	-
CodeAct	34.5	-	18.0

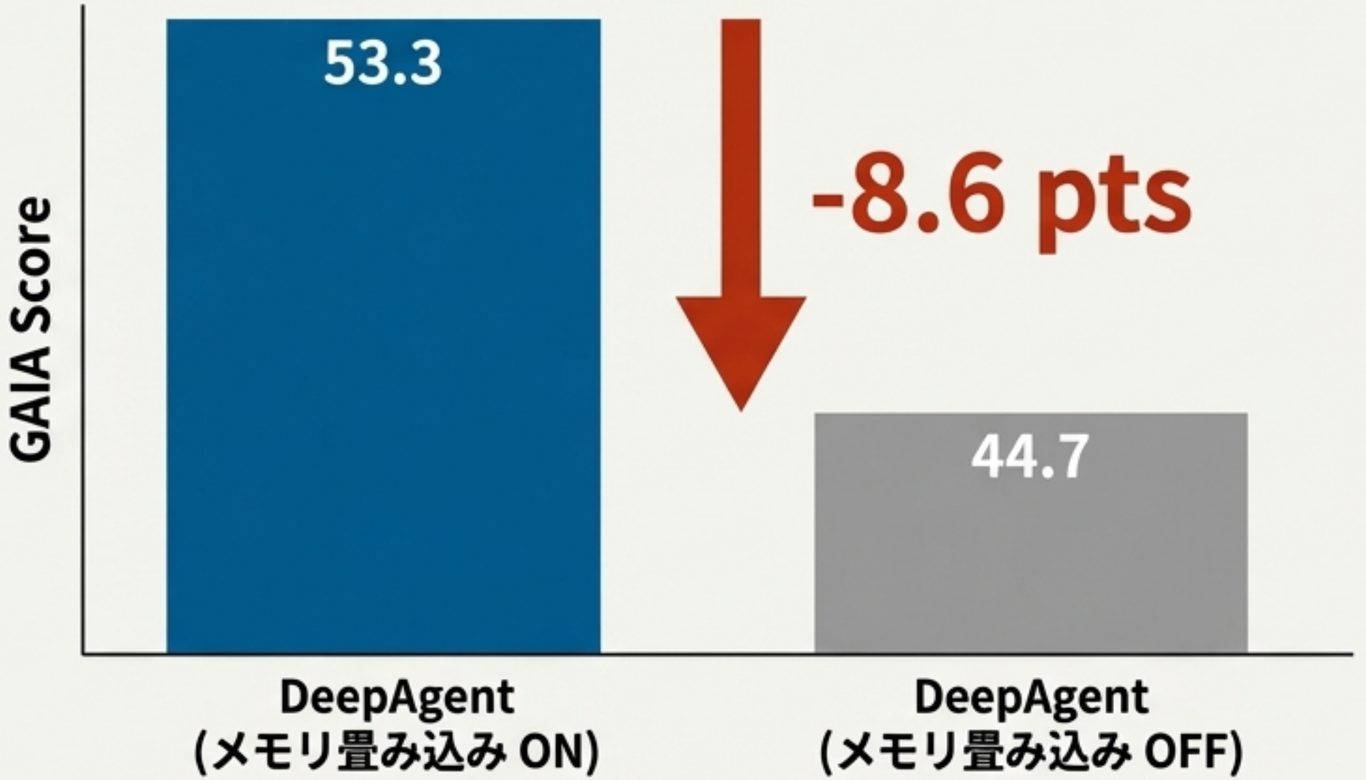
NEW SOTA

約2倍の性能
(Nearly 2x Performance)

GAIA:
汎用AIアシスタントの能力を測る難関ベンチマークで、既存のSOTA (HiRA: 42.5) を大幅に更新。

WebShop:
多段階のオンラインショッピングタスクにおいて、メモリ管理機能が長いセッションでの文脈維持に貢献。

成功の鍵：メモリ畳み込みがもたらす決定的差

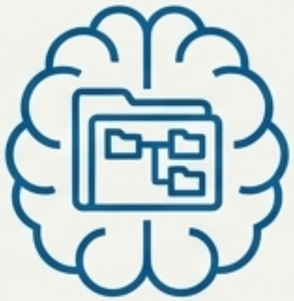


メモリ畳み込み機構を無効化するだけで、GAIAのスコアは53.3から44.7へ急落。
この結果は、長期的なタスクにおいて、情報を構造化して保持する能力が、
単なるツール使用能力以上に決定的であることを証明している。

本研究の意義：「システム2」思考への道筋を示す

統合的推論 (Unified Reasoning)

外部スクリプトに依存しない、新しいエージェントアーキテクチャの標準を提示。



構造化された記憶 (Structured Memory)

コンテキスト長の爆発問題に対する、スケーラブルで実用的な解決策。

スケーラブルなツール使用 (Scalable Tool Use)

16,000以上のAPIに対応し、実世界での応用可能性を解放。

現在のLLMが直感的な応答（System 1）に優れる一方、DeepAgentは熟考し、計画し、自己修正する能力（System 2）を工学的に実装する一つの解を示した。

産学連携の成果: RUC-NLPIRの理論的エレガンスと、Xiaohongshuの実用的スケーラビリティが融合したことで、このブレークスルーが実現した。