ChatGPT

LLM層とアプリケーション層の役割と構造の比較分析（Genspark・Manus・OpenAI Operator・Deep Research）

Genspark

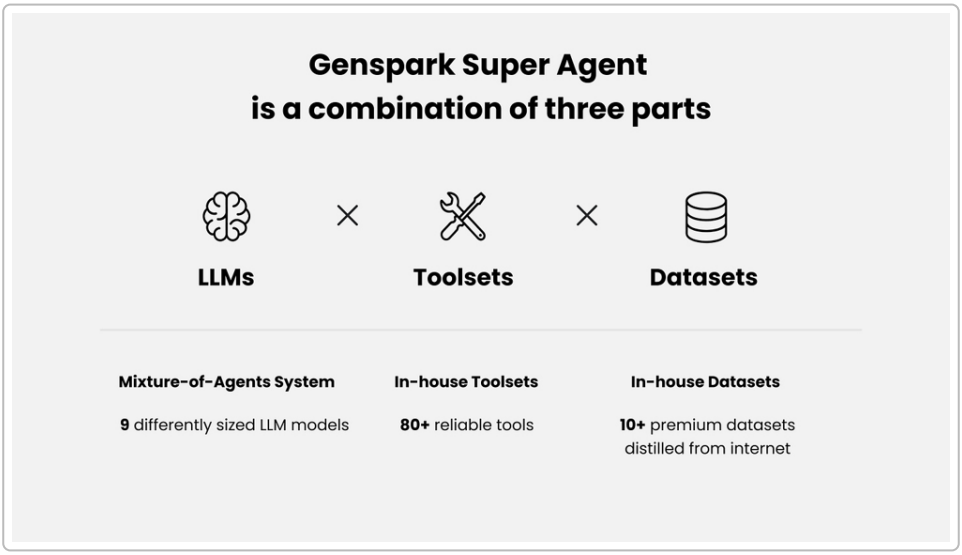

図: Genspark Super Agentのアーキテクチャ（LLMの集合 × 豊富なツールセット × プレミアムデータセット）。Gensparkは“大規模言語モデル(LLM)の集合知”を活用するAIエージェントプラットフォームです。LLM層では**Mixture-of-Agents**方式で9種類のLLMを役割分担させています [1]。各LLMはサイズや得意分野が異なり、ユーザーの問いに対して**タスクに最適なモデルを選択・ルーティング**する仕組みです [1]。また**MCP（Model Context Protocol）**という独自プロトコルで複数モデルやツール間のシームレスな連携を実現し、LLMが生み出す推論結果やコマンドを統合しています [1]。例えば事実関係の抽出に特化したエージェントと文脈分析に長けたエージェントが**同時に協調動作**し、それぞれの成果を統合して回答を構築します [2]。LLM層は推論や要約の中心であり、複数モデルを組み合わせることで単一LLMでは難しい複雑タスクにも高い性能を発揮します。

アプリケーション層では、**80種類以上の専用ツール**や外部データソースへのアクセスを管理しています [3]。ウェブ検索やコード実行、画像・動画生成、音声通話など多岐にわたるツールを内蔵し、LLMからのコマンドに応じて実行します [4] [5]。また**10種類以上の高品質データセット**を内部に持ち、ウェブから蒐集・精製した知識を活用できる点も特徴です【35†画像】。これらツール・データはアプリ層で**統一的に管理**され、LLMは必要に応じて「検索せよ」「計算せよ」などとコマンドを生成し、アプリ層が実行結果をフィードバックする構造です。生成した各ステップの履歴や中間結果は**コンテキストメモリ**に保持され、複数LLMが共有することで長い対話やマルチステップ推論でも一貫性を保っています（MCPにより文脈が共有されています [1]）。

LLM層とアプリ層のインターフェースは、**内部プロトコルとコマンド形式**で設計されています。Gensparkではユーザーの要求をまずLLM層が解析し、必要なアクション（例えば「ウェブ検索:　　○○について」や「Pythonコード実行: データ解析」等）をテキストまたはコード形式で発行します。アプリ層がその指示を解釈して該当ツールを起動し、得られた結果（検索結果のテキストや計算出力など）を再びLLM層へ**テキスト情**

報として返すサイクルになっています。このように**LLMがエージェントとしてツール呼び出しコマンドを生成し、アプリ層が実行して結果を渡す**というループでマルチステップのタスクを進めます。Genspark独自のMCPは、この過程で各モデルやツールが参照すべき文脈や共有データを統一フォーマットで扱う仕掛けと思われ、異種のLLM同士・LLMとツールの情報交換を円滑にしています [1]。

ユーザー視点の機能性・使いやすさ: Gensparkは一見すると高度な検索エンジンのように振る舞います。ユーザーがクエリを入力すると、従来の検索結果一覧ではなく**「Sparkpage」**と呼ばれる動的な要約レポートが生成されます [6]。Sparkpageには質問に関連する多様な情報源から集められた知見が章立てで整理され、重要ポイントの要約と元ソースへのリンクが含まれます [2]。各セクションに**引用元が明示**され信頼性を担保している点も特徴です。UIはシンプルで**検索ボックスに質問するだけ**と容易であり、生成されたレポート内には対話型のAIコパイロットが埋め込まれていて、ユーザーはその場で追加の質問や内容の深掘りができます [7] [8]。このように**対話型に結果を閲覧・追求できる柔軟性**は、通常のチャットボット以上に使い勝手が良いと評価されています [9]。またGensparkは商業色の強い低品質コンテンツをフィルタリングし、公平な情報提供を謳っているため、ユーザーは広告や偏りの少ない客観的なレポートを得られます [10]。さらに高度なタスクでは、画像生成や動画作成を頼んだり、**代理で電話をかけてもらう**ことすら可能で、例えばレストラン予約の電話をエージェントが自動で掛ける機能も提供されています [4] [5]。これら高度な連携もユーザーから見ると自然な対話をしているだけで完結し、**目的指向型に実務を任せられる**体験を実現しています。

開発者視点の設計・拡張性など: Gensparkは「LLMをコアとしたアプリケーション層の革新」を志向しており、**自前で巨大モデルを一から開発するのではなく、既存の複数LLMを統合して能力を引き出す**アーキテクチャになっています [11]。例えばOpenAIの高性能モデル（o1やo3-mini）を推論の司令塔として用い、その指示に基づき社内開発ないし外部提供の各種専門モデル（コード特化モデル、画像生成モデル等）を適材適所で組み合わせる構造です [11]。この設計により、最新かつ最適なモデルを組み替えて使える**LLM非依存性**が確保され、モデルのアップグレードも柔軟です。拡張性の面でも、新しいツールやAPIを追加すればLLMから呼び出せるようになるため、エージェントの能力を後から強化しやすいメリットがあります。実際、リリース後もマルチモーダル対応（画像・動画生成や音声発信機能の追加）など**機能拡張が迅速**に行われています [4]。セキュリティと制御設計については、ツールの実行権限をサンドボックス化するなどの言及はありませんが、Gensparkチームは課題として**プライバシーや安全性確保、エージェントの暴走防止**を挙げています [12] [13]。おそらくツール実行ログの監視や一部ツールの使用制限、出力のフィルタリング等を実装していると考えられます。またOpenAIが提唱する標準インターフェース（Assistant API）の採用検討にも触れており、将来的に外部開発者が統一的にエージェントを操作できる仕組みづくりにも関心を示しています [14]。総じてGensparkは**複数LLMと多様なツールを巧みに編成する設計思想**であり、単一のLLMに依存しない強みを活かしつつ、ユーザーには統合された一つの強力なエージェントとしてサービス提供しています。

Manus

Manusは中国発の汎用AIエージェントで、**「考えるだけでなく結果を出す」**ことを標榜しています [15]。LLM層は特定の単一モデルではなく、**複数の最先端LLMをラップしたオーケストレーター**として機能します [16] [17]。開発当初はAnthropic社のClaude（3.5 “Sonnet”版）を主要な推論エンジンとし、Alibabaの大規模モデルQwenを補助的に組み合わせて利用していました [18]。つまりManus自身で基盤モデルを一から訓練したのではなく、**ClaudeやQwenといった外部LLMを組み合わせた「頭脳」**を持っています [18]。さらに開発チームはバックエンドLLMを積極的にアップグレードしており、Claude 3.7への更新や将来的なGPT-4/Geminiの利用も示唆されています [19]。報道によれば、Manusはタスクに応じて複数モデルを動的に呼び分ける**マルチモデル動的呼び出し**を行っており、例えば**高度な論理推論にはClaude 3系、コーディングにはGPT-4、知識検索にはGoogle Gemini**といった具合に最適モデルを割り当てるそうです [20]。LLM層の役割はこのように**「各分野で最善の頭脳を借りる」**ことであり、一つのモデルに依存せずあらゆるタスクで高い成功率を狙っています [17]。

アプリケーション層の構造は、**クラウド上の仮想マシン環境**を基盤に据えた非常に実行力の高いものです [21] [22]。Manusはクラウド内のUbuntu　Linuxベースのワークスペース上で動作し、インターネットにも接続された**汎用コンピューティング環境**を自前に持っています [21] [22]。その中でManusは人間のパワーユーザと同じように、ウェブブラウザでページを閲覧したり、Linuxシェル（sudo権限あり）でコマンドを実行したり、PythonやNode.jsのコードを走らせたりと**自由度の高いツール操作**を行います [21] [22]。アプリ層はこの仮想環境と、そこで利用可能な各種ツール群（ウェブ、シェル、プログラミング言語インタプリタ等）の管理を担当しています。またManusは**反復型のエージェント制御ループ**によって動作しており、`Analyze → Plan → Execute → Observe`のサイクルで一連のタスクを進めます [23] [24]。まず現在の状態とユーザ目標を解析し [24]、次に取るべきアクションをプランニングし（使用するツールや具体的コマンドの決定）、その内容を**Pythonコードとして実行**します [21] [25]。実行結果（観測）を読み取り、次のループへ反映する―という流れをタスク完了まで繰り返します [26] [27]。Manusのユニークな点は、LLMによるアクション指定を専用のDSLやJSONではなく**汎用のPythonコード（CodeAct方式）**で行っていることです [21] [25]。LLMは「次に何をすべきか」を考えた結果をPythonスクリプトとして出力し、例えばウェブ検索と結果解析を含む複雑な操作も**一度に記述・実行可能**になっています [25]。コード形式の行動生成は柔軟で強力であり、条件分岐やループ、複数ツールの組み合わせを一つのアクションとして記述できるため、単純なAPI呼び出し型より格段に成功率が高まることが研究でも示されています [28]。アプリケーション層はこのコードを実行するインタプリタとして振る舞い、結果ログや出力をテキスト化してLLMに返します。Manusでは**1ステップに1アクションのみ実行**し必ず結果を待つ設計になっており、モデルが暴走して未確認の長いコマンド列を走らせないよう統制しています [29] [27]。この厳格な手順管理により、安全性と結果検証性を確保しつつエージェントの自律実行を制御しています。

さらにManusは内部的に**複数のサブエージェント**を並行動作させる**マルチエージェント構成**を採っています [30] [31]。例えば、メインのLLMが全体方針を決めるマネージャーだとすれば、個別のウェブ情報収集に特化したエージェント、コードを書くエージェント、データ分析するエージェント等が**それぞれ独立した仮想環境（サンドボックス）内で並行して動作**します [32] [33]。最終的にメインのオーケストレーターがそれらの結果を統合し、ユーザーに対する完成アウトプットを得る仕組みです [31]。このように問題を分解し**分担処理させる分散協調型の設計**によって、Manusは非常に複雑で大規模なプロジェクト（例：Excelで報告書を作り上げ、さらにその内容を使ってウェブサイトを公開する等）も一連のユーザー要求から自動で達成できます [34]。ユーザーから見れば単一のAIアシスタントが仕事をやり遂げたように見えますが、その裏では複数エージェントが役割を果たしているわけです [35]。

ユーザーにとっての機能性・利便性:　Manusは「あなたの代わりにあらゆるタスクを完遂するデジタル作業員」として設計されています [36]。ユーザーはやりたいプロジェクトや目的を**ごく簡単に指示するだけ**で、以降の面倒な手順はManusに任せきりにできます [15]。例えば「来月の日本旅行プランを立てて旅行ガイドを作って」と依頼すれば、観光情報を集め日程表を作り、画像付きの小冊子としてまとめるところまで自動でやってのけます（実際に**旅行ハンドブックを生成**するユースケースが公式に紹介されています [37]）。このようにManusは従来のチャットボットのような**提案や回答に留まらず、実際の成果物を出力**できる点が際立っています [38]。ユーザー視点では、Manusとの対話は必要最小限で済み、**「後は任せて休んでいてください」**といった体験になるため非常に手軽です [15]。また様々なユースケース向けにテンプレートや再現ビデオが用意されており、自分のやりたいことに近いシナリオを選んで試すだけでエージェントが自動進行するなど、**学習コストの低さ**も意識されています [39]。アウトプットの形式もレポート、表、コード、動画プレゼンなど多彩であり、目的に合った**最終成果物が直接得られる**ため「助言をもとに自分で作業する」手間が省けます [38]。もっとも自律性が高いぶん、ユーザーが途中経過を細かく指示・修正する余地は少なく、「本当に任せて大丈夫か？」という不安も生じえます。Manus側では各ステップのログや結果を確認できる**ステップ実行のリプレイ機能**を提供しており、ユーザーが後追い検証できるよう配慮しています（公式サイトの**Step-by-step　replays**機能） [39]。総じてManusは**対話の柔軟性よりも自動化・結果重視**の設計ですが、その分ユーザーの負担を劇的に減らし「休んでいる間に全て完了する」利便性を提供します。

開発者視点での設計思想・拡張性など: Manus開発チームは**既存の最高性能モデルをオーケストレーションする**戦略を採り、モデル開発コストを抑えつつ性能と柔軟性を両立しています [17]。ClaudeやQwenといった外

部LLMの組み合わせを状況に応じて更新できるため、特定ベンダやモデルへのロックインがなく**LLM非依存性**が高いと言えます。実際、将来の新モデル（GPT-4やGeminiなど）も適宜取り入れていく方針が示唆されており、**最良の部品を差し替えて使い続ける拡張性**があります [20]。アーキテクチャ上は各種モジュール（プランナー、知識リトリーバー、データソースモジュール等）に分割された**モジュラーデザイン**を採用しています。例えばタスクをブレークダウンする**Plannerモジュール**があり、高度な目標に対しては必要なサブタスクの一覧（疑似コードのようなプラン）を自動生成し、エージェントに順次実行させます [40] [41]。また**Knowledgeモジュール**は内部ナレッジベースから参考情報（スタイルガイドや定石など）を提示し、**Datasourceモジュール**は天気・金融など信頼性の高いAPIから最新データを取得して提供するなど、ただLLM任せにするのでなく専門機能を補佐として組み込んでいます [42] [43]。このように**Retrieval Augmented Generation（外部知識強化）**もサポートしており、モデルの内部知識に頼らず都度信頼情報を取得する設計です [44] [45]。セキュリティと制御面では、前述のように**コード実行を1ステップずつ許可**し逐次検証することで暴走を防いでいます [29]。加えて、おそらく全操作を**コンテナ化（Dockerサンドボックス）**して隔離し、外部への影響を最小化する対策も取っていると考えられます（オープンソースでManus相当物を再現する提案ではDocker隔離が推奨されています [46]）。Manus自体はクローズドソース商用サービスですが、その技術アプローチ（CodeActやマルチエージェント協調）は先進的であり、多くのOSSエージェントに影響を与えています。実際、有志による調査ではLangChainやPlaywright等を組み合わせManusの動作を再現することも可能と示唆されています [47]。もっともManusと同等の信頼性を得るには相当な**プロンプト設計やテスト**が必要で、商用レベルの安定稼働にはハードルがあるとも指摘されています [48]。総合するとManusの設計思想は**「最良のLLM＋汎用の計算環境＋厳格な制御ループ」**で自律エージェントを実現することであり、システム全体を通じて拡張性・柔軟性と安全性のバランスを追求しています。

OpenAI Operator

OpenAI Operatorは、OpenAI社が2025年初頭に公開した**ブラウザ操作特化のエージェント**で、Web上の反復作業を人間の代理でこなすものです [49]。LLM層には**Computer-Using Agent（CUA）**と呼ばれる新たな強化学習モデルが用いられており、GPT-4をベースに**視覚入力と高度な推論能力**を統合したものになっています [50]。このCUAモデルはウェブページのスクリーンショット画像を「見る」能力と、マウス・キーボード操作を「実行する」能力を持ち、まさに人間がGUIを扱うのと同じ感覚でブラウザを操作できます [51]。LLM層（CUA）は与えられたタスク指示に基づき、ページの画面から必要情報を読み取り、次にどのボタンを押すか/どのテキストボックスに入力するか等の**具体的アクション系列を逐次決定**します [50] [51]。その際にはChain-of-Thought形式の内部思考を行い、現在の画面状態や過去の操作履歴を踏まえてマルチステップのプランを動的に組み立てます [52]。例えば「ローマの1日ツアーを最高評価で予約して」という指示に対し、CUAは「TripAdvisorサイトを開く」「観光カテゴリを選ぶ」「ローマのツアーを検索」「ポップアップを閉じる」…等、**必要なGUI操作を順次判断・実行**していきます【26†画像】。LLM層はこのように**視覚認識（Perception）→ 推論（Reasoning）→ 行動（Action）**のループで動作し、最終的にタスクが完了するかユーザーの介入が必要と判断するまで自律的に繰り返します [53] [54]。

アプリケーション層は、Operator専用の**リモートブラウザ環境**（仮想マシン上のWebブラウザ）を提供し、LLMからの操作指示を実際のブラウザ操作にマッピングして実行します [51]。具体的には、アプリ層がブラウザのスクリーンショット画像やDOM情報を取得しそれをLLMに入力、LLMが「クリック」「スクロール」「文字入力」等の操作コマンドをテキストで出力し、アプリ層がそれを受け取って**仮想マウス・キーボードイベント**としてブラウザに適用する、というサイクルです [55] [56]。この通信は内部的には強化学習で直接エンドツーエンドに訓練されていますが、概念的には**画面＝観察空間、マウス/キー操作＝行動空間**としてモデルと環境がやり取りする構造です [57] [52]。Operatorでは個別サイト向けのAPI統合などは一切不要で、**あらゆるWeb GUIをそのまま操作インターフェースとして利用**できる汎用性が特徴です [51]。例えばフォーム入力やボタン押下といった操作はどのサイトでもGUI上は同じ行為なので、モデルが画面を理解しマウス/キー操作できれば特別な実装なしで対応できます。LLM層とブラウザ環境のインターフェースはOpenAI独自にチューニングされており、モデルはテキストによる一連の操作ログ（内部言語）を経由して環境を操作しま

す。OpenAIの研究ブログでは、**「画面（ピクセル）を認識してマウス・キーボードで操作する」**一貫したインターフェースで汎用的なコンピュータ操作を実現した点が強調されています [58] [59]。

ユーザー視点の機能性・使いやすさ: Operatorは現在ChatGPTのProユーザー向けに提供されているウェブアプリ（operator.chatgpt.com）として利用できます [60] [61]。使い方はシンプルで、ユーザーは**やりたい作業内容を自然言語で指示するだけ**です [62]。例えば「特定の商品をネット通販で購入しておいて」と入力すれば、あとはOperatorが自動でブラウザを開き、検索から購入手続きまで進めてくれます。ユーザーは進行状況をリアルタイムで確認することができ、実際の操作ログ（「◯◯ページに移動」「フォームに入力」等）が表示されます【26†画像】。操作が**2分間行われた**等の経過時間や実行ステップが可視化されるため、ユーザーはエージェントの挙動を追跡・理解しやすくなっています【26†画像】。Operatorは煩雑なウェブ上の反復タスク（フォーム入力、情報検索、チケット予約など）を肩代わりし、**日常業務の時間節約**に貢献します [63] [64]。また並行処理も可能で、ブラウザのタブに相当する**複数の会話スレッドを立ち上げて同時に別々のタスクを実行**させることもできます [65]。安全面でのユーザー体験にも配慮されており、ログイン情報の入力や支払いが必要な場面では自動操作をストップして**ユーザーに引き継ぐ**設計です [66]。例えば決済画面で「ここでカード番号を入力してください」とユーザーに促し、ユーザー操作後に再びエージェントが続行するといった挙動です [66]。これによりセキュアな情報はユーザー管理下に置かれ、Operatorに任せっぱなしでも安心感があります。現時点では米国の一部ユーザーへのプレビュー提供であり、未知の不具合もありますが、OpenAIはフィードバックを踏まえて改善・提供範囲の拡大を予定しています [67]。総じてOperatorは**対話型AIを「能動的なWeb操作代行者」へ昇華**させたもので、専門的な知識なしに誰でも使える手軽さで幅広いオンラインタスクを自動化できる点が大きな魅力です。

開発者視点での設計・特性: Operatorの心臓部であるCUAモデルは、**強化学習（RL）技術を駆使してGUI操作を習得**しています [50]。GPT-4の画像理解力と推論力を土台に、実際のブラウザ環境で何度もタスクを試行し報酬を与えることで、「どの画面で何をクリックすれば目的達成できるか」を学習させました。これにより既存のルールベースRPAとは異次元の汎用性と適応力を獲得しています。研究ブログによれば、CUAはWebブラウジングエージェントのベンチマークであるWebArenaやWebVoyagerで従来を大きく上回る成功率を達成しており [68] [69]、**見たことのないサイトでも動的に対処できる**性能が示されています。設計哲学としては、個別サイト毎の専用プログラムを書くのではなく**「画面」という普遍的インターフェースにAIを適応させる**発想で、これによりインターネット全体を操作対象とできる点が革命的です [51]。拡張性の面では、ブラウザ操作以外にもOS全体のGUI操作（例えばファイルマネージャや設定アプリの操作）に今後拡大する可能性があります。実際CUAモデルは汎用のPC操作ベンチマーク(OSWorld)でも成果を出しており、将来的には**PC作業全般を代行するエージェント**への発展も視野に入ります [68] [69]。もっともカバー範囲が広い分、安全対策にも細心の注意が払われています。OpenAIはOperatorのシステムカードで、エージェントがデジタル世界へアクセスするリスクに対処する方策を述べています [70]。例えば**重要処理にはユーザー確認を求める**（前述のログイン等）ほか、想定外の挙動時には自動でユーザーに制御を返すフェイルセーフを実装しています [71] [72]。また権限管理や操作ログの透明化も行っており、操作の一挙手一投足を記録・監査できるようにしています。開発者（＝OpenAI）側から見ると、Operatorは限定公開の研究プレビュー段階であり、**現実の複雑さをモデルに経験させながら洗練させている**途中です [67]。将来的にはChatGPT本体への統合やAPI提供も計画されており [67]、プラットフォームとしての発展も期待されます。現時点ではOpenAI独自モデルに強く依存したソリューションですが、その成果は業界にインパクトを与えており、「GUIを直接操作するAI」という新しい応用領域を切り拓いた点で意義深いプロジェクトです。

OpenAI Deep Research

OpenAI Deep Research（以下Deep Research）は、ChatGPT上で2025年2月にリリースされた**高度なインターネット調査エージェント機能**です [73]。複雑な質問を入力すると、AIが自律的にウェブ検索から情報収集・分析・統合し、最終的に**数ページにわたる包括的な調査レポート**を生成します [74]。LLM層にはOpenAIが新たに開発中の**o3モデル**系列の一つが使われており、ウェブ閲覧やデータ分析に最適化されたバージョンがDeep Researchの頭脳を担っています [75]。このモデルは**数百ものオンライン情報源を見つけ出し、分析**

し、総合する推論力を持ち、研究者アナリスト並みの詳細なレポートを自動作成できるよう調教されています [75]。特筆すべきは、モデルが**マルチステップのプランニングと動的な方針転換**を学習している点です [76]。エンドツーエンドの強化学習により、難易度の高い閲覧＆推論タスクを大量に経験させた結果、必要なデータを得るために**検索計画を立て、必要に応じて軌道修正しながら調査を進める**能力が獲得されています [76]。例えば「ある分野の最新動向」について尋ねられた場合、まずサブトピックに分割して順次検索し、各トピックごとに有力な情報源を多数当たり、得られた知見を付き合わせて結論を導く――といった**人間さながらの調査フロー**をモデル自ら考えて実行します [76]。

アプリケーション層はChatGPTのエージェント実行基盤上に構築されており、**Web閲覧用ツール**と**Python実行環境ツール**が統合されています [77]。LLM（o3モデル）は内部で「検索」「閲覧」「コード実行」等の**ツール使用コマンドを生成**でき、システムはそのコマンドを受けてブラウザで検索・ページ取得したり、Pythonスクリプトを実行してグラフを描画したりします [77]。得られたページ内容や実行結果データは再びモデルにフィードバックされ、モデルはそれを解析して次のアクションを決める、というループがタスク完了まで自動で続行します。Deep Research用モデルはテキストと画像の両方をコンテキストとして扱えるため、ウェブから取得した画像やPDFも解析対象です [78]。またPythonツールで生成したプロット図を画像として回答に埋め込んだり、ウェブ上で見つけた有用な図表を引用表示することもできます [77]。最終的なアウトプットでは、参照した情報源の**一文一文まで正確に出典として明記**するよう設計されています [77]。アプリ層のメモリ管理は、調査中に得た知見や中間要約を逐次モデルのコンテキストに保持する形で行われます。必要に応じてモデルは「どの情報を既に得たか」「何がまだ不明か」を自己点検し、新たな検索クエリを発行する**Reflectiveループ**も実装されています [79]（例：「Xについてもっと情報が必要か？」と内省し、必要なら追加検索する）。こうした**反復探索**を経て十分情報が集まると、モデルはレポート執筆フェーズに移行し、イントロダクションから結論まで一貫した文章を生成します。その際、得たデータを引用形式で差し込みつつ、自身の言葉で要点を説明・比較・分析するという高度な処理を行います [80]。

ユーザー視点の機能性・使いやすさ: Deep ResearchはChatGPTの一機能として提供され、ユーザーは通常のChatGPTと同様に**自然文で質問を投げかけるだけ**で利用できます。例えば「サッカー選手の平均引退年齢は？なぜポジションによって差があるのか？」と質問すると、Deep Researchモードでは人間が何時間もかける調査を**数十分で代行**し、包括的なレポートを返してくれます [73]。回答には序論・データ分析・具体例・結論といった構成が取られ、図表や引用も含まれるため非常に読み応えがあります。実際、従来モデルが簡潔に数値を答えるだけだった質問でも、Deep Researchは「様々な要因や背景を考慮した詳細な説明」を行います [80]。例えばNFL選手の引退年齢に関する質問では、単に平均年齢を答えるのではなく、**ポジション別の寿命の違いやその理由（キッカーが長く現役を続けられる背景など）**を統計データとともに解説し、ユーザーが「なぜ」を理解できるようにしています [80]。このように**回答の精緻さと明確さ**において抜きん出ており、知的好奇心に応える質の高いアウトプットが得られるのが利点です。対話の柔軟性も確保されており、ユーザーは生成されたレポートに対してさらに「この部分を詳しく教えて」「別の角度からも分析して」等と追問できます。その場合、エージェントは既存の調査結果を踏まえつつ追加の検索や分析を行い、補足情報を提供します。ツール連携の動作は舞台裏で行われるため、ユーザーには**一連の会話として自然に映る**点も使いやすさに寄与しています。レスポンス精度に関しては、参照元が明示されることで**事実性の裏付け**が取れる安心感があります。モデル自体も事実誤認を減らす訓練をされており、引用箇所の厳密さ（文章単位での引用）からも回答の信頼性は高いです [77]。ただし高度な推論を伴うため**応答に時間がかかる**場合があり、簡単な質問でも数分以上待つケースがあります（その分しっかりした内容を返す設計です）。現在はPlus/Proユーザー向けに月あたりの利用回数制限付きで提供されており [81]、必要性の高い時にだけ使う「調査モード」の位置づけですが、この制限は将来的なモデル効率向上で緩和される見込みです [81]。

開発者視点での設計・拡張性など: Deep ResearchはOpenAIによる**次世代エージェントの一形態**であり、先行するOpenAI o1モデルから継承したコード・数学能力に加え、大量文脈処理と情報検索能力を強化した設計です [82]。End-to-Endの強化学習によってツール使用まで含めた複雑タスクを学習させる手法は、研究的にも新規性が高く（先行例としてWebGPTやMRAIなどがありましたが、それを大規模に推し進めています）、Deep Researchのモデルは数多くの公開ベンチマークでSOTAを更新しています [82] [83]。例えば人類最難関テストと称される「Humanity's Last Exam」では従来モデルを大きく上回るスコアを記録し、特に化学・人

文・数学分野で大幅な向上を示しました [84] [85]。また実世界の問題に答えるGAIAベンチマークでも従来トップを抜き去り総合1位となっています [83] [86]。これらは**長文推論やマルチモーダル解析、ツール使用を統合**したモデル設計の優位性を物語っています。開発面では、OpenAIが同時期に発表した**「OpenAI Agents SDK」**が深く関係します。これは外部開発者がOpenAI流のエージェント（ツール使用型LLM）を構築できる枠組みで、Deep Research相当のワークフローを任意のモデルで再現することも可能です [87] [88]。実際、有志によるOSS実装「OpenDeepResearch」では、Agents SDK上で**任意のOpenAI API互換モデル（例えばGeminiや地元LLM）**を使って類似の調査エージェントを構築できることが示されています [88] [89]。この実装ではトピックの初期プランニング、サブトピックごとの平行調査、結果の統合といったステップをPipeline化しており、Deep Researchの内部処理を簡易化した形で追体験できます [90] [91]。これはDeep Researchの設計が比較的モジュール分割しやすく、**LLMからツールへのコマンド体系やタスク分割戦略が汎用的**であることを示唆します。セキュリティ・制御の観点では、Deep Researchは基本的に閲覧・計算というホワイトな操作のみを行い、悪意ある行動は取りえないよう制限されています。アクセスするサイトも一般公開情報が中心で、企業のログイン情報などは扱いません。また引用先が自動で記録されるため、モデルが不適切な情報源を参照すればユーザーに透けて見える仕組みであり、**透明性による牽制**が働いています。LLM非依存性について言えば、Agents SDKによりOpenAI以外のモデルでも同様の処理を行える道が開けていますが、現状Deep Research自体はOpenAIの専用モデル・専用インフラ上で完結しており、ブラックボックス的な部分も多いです。しかしその分チームによる厳格なテストと安全対策が施されており、またPlusユーザーへの限定提供でフィードバックを集めることで、安全かつ有用なエージェント機能へと洗練させています [73] [67]。総じてDeep Researchは**OpenAIレベル3エージェント**（長時間自律行動できるAI）の先駆けとも言える存在であり、ユーザーに新たな価値を提供すると同時に、開発者コミュニティにも多くの知見を提供しています。

まとめ: 各システムの比較表

最後に、Genspark・Manus・Operator・Deep Researchの特徴を**ユーザー視点**と**開発者視点**の観点でまとめます。

観点	Genspark	Manus	OpenAI Operator	OpenAI Deep Research
LLM層の構成	複数LLMによるMixture-of-Agents（9モデルをタスク別に利用） [1]。推論専用のコアLLM+各種専門LLMで協調	外部の高性能LLMを組合せたラッパー（ClaudeやQwen等） [18]。タスクに応じて複数モデル動的呼び出し [20]	単一の強化学習LLM（GPT-4ベースCUA） [50]。視覚入力+推論を統合し行動決定	単一の高度LLM（OpenAI o3系） [75]。長文推論特化でブラウジング・コード実行機能を統合学習 [76]
アプリ層・ツール	80+の内蔵ツール＆10+データセット [3]。検索・コード・画像/動画生成・電話発信まで網羅 [4]	クラウドVM上のブラウザ・Shell・言語実行環境等フル装備 [22]。汎用PC操作やAPI呼び出しも可能	リモートブラウザ環境（画面/マウス/キー操作） [51]。追加API不要でWeb GUIをそのまま操作	ChatGPTプラグイン環境（Web閲覧ツール+Python実行） [77]。取得ページテキストやコード出力をモデルに供給

観点	Genspark	Manus	OpenAI Operator	OpenAI Deep Research
LLMとアプリの連携	内部プロトコル(MCP)でコンテキスト共有 [1]。LLMがテキスト/コードでツール指示→アプリが実行結果をテキスト化返送	LLMがPythonコード（アクション）生成 [25] →VMで実行→結果ログをLLMにフィードバック。1サイクル1アクションで逐次制御 [29]	モデルに画面スクショ入力→LLMが操作指示（クリック等）→環境に適用→新スクショ取得…をループ [54] [92]	LLMが関数呼び出しで「検索/閲覧/計算」を要求→ブラウザやPythonツール実行→テキスト結果をモデルに返す反復。最終回答に引用や図を埋め込み [77]
ユーザー体験	**高度な検索レポート**（Sparkpage）として結果提示 [6]。段落構成・引用リンク付き、対話で深掘り可能 [8]。画像/動画生成や電話代行など**多数の機能をチャットで利用**可 [4] [5]	**「任せて休める」自動遂行** [15]。プロジェクト全体を指示→完成品（レポート,コード,サイト等）を直接出力 [38]。途中経過はリプレイで閲覧可能。人手作業の丸ごと代行	**ブラウザ操作の代行**。日常のウェブタスクをチャット指示で自動化 [63]。操作ログが逐次表示され安心感あり。必要時はユーザーに入力要求し協調 [66]。複数タスク並行可能 [65]	**調査レポート自動生成**。複雑問いにも統計や背景まで網羅した長文回答 [80]。引用・グラフ付きで信頼性◎。ChatGPT上で利用、追加質問でさらなる深堀りも自然にできる
開発視点	**LLM非依存・組合せ志向**: 最良モデル群とツールを統合 [11]。モジュール増減で機能拡張容易。安全性はログ監視やAPI標準化検討 [14] 等	**汎用環境＋制御ループ**: 既存LLMを活用し迅速開発 [17]。マルチエージェント構造で並行処理と専門性確保 [32]。Docker隔離や逐次実行で制御。OSS再現も可能だが調整困難 [48]	**RLと視覚で汎用性追求**: 個別サイト対応不要でスケーラブル [51]。安全設計重視（手動確認箇所設定等） [66]。OpenAI専用モデル依存だが今後API公開の可能性	**高度統合モデル**: エンドツーエンド学習で検索+推論+コード実行を一体化 [76]。評価ベンチでSOTA達成 [84]。Agents SDK公開で他モデルへの展開も視野 [88]。OpenAI管理下で安全運用

＜出典＞ Genspark: [93] [2] [9] ほか; Manus: [18] [26] [45] ほか; OpenAI Operator: [50] [54] [66] ほか; OpenAI Deep Research: [74] [76] [80] ほか.

---

[1] [3] [4] [5] [93] INSANE One-click MCP AI Agent Hits the Market | HackerNoon
https://hackernoon.com/insane-one-click-mcp-ai-agent-hits-the-market

[2] [6] [7] [8] [9] [10] Genspark and Its Alternatives: An In-Depth Analysis of AI Research Agents | by ByteBridge | Medium
https://bytebridge.medium.com/genspark-and-its-alternatives-an-in-depth-analysis-of-ai-research-agents-d23feaaddc25

[11] Don’t Sleep On Genspark’s Super Agent - by Daniel Nest
https://www.whytryai.com/p/genspark-super-agent

[12] [13] [14] Ai Agents Framework - Genspark
https://www.genspark.ai/spark/ai-agents-framework/58fdf85a-a516-3742-b7fa-3a5273646c75

15 37 39 Manus
https://manus.im/

16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 40 41 42 43 44 45 46 47 48 In-depth technical investigation into the Manus AI agent, focusing on its architecture, tool orchestration, and autonomous capabilities. · GitHub
https://gist.github.com/renschni/4fbc70b31bad8dd57f3370239dccd58f

38 Manus在紅什麼？外媒評測訂餐、訂位、訂票…都碰壁：它是中國第二個DeepSeek時刻？
https://tw.stock.yahoo.com/news/manus%E5%9C%A8%E7%B4%85%E4%BB%80%E9%BA%BC-%E5%A4%96%E5%AA%92%E8%A9%95%E6%B8%AC%E8%A8%82%E9%A4%90-%E8%A8%82%E4%BD%8D-%E8%A8%82%E7%A5%A8-%E9%83%BD%E7%A2%B0%E5%A3%81-043708332.html

49 50 51 60 61 62 63 64 65 66 67 71 72 Introducing Operator | OpenAI
https://openai.com/index/introducing-operator/

52 53 54 55 56 57 58 59 68 69 70 92 Computer-Using Agent | OpenAI
https://openai.com/index/computer-using-agent/

73 74 75 76 77 78 80 81 82 83 84 85 86 Introducing deep research | OpenAI
https://openai.com/index/introducing-deep-research/

79 A Comparison of Deep Research AI Agents | by Omar Santos | AI Security Chronicles
https://aisecuritychronicles.org/a-comparison-of-deep-research-ai-agents-52492ee47ca7

87 88 89 90 91 Open Source Deep Research (using the OpenAI Agents SDK) : r/AI_Agents
https://www.reddit.com/r/AI_Agents/comments/1jk14wz/open_source_deep_research_using_the_openai_agents/