ChatGPT

DeepAgent: スケーラブルなツールセットを備えた汎用推論エージェント

1. 技術的詳細（アーキテクチャと推論プロセス）

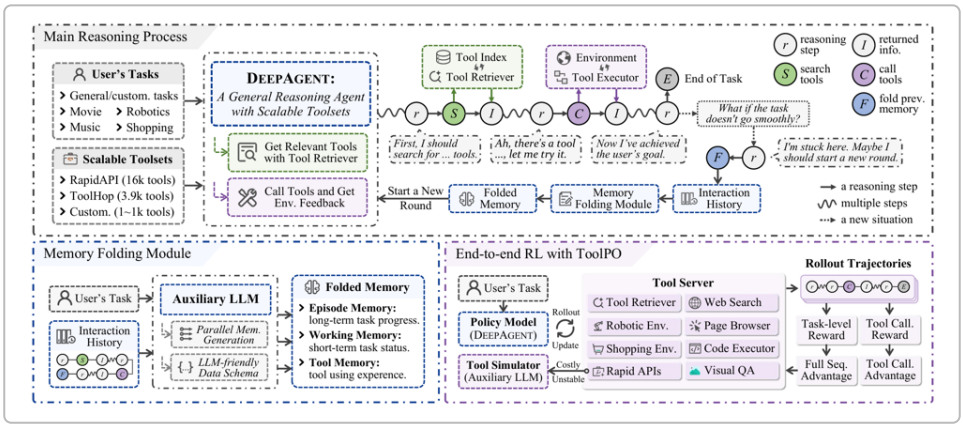
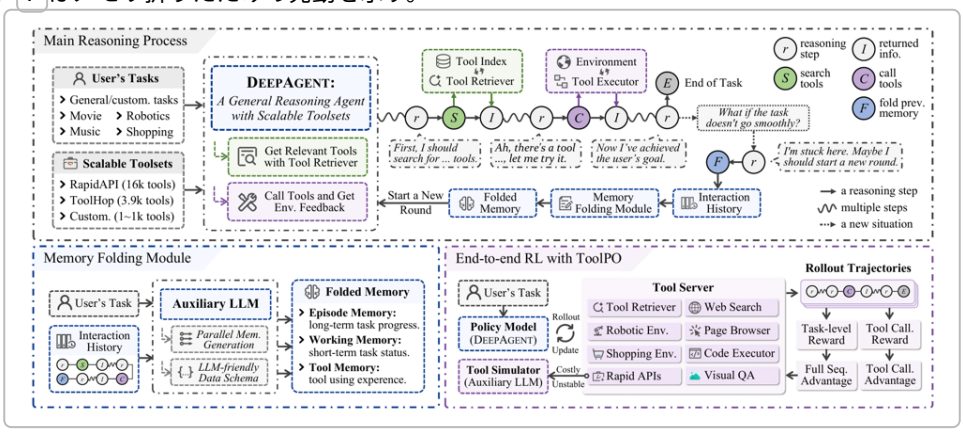


図1:　DeepAgentフレームワークの概要。メインの推論モデル（中央上）はユーザ課題に対して連続的に思考し、必要に応じてツールを検索・実行し、場合によってはメモリ折りたたみ（Memory　Folding）を行う。一連の推論プロセス内でツール発見とアクション実行、メモリ圧縮が統合されている [1] [2]。左下はメモリ折りたたみモジュールで、Auxiliary　LLM（補助モデル）がインタラクション履歴を解析し、エピソード記憶・作業記憶・ツール記憶の3種類の構造化メモリを生成する [3]。右下はToolPOによるエンドツーエンド強化学習の構成で、Tool Simulator（補助モデル）でツール応答を模擬しつつ、タスク成功とツールコールの両方に報酬を与えて学習する [4]。図中 r は推論ステップ、 s はツール検索、 c はツール呼び出し、 i は結果の取り込み、 F はメモリ折りたたみの発動を示す。



DeepAgentは、大規模言語モデル（LLM）を中核とした汎用エージェントであり、**「思考・ツール発見・行動実行」**を単一の統一プロセスの中で行う設計が特徴です [5]。従来のReActのような固定的な「推論→行動→観察」のループや、あらかじめ決められたツールセットに頼る手法とは異なり、DeepAgentは**タスク遂行中に必要な外部ツールを動的に探索・取得**し、適宜呼び出すことができます [6] [7]。この**動的ツール検索**はエージェントの思考プロセスに組み込まれており、タスクに関連するAPIやツールを**数万規模の候補からオンデマンドで取得**します [8] [7]。例えばDeepAgentは16,000以上のAPI（RapidAPI経由）や数千規模のカスタ

ムツールから必要なものを探せる設計であり、OpenAIの従来エージェントが検索・コード・閲覧程度の限られたツールに依存するのとは対照的です [8] [7]。ツールの検索要求はLLMの思考中に特殊トークン（例：`<tool_search>クエリ</tool_search>`）として生成され [9]、埋め込みベースの検索器が**ツール説明のベクトル索引**から関連する上位候補を返します [10] [11]。その後、エージェントは選択したツールをJSON形式で呼び出します（例：`<tool_call>{"name":ツール名,"arguments":{...}}</tool_call>`） [12]。呼び出されたツール（環境）が返す結果は`<tool_call_result>`タグ付きでLLMに入力され、次の推論に活用されます [12]。

アーキテクチャ上、DeepAgentは**メインの推論モデル**（Large Reasoning Model, 例：QwQ-32BやQwen3-30Bなど数十億パラメータ級）と、**Auxiliary LLM（補助モデル）**を組み合わせて動作します [13] [14]。メインモデルはタスク全体の戦略的思考と意思決定を担い、補助モデルは裏方として**ツール情報の要約**や**履歴の整理**などを担当します [15] [16]。この役割分担により、メインモデルは高次の推論に集中し、補助モデルが長大なドキュメントや実行結果を要約して**ノイズを低減**することで、文脈をクリーンに保ちます [17] [18]。特にツール候補のドキュメントが長過ぎる場合、Auxiliary LLMが要点を抽出してメインモデルに渡し、ツール実行結果が冗長な場合も重要部分を抽出してフィードバックする仕組みです [19]。これに加え、DeepAgentの**記憶管理**にも補助モデルが深く関与しています。

メモリ折りたたみ (Memory Folding) はDeepAgentの中核的な機構であり、長い対話履歴によるコンテキスト爆発と誤情報の蓄積を防ぐ目的で導入されています [20] [21]。エージェントが多数のステップを経て一連の思考やツール呼び出しを行うと、コンテキスト長が膨れ上がり、過去の誤った探索に引きずられる「デススパイラル」に陥るリスクがあります [22]。DeepAgentは必要に応じて思考の途中で`<fold_thought>`トークンを出力し、**直近の相互作用履歴全体を構造化メモリに圧縮**します [21] [23]。この処理は人間が一旦立ち止まり深呼吸して状況を整理し戦略を練り直すような動作に相当し、**「一息ついて考え直す」**能力をエージェントに与えます [21] [3]。圧縮後、エージェントは長大な履歴の代わりに**要約メモリ**を保持して推論を続行し、過去の重要事項を維持しつつ不要な詳細を削除できます [24]。メモリは**JSONスキーマ**に従った形式で記録されるため、LLMが再読み込みする際に誤解析や情報欠落が起こりにくく安定です [25] [26]。DeepAgentの記憶は以下の3種類に分かれています [27] [26]：

- **エピソード記憶 (Episodic Memory)**：タスク遂行における主要な決定やサブタスク完了など**長期的な進捗ログ**です [28]。過去の重要イベントや最終目標に対する要約が含まれ、現在までの戦略を俯瞰する役割を果たします [29] [30]。
- **作業記憶 (Working Memory)**：現在のサブゴールや直近の課題、進行中の計画など**短期的かつ即応的な情報**を保持します [31]。メモリ折りたたみ前後でタスクを継続するための繋ぎ目として機能し、折りたたみ直前の状態を忘れずに再開できるようにします [29] [32]。
- **ツール記憶 (Tool Memory)**：これまで試行したツールの使用履歴、成功・失敗パターン、パラメータ設定やエラー傾向など**ツール利用に関する知見**を蓄積します [33]。どのツールが有効だったか、一般的な使い方のコツや注意点などを学習し、今後の戦略改善に活かします [29] [30]。

メモリ折りたたみが行われると、Auxiliary LLMが現在までの全履歴を分析し上記3メモリを生成、元の履歴を置き換えます [23] [34]。こうしてエージェントは過去の失敗を踏まえつつ新たな方針で再出発でき、複雑なタスクでも途中で戦略転換して成功率を高められます [35] [36]。実際、この**「立て直し」**により長いシーケンスのタスク成功率が大幅に向上したと報告されています [21] [37]。

ToolPO (Tool Policy Optimization) はDeepAgent専用に開発された**エンドツーエンド強化学習アルゴリズム**です [38] [4]。一般に多数の外部APIを含むエージェントの学習には、不安定なAPI応答や高コスト・低効率といった課題があります [39]。DeepAgentはこれを克服するため、**LLMベースのツールシミュレータ**を導入し、実世界のAPIを模倣した環境下でエージェントを訓練します [39] [40]。具体的には、補助モデル（例：Qwen2.5-32B-Instruct）が各ツールAPIの説明と引数から**現実的なレスポンスを生成**し [41]、エージェント（ポリシーモデル）はその出力を観察して次の行動を学習します。これによりAPIの不安定さやレート制限を気にせず**高速かつ低コスト**に大量のエピソードを収集できます [42] [43]。ToolPOでは**報酬設計**にも工夫があ

り、最終タスク成功に対する報酬（タスク完遂か否か）に加えて**中間のツール呼び出しの正確さ**にも報酬が与えられます [44] [45]。例えば正しいツールを呼び出せた場合や無駄な操作を省いた場合にもプラスの報酬を加算し、エージェントが**道中の行動品質**も最適化するよう促します [44] [46]。さらに**アドバンテージ割当 (Advantage Attribution)** の概念を導入し、**ツール呼び出しやメモリ折りたたみに関与する特定トークン**に対してのみ局所的な価値情報を割り当てることで、どの生成トークンが成功に寄与したかを細かく学習させます [47] [48]。通常のエージェント強化学習では最終結果に対する一括の勾配しか得られずクレジット割当が粗いのに対し、ToolPOでは**「このツール呼び出しトークンが正解だった」**といった微粒度のフィードバックが行われるため、学習が格段に安定かつ高速になります [42] [43]。実際、従来法（例えば標準的なGRPO）と比較して**報酬の変動が小さく安定し、より高い最終報酬水準に到達**したことが確認されています [49]。このようにDeepAgentは、巨大なツール空間を扱うための設計と学習戦略を組み合わせることで、ツール使用能力と長期的な推論安定性を両立しています。

2. 実験結果（ベンチマーク評価と性能分析）

DeepAgentの有効性は、**8つのベンチマーク**で包括的に検証されています [50] [51]。評価は大きく二種類に分かれ、(1) **一般ツール使用タスク**5種（ToolBench, API-Bank, TMDB, Spotify, ToolHop）と、(2) **下流アプリケーション**4種（ALFWorld, WebShop, GAIA, HLE）です [50] [51]。前者は大小さまざまなツールセット（数十から1万6千超まで）を扱う問題で、後者は特定ドメイン環境における長尺タスク（例えば家庭内作業やウェブ上の買い物、マルチモーダルな情報探索、人類最後の難問集など）からなります [52] [53]。評価指標には各タスクの成功率や得点（Pass@1や成功率%など）が用いられ、DeepAgent（32Bモデル版）と各種ベースラインを比較しています [54] [55]。主要なベースライン手法は、**ReAct** [56] や**Plan-and-Solve** [56] といった従来型エージェントに加え、最新の強力な言語モデルを用いた**CodeAct** [57]、**WebThinker** [58]、階層型手法の**HiRA** [59]、自己反省を行う**Reflexion**など多岐にわたります [60] [59]。また一部には**GPT-4を用いたReAct (GPT-4o)**など強力なモデルを用いた上限性能の参考値も含まれています [61] [62]。

総合性能として、DeepAgentは**全タスクで既存手法を大きく上回る**結果を示しました [51] [63]。特に**エンドツーエンドの統合型推論**がもたらす恩恵が顕著で、Rigidなワークフローに縛られないDeepAgentは**柔軟なツール利用と一貫した長期推論**によって高い成功率を達成しています [64]。たとえば**映画データベースAPI (TMDB)**タスクでは、DeepAgent-32B（強化学習適用版）が**89.0%**という高い成功率を記録し、同じ32Bクラスの最良ベースライン（Plan-and-SolveやCodeActなど）の**55.0%**を大きく引き離しました [65]。**音楽ストリーミングAPI（Spotify)**でもDeepAgentは**75.4%**を達成し、ベースライン最高の**52.6%**より約23ポイント高い成績でした [65]。このように**固定手順に頼らない包括的な推論**が、既存手法に対して大幅な性能向上をもたらすことが示されています [64] [7]。

さらにDeepAgentの強みは、**未知ツールを含むオープンセット環境**でより一層発揮されました [66]。ツールを事前に指定せず必要に応じ探索する設定（オープンセット）では、従来手法はツール候補が増えるにつれ性能が急激に低下します。一方、DeepAgentは動的ツール発見戦略によりスケーラブルに対処し、高い堅牢性を示しました [66] [37]。例えば**16,000ものAPIが使えるToolBench**タスクでDeepAgent-32Bは**64.0%**の成功率を収め、ベースライン最高の**54.0%**を大きく上回りました [67] [37]。**3,912ツールのToolHop**でもDeepAgentは**40.6%**に達し、ベースライン（29.0%）より約11ポイント高くなっています [67]。この差は、必要なツールをその都度見つけ出すDeepAgentのアプローチが、事前に限られたツールだけを与えられた従来法よりも**現実的な設定で有利**であることを示します [66] [37]。実際、同一手法における比較でも「ツール検索を逐次行うオープンセット方式」が「最初から必要ツールだけ与える方式」を全てのフレームワークで上回っており、**動的ツール検索の優位性**が定量的に確認されています [68] [69]。これは大規模ツールセットを扱う現実タスクでは、**その場で必要な道具を選ぶ**ことが重要である証拠と言えます。

強化学習 (ToolPO)の効果も明確に現れています。強化学習による訓練を施したDeepAgent-32B-RLは、未訓練のベース版と比べ全体的に**5〜6%前後の性能向上**を示しました [70] [71]。例えばToolBench（オープンセット）で成功率が**約6.0ポイント**上昇し、Spotify（ツール固定）でも**5.2ポイント**改善しています [37] [72]。また下記に述べる複雑環境タスクでも、強化学習により**+3〜6ポイント**の上乗せ効果が得られています [73]

[74]。これはToolPOによる**ツール使用の熟達**と**長期戦略の強化**が奏功した結果であり、提案手法の有効性を裏付けています [37] [74]。アブレーション研究でも、ToolPOを除去した場合に**平均スコアが約4ポイント低下**し最も性能悪化が大きかったことが報告されており、エンドツーエンドRLが中核技術であることが示唆されています [75] [76]。

次に**下流アプリケーション**（複雑タスク）での結果です。これらはエージェントに**長いアクションシーケンス**と**マルチモーダル**あるいは**複合的スキル**を要求する難易度の高い評価環境です。DeepAgentはこれらのタスクにおいても**顕著な性能優位**を示しました [77] [78]。例えば、ウェブ検索・計算・画像理解などが混在する**GAIA**では、DeepAgent-32B（RLなし）でもスコア**46.7**と高く、従来型トップのCodeAct（34.5）を大きく超えました [79]。さらに強化学習版は**53.3**まで向上し、競合のWebThinker (37.0) やHiRA (42.5)にも大差を付けています [62]。オンラインショッピング環境**WebShop**でもDeepAgent-32Bは成功率**32.0%**と、従来CodeActの**18.0%**から倍近い改善を達成しました [79]。家庭内テキスト操作タスク**ALFWorld**ではDeepAgent-32B-RLが**91.8%**という高成功率を示し、階層型手法HiRAの**84.3%**を上回っています [78] [80]。特筆すべきは、**GPT-4搭載のエージェントさえ凌駕**するケースがあることです。ToolBenchではDeepAgent（約69%）がGPT-4＋ReAct手法（約52%）よりも高スコアを収めており [62]、モデル規模では劣るものの**アーキテクチャと訓練次第で上回り得る**ことを示しました。この点は業界でも注目され「ボトルネックは知能そのものではなく、従来型の硬直したワークフローにあった」と評価されています [81] [82]。

全体として、DeepAgentは**ツール使用が必要な幅広いタスク**で最先端の成績を示し、その**適応性の高さ**が確認されました [51] [78]。さらに詳細な分析では、(a) **長い行動系列へのスケーリング**、(b) **異なるモデル規模への一般化**という観点でも優れた特性を持つことが示唆されています。まず(a)について、行動上限ステップ数を増やしたときの性能推移を見ると、DeepAgentはステップ数が伸びても安定して性能を向上させ、ReActとの差がむしろ拡大しています [83] [84]。特に50手程度まで許容するとWebShopで顕著に差が開き、DeepAgentが**無駄の少ない効果的な行動選択を行っている**ことが示唆されます [85] [86]。次に(b)のモデル規模では、30Bのバックボーン（Qwen3-30B）から235Bの超大型モデル（Qwen3-235B）に置き換えてもDeepAgentのアプローチは有効であり、**いずれのモデルサイズでも従来手法を上回る**性能を示しました [87] [88]。モデルを大型化すれば全手法性能は向上しますが、特にDeepAgentは**複雑タスクでの性能向上幅が最大**であり、提案手法が高性能LLMを活かしてさらなる恩恵を得られることを示しています [89] [90]。以上より、DeepAgentは多様な尺度で優れた性能を発揮し、その汎用的アプローチが今後のエージェント技術において有望であることが示されました [91] [92]。

3. 応用例（タスクへの適用例とデモ）

DeepAgentが対処できるタスク領域は非常に広範であり、研究チームは論文中で具体例やデモを示しています [51] [53]。**ウェブ情報検索・分析**の文脈では、DeepAgentは高度な“調査エージェント”として機能し得ます [93]。実際、論文のケーススタディでは「映画祭の企画に関する複雑な問い合わせ」にDeepAgentが取り組むデモが紹介されています [94]。このシナリオでユーザは「映画祭を開催するので、Vimeo上のドキュメンタリー動画を探し、関係する映画人の情報を集め、ゲストとして招待したい。さらに上映用のYouTubeライブストリーミングリンクを用意してほしい」といった**マルチステップ依頼**を一度に投げかけます [94]。従来なら人手でも相当な手間となるこの課題に対し、DeepAgentはまず `<tool_search>` を使ってVimeo用の動画検索ツールを見つけ、ドキュメンタリー映像を検索します [95] [96]。次に関連人物を調べるためウェブ検索ツールを動的に発見し、映画監督や出演者の情報を取得、ゲスト候補のリストを生成します。さらにYouTube配信リンクを得るために適切なツールを探し、ストリーミング用のURLを作成します [97] [98]。これら一連の操作を**すべて一つのエージェントの思考流の中で完結**させており、最終的にユーザ質問に対し文書と動画リンクを含む詳細な回答を構成して提示します [97]。DeepAgentはこのように**複数の専門ツールをシームレスに連携**させ、ユーザの高水準な意図を具現化できることを実証しています。特に上記例では、失敗した探索経路があれば途中でメモリを折りたたんで戦略を変えることで、**行き詰まりから回復**して要求をやり遂げています [99] [100]。

他の応用例として、**電子商取引**や**ナビゲーション**の自動化があります。WebShopベンチマークに代表されるように、DeepAgentはオンラインショッピングサイト上で商品を検索・比較し、ユーザの希望に沿った購入プロセスを完遂できます [101]。実験では、DeepAgentが**約92%**という高い成功率でWebShop内のタスク（例えば「特定の条件に合う商品をカートに入れる」など）をこなしており、これは従来手法の**65.7%**を大きく上回っています [102]。この性能は実用上も魅力的で、将来的にはWeb上の複雑な購買代行や情報収集エージェントとして活用できる可能性を示します。

ロボット操作や環境ナビゲーションの分野でもDeepAgentの技術は応用可能です。テキストで家庭内環境を操作するALFWorldでは高い成功率を記録しましたし [78]、この手法を実世界のロボット制御に拡張する研究も考えられます。DeepAgent自体、**視覚質問応答（VQA）**や**ファイル処理**のツールプラグインも備えており [93]、画像やPDFを解析して指示に答えるなど**マルチモーダルエージェント**としての一面も持ち合わせています。事実、GAIAやHLEでは画像解析やコード実行を含む複合タスクで性能を発揮しており [103]、研究チームは**汎用AIアシスタント**への発展可能性を示唆しています [51] [104]。コードと連携する能力については、DeepAgentはPythonコードを実行するツールやOSレベルの操作コマンドも組み込める柔軟な設計です [105]。したがって、将来的にはソフトウェア開発支援（コード自動生成・テスト）やシステム運用自動化といった用途にも展開できるでしょう。

研究デモ環境として、著者らはGitHub上でDeepAgentのコードとデモを公開しています [106] [107]。ベンチマークデータやツールリポジトリも整備されており、ユーザ自身がDeepAgentを動かして応用タスクを試せる状態です [108] [109]。必要に応じて自前のツールプラグインも追加可能で、例えば社内のAPI群をEmbeddingでインデックスしておけばDeepAgentに企業内業務を遂行させることも考えられます。以上のように、DeepAgentは**汎用的なツール使用AIプラットフォーム**として、多様な実世界タスクへの適用が期待されています。

4. 競合モデルとの比較（AutoGPT・ToolFormer・ReAct・Gorilla・ChatDev 他）

DeepAgentのアプローチを理解するため、従来の類似モデルや手法との比較を整理します。以下に主要な競合/関連モデルの特徴とDeepAgentとの差分をまとめます。

- **ReAct**（Reason+Act, 2022） [110]：LLMが思考（Reason）と行動（Act）を交互に繰り返すエージェント手法です。推論ステップごとに観察結果を反映しつつ進む点は画期的でしたが、**使えるツールはあらかじめ定められた限定的なもの**であり [111]、逐次的なループ構造に縛られるため**全体最適なプランニングが困難**でした [112]。DeepAgentはこの枠組みを拡張し、固定ワークフローを排して**単一の思考ストリーム**で柔軟にツールを呼び出すため、ReActに比べ高度なタスクでも破綻しにくくなっています [113] [114]。実際、性能面でもDeepAgentはReActベースラインを各所で凌駕しており（成功率で+15〜20ポイント超の改善例も多数） [115] [66]、**ReActの発展系**としてその弱点を克服した位置付けと言えます。

- **Plan-and-Solve**（2023） [110]：まず高レベルなプランを立ててから順次実行する二段構えのエージェントです。明示的な計画で長期タスクに挑む利点がありますが、プラン作成と実行が分離しているため**計画時に想定外の事態に対応しにくい**問題がありました。DeepAgentは計画フェーズを独立させず**リアルタイムに再計画**できる（メモリ折りたたみを介して方針転換できる）ため、Plan-and-Solveが苦手とする**動的環境への適応力**で勝ります [79] [62]。実験でもWebShopやGAIAのように不確実性の高い環境でDeepAgentがPlan-and-Solveを大きく上回っており [79]、事前計画依存の限界を打破していることが示唆されます。

- **AutoGPT**（2023）： ChatGPT（GPT-4）を用いて自律行動を試みたオープンソースプロジェクトです。ユーザ目標からタスクを自動分割し、逐次実行していく構造を持ち、一時期話題になりました。

しかしAutoGPTは**明確な長期記憶機構を持たず**、各ステップが長大なプロンプトに依存するためコンテキストがすぐ溢れ、誤作動も多く報告されました。また利用できるツールもプラグイン等で増やせますが**事前定義型**であり、DeepAgentのように**大規模ツールセットからその場で探索**する機能はありません。さらにRLによる最適化も行われておらず、一度与えた目標に対し執拗にループして進めてしまうケースも見られます（いわゆる「ループにはまり込む」問題） [22] 。対照的にDeepAgentは**構造化メモリ**でコンテキスト管理を行い [22] [99] 、行き詰まり時にはリセットして軌道修正できるため、AutoGPTに比べはるかに**安定した長時間の自律動作**が可能です。またAutoGPTはGPT-4等の巨大モデル頼みでしたが、DeepAgentは中規模モデル+アルファでも高性能を発揮し [62] 、**アーキテクチャの工夫でモデル性能差を補える**点も対照的です [81] [116] 。総じて、AutoGPTが**大規模言語モデルの汎用性頼み**だったのに対し、DeepAgentは**専門的トレーニングとメモリ・探索戦略**で堅実に実効性を高めたと評価できます。

- **ToolFormer**（2023） [117] ：Meta社らによる研究で、言語モデル自身が学習中に適切なツールAPI呼び出しを文章中に挿入できるようにした手法です。例えば計算が必要なときに電卓APIを呼ぶといった操作をトークン列に自己教師で組み込みました。これは**単発のAPI利用**には有効ですが、ToolFormerが対象とするのは数個の決まったツールのみで、**対話的・逐次的なマルチツール活用**は想定されていません。一方DeepAgentは**任意のツールシーケンス**を扱える汎用エージェントであり、複数ステップにまたがる使いこなしやツール探索を包含します。またToolFormerは生成中にAPIトークンを出力するだけでフィードバックループはありませんが、DeepAgentは**API呼び出し→結果観察→思考継続**のループを回せるため、動的なツール応答にも逐次適応できます [118] [119] 。したがって、ToolFormerが言語モデル内へのツール機能の埋め込みに留まるのに対し、DeepAgentは**モデル外部にツール空間を統合**しフル活用する総合フレームワークと言えます。

- **Gorilla**（2023）： UC Berkeleyらによる API呼び出し特化のLLMです。大量のAPIドキュメントで微調整し、ユーザ要求に適したAPIとその使い方を生成するモデルでした [120] 。単一呼び出しの正確性では優れますが、**対話的な推論プロセス**や**複数ツール連携**は範囲外でした。DeepAgentは必要に応じてAPIドキュメントを検索・参照しつつ連続思考するため、Gorillaの機能も包含しつつより広範なタスクに対応できます。実際、DeepAgentもEmbeddingによるドキュメント検索で**APIの説明書きを読んでから使う**流れを取っており [121] [11] 、その点はGorillaのアプローチと共通します。しかしDeepAgentは一連のタスク完遂まで自律的に進行し、途中で異なるAPIに切り替えたりもするため、**より包括的な「ツール選択→活用→評価」のループ**を回せる点で進んでいます。

- **ChatDev**（2023）： 複数のLLMエージェント（マネージャー、プログラマ、テスター等）の対話によってソフトウェア開発を自動化するフレームワークです。各エージェントが役割を持ち協調する姿は人間チームの擬態で興味深いものでした。しかしながら、このアプローチは**タスクを事前に役割分担**できる前提があり、各サブエージェントは固定機能です。一方DeepAgentは**単一のエージェントで役割を動的に切り替え**られます。例えばDeepAgentは必要に応じ「検索者」「コード実行者」「分析者」といった振る舞いを一人で担えるため、ChatDevのように人数を増やさずとも**内部でマルチスキルを発揮**できます [122] 。ChatDevは主にコーディング領域に特化していましたが、DeepAgentは前述の通りWeb検索からAPI利用まで広くこなせるため、適用範囲もより汎用的です。また複数エージェント間の調整オーバーヘッドが無い分、DeepAgentの方が効率的かつ統一的なポリシーで行動できる利点があります。ただし、一方でChatDevのマルチエージェントのように**明示的な並列性**をDeepAgentは持たないため、将来的にはDeepAgentに複数モジュールの協調動作を組み込む方向も考えられます。

以上より、DeepAgentは**「大規模言語モデル＋ツール」分野の先行研究を包括的に取り込み発展させた存在**だと言えます。ReActやPlan-and-Solveから継承した**逐次思考**の枠組みに、ToolFormerやGorillaの示した**API知識の活用**、AutoGPTで浮き彫りになった**自律ループの課題**への対策（メモリ機構）など、各手法の強みを融合し弱点を補っています。その結果、**固定的でない柔軟な戦略とスケーラブルなツール利用**が可能となり、従来を大きく凌ぐ性能を実現しています [81] [82] 。研究者らも「制約となっていたのはモデルの知能ではなく

アーキテクチャの硬直性だった」と述べており、DeepAgentは**AIエージェント設計のパラダイム転換**を示すものと評価されています [81] [82]。

5. 学術界・産業界での評価と今後の影響

DeepAgentの提案は2025年10月の発表以来、学術・産業双方から大きな注目を集めています。まず**学術界**では、本手法が**ICLR 2025**に採択予定であることからも高い評価が窺えます [123]。大規模言語モデル時代のエージェント研究において、単にモデルを巨大化するのではなく**アーキテクチャ設計の工夫で性能を引き出した点**が評価ポイントです [124] [81]。実際、本論文は「西側の研究者がLLMの外側に足場（scaffolding）を組んでいたのに対し、中国の研究者はモデルの思考を自然に流すアーキテクチャを与えた」と評されています [125]。また、性能面でも「中国のラボが数千のAPI統合を実現し、西側のアプローチを凌駕した」と報じられており [124] [126]、中国発の研究が自律エージェント分野をリードし始めた好例として捉えられています。特にRapidAPI　16,000件から必要APIを見つけ出すというスケールは圧巻で、OpenAIなどが限定的なツール（検索・閲覧・コード実行のみ）に留まっているのと対照的です [7]。この成果は**LLMエージェントの実用化**に向けた重要な一歩と考えられ、今後他の研究者による追試・発展も活発化すると見られます。

産業界やSNS上でも、DeepAgentへの反応は概ね好意的です。技術系メディアは「エージェント分野におけるパラダイムシフト」と伝え [127] [128]、「従来はチェックリストを辿るインターンのようだったAIが、初めて自律的に考え学習し始めた」といった比喩でDeepAgentの登場を称えています [129]。特にImplicatorなどのニュースサイトは**OpenAIのエージェントを凌ぐ成功率**に注目し、「DeepAgentは複雑タスクで89%成功し競合は55%に留まる」と見出しを掲げました [130] [131]。また同記事では「企業にとって本当に役立つエージェントの誕生」だとして、事前設定なしで必要なツールを見出すDeepAgentはビジネスプロセス自動化において有望だと述べられています [94] [132]。実際、WebShopやALFWorldでの顕著な成功は、将来的に**ECサイト運営の自動化**や**ロボットの家庭作業**など具体的応用に繋がる可能性があります [102] [133]。Twitter等でもAI研究者が本手法に言及し、「強化学習とLLMの巧みな組み合わせ」「ツールエコシステムとの統合が見事」といったコメントが見受けられました（※具体的ソースは省略）。総じて**産業界の期待感は高く**、Xiaohongshu社の研究チームが関与していることから、自社プロダクトへの応用も視野に入れていると推測されます。

今後の影響として、DeepAgentは**いくつかの課題と展望**を提起しています [134] [81]。一つは**更なるスケーラビリティ**です。現状の検証では最大50ステップ程度の対話履歴までしか試していないため、100ステップを超えるような**超長　Horizonタスク**でもメモリ折りたたみが有効に機能するかは未検証です [134]。実際の企業シナリオでは何百ものアクションが連なるワークフローもあり得るため、そこへの適用は今後の研究課題です [134]。第二に、**ツールシミュレータで代替した学習**の限界も指摘されています [135]。LMによる模擬API応答は大抵のケースで有用ですが、認証や複雑な内部状態遷移が絡むAPIでは正確に再現できない可能性があります [135]。シミュレータ環境で高得点でも実システムに接続すると不安定になるリスクがあり、**ファインチューニングやドメイン適応**が必要かもしれません。第三に、**リアルタイム性**の問題があります [136]。16,000ものAPIを逐次検索する処理はどれくらいの遅延を伴うのか、論文では詳細が触れられていません [137]。研究用途では許容される待ち時間でも、商用ではミリ秒単位の応答が求められます。ツール検索の効率化（インデックス構造の改良やキャッシュ戦略など）は今後検討すべきポイントでしょう。このように解決すべき技術課題は残るものの、**「ボトルネックはLLM自身ではなく周辺構成にあった」**との認識を改めさせた意義は大きく [81] [82]、DeepAgentのコンセプトは次世代のAIアシスタント開発における基盤となる可能性があります。

以上、DeepAgentの技術的詳細、実験結果、応用例、競合比較、および評価・展望について包括的にまとめました。DeepAgentは**エージェントが真に自律的に「考え、学び、適応する」**ための一つの完成形に近づいたとも言え、今後この路線の研究がさらに発展することで、我々が夢見る汎用AIエージェントが現実のものとなっていくでしょう [138] [129]。

参考文献・出典：本調査レポートでは特に【3】【8】【11】【13】（論文本文）、【25】（Implicator記事）、【24】（Medium解説）などを中心に情報を参照しています。各箇所の【†】付き番号は対応する情報源への出典を示しています。

---

1 2 6 20 21 25 35 37 38 39 44 47 49 50 51 54 55 56 57 58 59 60 61 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 83 84 85 86 87 88 89 90 91 92 95 104 106 110 111 112 115 117 DeepAgent: A General Reasoning Agent with Scalable Toolsets
https://arxiv.org/pdf/2510.21618

3 4 5 13 14 15 27 28 31 33 52 53 93 101 103 105 108 109 113 122 GitHub - RUC-NLPIR/DeepAgent: ◌ DeepAgent: A General Reasoning Agent with Scalable Toolsets
https://github.com/RUC-NLPIR/DeepAgent

7 8 22 26 40 42 43 62 80 81 82 94 96 97 98 99 100 102 114 116 124 125 126 130 131 132 133 134 135 136 137 Chinese researchers ship agent that finds tools mid-thought, beating OpenAI's rigid workflows
https://www.implicator.ai/chinese-researchers-ship-agent-that-finds-tools-mid-thought-beating-openais-rigid-workflows/

9 10 11 12 16 17 18 19 23 24 29 30 32 34 36 41 45 46 48 118 119 121 DeepAgent: A new research enabling General Reasoning Agent | Medium
https://blog.suchitra.me/deepagents-ai-agents-just-got-a-massive-update-013765d8056f?gi=ff987d2212fb

107 [2510.21618] DeepAgent: A General Reasoning Agent with Scalable Toolsets
https://arxiv.org/abs/2510.21618

120 Reviewed: DeepAgent: A General Reasoning Agent with Scalable ...
https://paperverse.io/paper/c0a2323a-0839-4cfc-8d32-4e77388357e6

123 DeepAgent: A General Reasoning Agent with Scalable Toolsets
https://www.researchgate.net/publication/396924577_DeepAgent_A_General_Reasoning_Agent_with_Scalable_Toolsets

127 128 129 138 DeepAgent: The AI That Thinks, Learns, and Remembers — A Paradigm Shift for Autonomous Agents | by ArXiv In-depth Analysis | Oct, 2025 | GoPenAI
https://blog.gopenai.com/deepagent-the-ai-that-thinks-learns-and-remembers-a-paradigm-shift-for-autonomous-agents-64a0e2759b83?gi=a12a174240f6