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に関する包括的研究報告書

Gemini

1. 序論：自律型AIエージェントの進化と現在の課題

1.1 大規模言語モデルから自律的エージェントへのパラダイムシフト

人工知能研究の最前線において、大規模言語モデル（LLM）は静的な知識ベースとしての役割を超え、環境と相互作用し、複雑なタスクを遂行する「自律型エージェント（Autonomous Agents）」へと進化を遂げています。2025年10月、中国人民大学（Renmin University of China, RUC）の高知能自然言語処理研究所（RUC-NLPIR）と、ソーシャルメディアプラットフォーム大手である小紅書（Xiaohongshu）の研究チームによって発表された「DeepAgent」は、この進化の過程における重要な転換点を示すものです [1]。

従来、LLMを用いたエージェント開発は、モデルの推論能力そのものよりも、プロンプトエンジニアリングや外部スクリプトによる制御構造（ワークフロー）に依存していました。代表的な手法であるReAct（Reason-Act）やCodeAct、Plan-and-Solveなどは、エージェントに対して「思考し、行動し、観察する」という固定的なループを強制します [2]。これらのフレームワークは、特定の狭いドメインや、事前に定義された少数のツールセット内では一定の成果を上げてきましたが、現実世界の複雑で動的なタスクにおいては、その硬直性が深刻なボトルネックとなっていました。

DeepAgentの研究チームが指摘する主要な課題は、既存のフレームワークが「事前定義されたワークフロー（Predefined Workflows）」に依存している点にあります [2]。現実世界のタスクは予測不可能です。例えば、ユーザーの曖昧な要求から出発し、必要な情報を収集する過程で新たな疑問が生じ、使用すべきツールが当初の想定とは異なるものになることは日常茶飯事です。このような状況下では、静的な「思考-行動」ループは脆く、予期せぬエラーや探索の行き詰まり（Dead end）に対処できません [4]。

1.2 長期的インタラクションにおけるコンテキストの限界

エージェントが直面するもう一つの重大な課題は、長期的なタスク遂行に伴う「コンテキスト長の爆発（Context Length Explosion）」です [2]。複雑なタスク、例えば「特定の映画祭を企画し、候補作品を選定し、関係者に連絡を取る」といったシナリオでは、エージェントは検索エンジン、データベース、メールツールなどを数百回にわたって呼び出す可能性があります。

既存のエージェントアーキテクチャでは、これらのインタラクション履歴（ツールの実行結果、エージェントの思考過程、エラーログなど）が全て生のテキストとしてコンテキストウィンドウに蓄積されていきます。その結果、以下の3つの致命的な問題が発生します。

1. **情報の埋没:** 初期の重要な指示や、数ステップ前に得られた重要な手掛かりが、膨大なログの中に埋もれてしまい、LLMの注意機構（Attention Mechanism）が適切に機能しなくなります [4]。
2. **コストとレイテンシの増大:** コンテキストが長くなるほど、推論にかかる計算コストと時間は指数関数的に増大します。
3. **推論精度の低下:** 無関係な履歴や失敗した試行のログがノイズとなり、モデルが幻覚（Hallucination）を起こしたり、誤った推論パスに固執したりする原因となります [5]。

1.3 DeepAgentの提案：統合的推論への回帰

これらの課題に対し、DeepAgentは「Unified Agentic Reasoning（統合的エージェント推論）」という概念を導入しました [1]。これは、ツール使用やメモリ管理を外部の制御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というアプローチです。DeepAgentは、単一の思考ストリームの中で、自律的に「考えること」「ツールを探すこと」「行動すること」、そして「記憶を整理すること」をシームレスに実行します [6]。

本報告書では、DeepAgentのアーキテクチャ、特にその中核をなす「自律的メモリ畳み込み（Autonomous Memory Folding）」機構と、スケーラブルなツール探索能力、そしてそれらを支える新しい強化学習手法「ToolPO」について、技術的な詳細を徹底的に分析します。また、ベンチマークにおける定量的評価に基づき、なぜこのアーキテクチャが従来手法を凌駕するのか、その深層にあるメカニズムを解明します。

2. アーキテクチャと理論的枠組み

2.1 エンドツーエンドの深層推論エージェント

DeepAgentの最も革新的な点は、その「エンドツーエンド（End-to-End）」な設計思想にあります。従来のモジュラー型エージェント（例えばLangChain等で構築されるシステム）では、プランナー、ツールセレクター、エグゼキューターといった異なるモジュールが独立して動作し、それらをコードで繋ぎ合わせていました。対照的に、DeepAgentは、強力な「メイン推論モデル（Main Reasoning Model）」がすべてのプロセスを主導します [1]。

この統合された推論プロセスにおいて、モデルは以下の4つの異なるアクションタイプを、文脈に応じて動的に生成します [7]。

アクションタイプ	機能概要	役割と意義
内部思考 **(Internal Thought)**	自然言語による推論と計画	タスクの現状分析、次の一手の検討、論理的な整合性の確認を行います。外部への出力ではなく、モデル内部の自己対話として機能します。
ツール検索 **(Tool Search)**	<tool_search>トークンの発行とクエリ生成	必要なツールが手元にない場合、外部のツールレジストリ（16,000+ API）から動的にツールを検索・取得します。
ツール呼び出し **(Tool Call)**	<tool_call>トークンとJSON引数の生成	特定されたツールを、正確な引数とともに実行します。API仕様に基づいた厳密な構文生成が求められます。
メモリ畳み込み **(Memory Fold)**	<fold_thought>トークンの発行	現在のコンテキストが冗長、あるいは探索が行き詰まったと判断した際、自律的に記憶の圧縮・再構成プロセスを起動します。

この設計により、エージェントは「検索してから考える」「失敗したら記憶を整理して出直す」といった、

人間のような柔軟な認知戦略をとることが可能になります [6]。これは、従来の「思考→行動→観察」という固定ループからの完全な脱却を意味します。

2.2 モデルバックボーンとハイブリッド構成

DeepAgentのフレームワークは、単一の巨大モデルだけに依存するのではなく、役割分担された複数のモデルによって構成されています。論文およびリポジトリの推奨構成によると、推論の中核を担うメインモデルには、高度な論理的思考能力を持つモデルが採用されています [6]。

特に注目すべきは、Qwen（通義千問）シリーズの「Thinking」モデルや「QwQ」モデルの採用です。

- メイン推論モデル **(Main Reasoning Model):** QwQ-32B や Qwen3-30B-A3B-Thinking など、Chain-of-Thought（思考の連鎖）能力が強化されたモデルが推奨されています。これらは、複雑な問題をステップバイステップで分解し、長期的な整合性を保つ能力に長けています [6]。
- 補助モデル **(Auxiliary LLM):** 記憶の圧縮やツール検索結果のフィルタリングなど、推論負荷は低いものの高い処理速度と指示従順性が求められるタスクには、Qwen2.5-Instruct などの軽量かつ高速なモデルが使用されます [1]。

この「メインモデル＋補助モデル」のハイブリッド構成は、計算資源の効率化とエージェントの応答性の向上を両立させるための実用的な設計です。特に、後述するメモリ畳み込みのようなバックグラウンド処理を補助モデルにオフロードすることで、メインモデルは常にクリアなコンテキストで推論に集中することができます。

3. 自律的メモリ畳み込み機構（Autonomous Memory Folding）

3.1 認知科学的基盤と「構造化された忘却」

DeepAgentの技術的優位性の核心は、「自律的メモリ畳み込み（Autonomous Memory Folding）」にあります。これは単なる技術的なデータ圧縮ではなく、人間の認知プロセス、特に「ワーキングメモリ」と「長期記憶」の相互作用に強く触発されたメカニズムです [4]。

人間は複雑な問題を解決する際、過去の全ての詳細な行動ログを記憶しているわけではありませ

ん。「試行錯誤の結果、この方法は駄目だった」「現在の目標はこれである」といった、抽象化・構造化された情報を保持し、不要な詳細（例えば、失敗した試行の具体的な手順など）は意図的に忘却します。DeepAgentはこのプロセスを工学的に模倣しています。

3.2 トリガーと実行プロセス

エージェントは推論の過程で、以下の条件を満たしたと判断した場合、自律的に <fold_thought> トークンを生成します [1]。

1. コンテキストの飽和: インタラクション履歴が長くなりすぎ、推論効率が低下する恐れがある場合。
2. サブタスクの完了: 一連の作業区切りがついた段階。
3. 探索の行き詰まり: 現在のアプローチが失敗し、戦略を練り直す必要がある場合（Dead end detection）。

このトークンが生成されると、メインの推論ストリームは一時停止し、補助LLMが起動します。補助LLMは、これまでの全インタラクション履歴（s_t）を入力として受け取り、それを分析・圧縮し、特定のJSONスキーマに従って再構築します [1]。

3.3 脳模倣型メモリ・スキーマの詳細

圧縮されたメモリは、自由形式の要約テキストではなく、厳格に定義されたJSON構造として保存されます。この「構造化（Structured）」という点が極めて重要です。自由形式のテキスト要約は、情報の欠落や幻覚（Hallucination）を引き起こしやすいのに対し、JSONスキーマは情報の粒度と関係性を保持し、モデルが再読み込みした際の解釈の曖昧性を排除します [4]。

DeepAgentのメモリシステムは、以下の3つの主要コンポーネントで構成されています。

3.3.1 エピソード記憶 (Episodic Memory)

- 定義: タスク全体のハイレベルな進行状況を記録する長期記憶。
- 内容: 完了したサブタスク、重要な意思決定の分岐点、これまでに判明した確定事実。
- JSON構造の役割: 時間的な順序関係と因果関係を保持します。「Aを実行した結果Bが判明したため、Cの方針を採用した」というような文脈の大枠を提供し、エージェントがタスクの全体像（

Global Perspective）を見失うのを防ぎます [6]。

3.3.2 ワーキングメモリ (Working Memory)

- 定義: 現在の焦点（Attentional Focus）を維持するための短期記憶。
- 内容: 直近のサブゴール、現在実行中のプラン、保留中の疑問点。
- 役割: コンテキストがリセットされた後でも、エージェントが即座に作業を再開できるようにします。「今何をしようとしていたか」という即時的な文脈を提供します [6]。

3.3.3 ツール記憶 (Tool Memory)

- 定義: ツール使用に関する経験的知識（Procedural Knowledge）の蓄積。
- 内容: 使用したツールの名称、有効だった引数のパターン、発生したエラーの種類、ツールの特性に関する洞察。
- 重要性: これはDeepAgent独自の特徴であり、試行錯誤からの学習を可能にします。例えば、「このAPIは日付フォーマットとしてYYYY-MM-DDを要求し、YYYY/MM/DDではエラーになる」といった具体的な知見を記録することで、将来のインタラクションにおいて同じエラーを繰り返すことを防ぎます [6]。

この3層構造のメモリシステムにより、DeepAgentは「深呼吸（take a breath）」をするようにコンテキストを一新し、圧縮されたしかし高密度な情報を持って推論を再開することができます [6]。これは、従来の「要約」手法が抱えていた「詳細情報の喪失」と「コンテキスト長」のトレードオフを解消する画期的なソリューションです。

4. スケーラブルなツールセットと動的検索機構

4.1「オープンセット」ツールの課題

従来のエージェント研究（例えばToolBenchやGPT-4のプラグイン機能）では、使用可能なツールは数十個程度に限定されており、それらの定義はプロンプト内にハードコードされていました。しかし、現実のウェブ環境やエンタープライズ環境には数千、数万のAPIが存在します。これら全てをプロン

プトに含めることは物理的に不可能です。

DeepAgentは、**16,000**以上の**RapidAPI**を含む大規模なツールセットに対応するように設計されています [6]。この「オープンセット（Open-set）」環境において、エージェントは事前に与えられた道具だけでなく、未知の道具を自ら発見し、使用方法を学習する必要があります [7]。

4.2 ツール検索メカニズムの実装

DeepAgentは、ツール検索を「思考の一部」として統合しています。具体的なメカニズムは以下の通りです。

1. 高密度検索（Dense Retrieval）とベクトルデータベース:
   すべてのツール（API）のドキュメント（説明文、引数定義など）は、事前に埋め込みモデル（Embedding Model、例：bge-large-en-v1.5など）を用いてベクトル化され、高密度インデックス（Dense Index）としてデータベースに格納されています 7。
2. 動的クエリ生成:
   エージェントは推論中に「現在の問題を解決するための手段がない」と判断すると、<tool_search>トークンに続いて検索クエリ（自然言語）を生成します。例えば、「映画祭のためにドキュメンタリー映画を探すAPIが必要だ」といった思考がクエリに変換されます 4。
3. コンテキストへの動的注入:
   システムは生成されたクエリとベクトルデータベース内のツール定義とのコサイン類似度を計算し、最も関連性の高いトップK個のツールを取得します。これらのツール定義は即座にLLMのコンテキストに挿入され、エージェントはその瞬間からそのツールを使用可能になります 9。

4.3 戦略的意義

このメカニズムにより、DeepAgentは「静的な知識」に縛られず、インターネット上の広大な「動的な機能」にアクセスできるようになります。例えば、映画祭の企画タスクにおいて、Vimeoでの動画検索、LinkedInでの専門家検索、Google Calendarでのスケジュール調整といった、全く異なるドメインのツールを、必要に応じて次々と発見・連携させることが可能になります [4]。これは、OpenAIなどが提供している「Search, Code, Browsing」という限定されたツールセットとは対照的であり、エージェントの汎用性を劇的に高めるものです [4]。

5. トレーニング手法：ToolPO (Tool Policy Optimization)

DeepAgentの高度な振る舞いを支えるのが、新たに開発された強化学習（RL）手法「ToolPO」です。

5.1 既存の学習手法の限界

自律型エージェントのトレーニングには、これまでSFT（教師あり微細調整）や、一般的なRL手法（PPOなど）が用いられてきました。しかし、これらには以下の問題がありました。

- **SFTの限界:** 教師データ（人間の操作ログやGPT-4の生成ログ）を模倣するだけでは、未知のエラーからの回復や、創造的なツール使用を学習できません。
- **報酬のスパース性:** エージェントのタスクは多段階にわたるため、最終的に「成功したか否か」という報酬しか得られない場合が多く、どの中間ステップ（どのツール呼び出し）が良かったのか、あるいは悪かったのかを特定すること（Credit Assignment Problem）が困難でした [7]。
- **環境の不安定さとコスト:** 実際のAPIを使用して何万回もの試行錯誤（Rollout）を行うことは、API利用料、レート制限、ネットワークエラーなどの観点から非現実的です [10]。

5.2 ToolPOの革新点

ToolPOは、これらの課題を解決するために設計された、ツール使用特化型のエンドツーエンドRL戦略です。

5.2.1 LLMベースのツールシミュレータ (Tool Simulator)

DeepAgentチームは、実際のAPIを叩く代わりに、補助LLMを用いた「ツールシミュレータ」を開発しました [8]。

- **機能:** このシミュレータは、エージェントが生成したAPIコールを受け取り、実際のAPIが返すであろうレスポンス（成功時のJSONデータや、エラーメッセージ）を予測して返します。
- **利点:** これにより、外部APIの不安定さやコストを排除し、高速かつ安定した環境で大規模な試行錯誤（Exploration）が可能になります。エージェントは「安全な箱庭」の中で、失敗を恐れずに多様な戦略を試すことができます。

5.2.2 微細なアドバンテージ帰属 (Fine-grained Advantage Attribution)

ToolPOの最も技術的な貢献は、報酬の割り当て方法にあります 2。
通常のRLでは、一連の行動系列（Trajectory）全体に対して報酬を与えますが、ToolPOは「ツール呼び出し（Tool Call）」に関連するトークンに対して、きめ細かく信用（Credit）を割り当てます。

- メカニズム: あるツール呼び出しが行われた際、その行動が最終的な成功確率をどれだけ高めたか（Advantage）を計算し、そのツール呼び出しを構成するトークン群に直接報酬をフィードバックします。
- 効果: これにより、エージェントは「タスク全体としては失敗したが、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学習できるようになります。実験結果において、ToolPOはGRPO（Group Relative Policy Optimization）と比較して、学習中の報酬の変動が少なく、より高い収束性能と安定性を示しています [3]。

6. 実験結果とベンチマーク評価の分析

DeepAgentの有効性は、8つの多様なベンチマークにおける広範な実験によって実証されています。ここでは、主要な結果とその背景にある要因を分析します。

6.1 一般的なツール使用タスク (General Tool Usage)

ToolBench、API-Bank、TMDB、Spotify、ToolHopなどのデータセットにおいて、DeepAgentはベースライン手法（ReAct, CodeActなど）を一貫して上回りました。

表1: 一般ツール使用タスクにおける**Pass@1**成功率（%）の比較 [1]

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

*(注: ToolHopの*ReAct*スコアは関連資料からの推定値を含む比較)*

- 分析:
  - オープンセットでの優位性: ToolBenchやToolHopのような、数千のツールから適切なもの

を選択しなければならないタスク（Open-set retrieval）において、DeepAgentはReActやCodeActを大きく引き離しています。これは、DeepAgentの動的ツール検索メカニズムが機能している証左です。
    - 複雑なクエリへの対応: Spotifyタスク（プレイリスト操作など）において、DeepAgentは92.0%という圧倒的な成功率を記録しました。対してReActは45.5%にとどまっています。この差は、DeepAgentが複数のAPI（検索、再生、リスト追加）を論理的に組み合わせる能力に長けていることを示唆しています [4]。

6.2 ダウンストリーム・アプリケーションと実世界タスク

より長期的で複雑なインタラクションを要するアプリケーションタスク（ALFWorld, WebShop, GAIA, HLE）における評価です。

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 [1]

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

- **GAIA (General AI Assistants benchmark):** GAIAは「汎用AIアシスタント」の能力を測るための難関ベンチマークであり、推論、ツール使用、マルチモダリティが求められます。DeepAgentはここで53.3のスコアを記録し、既存のSOTAであるHiRA (42.5) を大きく上回りました。GAIAのテキスト専用サブセットでは、DeepSeek-R1 (43.7%) をも凌駕する58.3%を記録しています [9]。
- **WebShop:** オンラインショッピングのシミュレーションにおいて、DeepAgentはCodeAct (18.0%) の約2倍となる34.4%の成功率を達成しました。WebShopは商品検索、オプション選択、購入手続きといった多段階のプロセスを必要としますが、DeepAgentのメモリ管理機能が、長いセッション中での文脈維持に寄与していると考えられます。
- **Humanity's Last Exam (HLE):** 最も難易度が高いとされるHLEベンチマークにおいても、DeepAgentはワークフロー型エージェントよりも高いスコアを記録しており、AGIに向けた汎用推

論能力の高さが示されています [7]。

6.3 アブレーション研究：各コンポーネントの貢献度

DeepAgentの性能がどの機能に由来するのかを検証するアブレーション研究の結果は、メモリ畳み込みの重要性を浮き彫りにしています。
特にGAIAタスクにおいて、メモリ畳み込み機構（Memory Folding）を無効化すると、スコアは53.3から44.7へと8.6ポイントも急落しました 9。これは、長期的なタスクにおいて、情報を構造化して圧縮・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ます。また、ToolPOによる強化学習も、ベースモデル（Base）からRLモデルへのスコア向上（GAIAで+6.6、ALFWorldで+3.7）に明確に寄与しており、学習戦略の有効性が確認されました 3。

7. 議論と結論

7.1 学術的・産業的意義

本研究は、中国人民大学（RUC）の高知能自然言語処理研究所（RUC-NLPIR）と、Xiaohongshu（小紅書）の共同研究成果です。RUC-NLPIRは情報検索と生成AIの融合において世界的な実績を持つ研究グループであり、Xiaohongshuは膨大なユーザー行動データと実世界アプリケーションの知見を持つ企業です 7。
この産学連携により、DeepAgentは「理論的なエレガンス（脳模倣型メモリ）」と「実用的なスケーラビリティ（16k API対応、低コストなToolPO）」を高い次元で両立させることに成功しました。特にXiaohongshuは、DeepAgent以外にも視覚情報を用いた思考エージェント「DeepEyes」などの研究も進めており、今回の成果はマルチモーダルな自律エージェントへの発展に向けた重要な布石となるでしょう 13。

7.2 エージェントアーキテクチャの未来

DeepAgentが示した「統合的推論（Unified Reasoning）」と「構造化された記憶（Structured Memory）」は、今後のAIエージェント開発の標準的な指針となる可能性があります。

- **System 2への進化:** 現在のLLMは直感的な応答（System 1）に優れていますが、DeepAgent

のようなアーキテクチャは、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ています。
- ハードウェアとコスト: メモリ畳み込みは、無限に増大するコンテキスト長に対する現実的な解ですが、その実行には補助モデルによる追加計算が必要です。今後は、このオーバーヘッドを最小化するための専用ハードウェアや、より効率的なモデル蒸留技術が求められるでしょう。

7.3 結論

DeepAgentは、従来のエージェントが抱えていた「硬直したワークフロー」と「コンテキストの爆発」という二重の課題に対し、アーキテクチャレベルでの根本的な解決策を提示しました。16,000以上のツールを動的に発見・活用し、長期間にわたるタスクの文脈を自律的なメモリ操作によって維持するその能力は、既存の最先端モデルを凌駕するものです。この研究は、AIが単なる「チャットボット」から、現実世界で実質的な労働を担う「真の代理人（Agent）」へと進化するための、極めて重要なマイルストーンであると結論付けられます。

免責事項: 本報告書は、2025年10月に発表された公開論文および関連資料に基づき作成されています。DeepAgentのコードおよびデモは、GitHubリポジトリ（https://github.com/RUC-NLPIR/DeepAgent）にて公開されています。

引用文献

1. DeepAgent: A General Reasoning Agent with Scalable Toolsets - arXiv, 11月 25, 2025にアクセス、https://arxiv.org/html/2510.21618v1
2. [2510.21618] DeepAgent: A General Reasoning Agent with Scalable Toolsets - arXiv, 11月 25, 2025にアクセス、https://arxiv.org/abs/2510.21618
3. DeepAgent: A General Reasoning Agent with Scalable Toolsets - arXiv, 11月 25, 2025にアクセス、https://arxiv.org/pdf/2510.21618
4. Chinese researchers ship agent that finds tools mid-thought, beating OpenAI's rigid workflows - Implicator.ai, 11月 25, 2025にアクセス、https://www.implicator.ai/chinese-researchers-ship-agent-that-finds-tools-mid-thought-beating-openais-rigid-workflows/
5. DeepAgent: A General Reasoning Agent with Scalable Toolsets (Oct 2025) - YouTube, 11月 25, 2025にアクセス、https://www.youtube.com/watch?v=odpbuzD6ZDo
6. RUC-NLPIR/DeepAgent: 🛠️ DeepAgent: A General ... - GitHub, 11月 25, 2025にアクセス、https://github.com/RUC-NLPIR/DeepAgent
7. DeepAgent: A Deep Reasoning AI Agent that Performs Autonomous Thinking, Tool Discovery, and Action Execution within a Single Reasoning Process - MarkTechPost, 11月 25, 2025にアクセス、https://www.marktechpost.com/2025/11/01/deepagent-a-deep-reasoning-ai-age

nt-that-performs-autonomous-thinking-tool-discovery-and-action-execution-within-a-single-reasoning-process/
8. DeepAgent: A General Reasoning Agent with Scalable Toolsets - ChatPaper, 11月 25, 2025にアクセス、https://chatpaper.com/paper/203005
9. DeepAgents: AI agents just got a massive upgrade - Medium, 11月 25, 2025にアクセス、https://medium.com/@suchitraidumina/deepagents-ai-agents-just-got-a-massive-update-013765d8056f
10. Deep Agent | PDF | Cognitive Science - Scribd, 11月 25, 2025にアクセス、https://www.scribd.com/document/943049836/Deep-Agent
11. Xiaoxi Li - PhD student, Renmin University of China - OpenReview, 11月 25, 2025にアクセス、https://openreview.net/profile?id=~Xiaoxi_Li4
12. Team - RUC-NLPIR, 11月 25, 2025にアクセス、https://ruc-nlpir.github.io/team/
13. Discover, Discuss, and Read arXiv papers - Explore | alphaXiv, 11月 25, 2025にアクセス、https://www.alphaxiv.org/?organizations=Xiaohongshu+Inc.&datasets=true