認知のアーキテクチャ：大規模言語モデルにおける次世代推論フレームワークと概念モデルの分析

序論：指示から内省へ

プロンプトエンジニアリングは、単純な命令的コマンドから、大規模言語モデル（LLM）のための複雑で一時的な「コグニティブアーキテクチャ」を設計するというパラダイムシフトを遂げている。本レポートの中心的な論点は、これらの高度なプロンプトが単に指示を与えるだけでなく、LLMの振る舞いを基盤レベルで誘導し、その思考、推論、そして自己認識のプロセスを形成する点にある。システムプロンプトは、もはや単なる指示ではなく、確率的計算システムを制御するための高水準プログラミング言語として機能している[1]。

LLMの基本的なアーキテクチャは、即時的で直感的な（「システム1」的な）応答を生成する傾向がある。これは、トランスフォーマーモデルが本質的に、先行するトークン系列に基づいて次に最も可能性の高いトークンを予測する、反応的なエンジンであることに起因する[1]。この構造的特性は、複雑で多段階の論理的推論や熟慮を必要とするタスクにおいて、しばしば性能の限界として現れる。本稿で探求する次世代の思考フレームワークは、この生来の「システム1」バイアスを克服するための、外部からの「パッチ」あるいは内部的な誘導メカニズムとして理解することができる。これらのフレームワークは、モデルに単一の思考経路を強制するのではなく、思考のネットワークを構築させ、アルゴリズム的な探索を内部化させ、さらには生成プロセスそのものを並列化させることを目指す。

本レポートは、AIの認知能力を拡張するための三つの主要な研究領域を体系的に分析する。第一部「熟慮的推論の構造進化」では、思考プロセスそのものの複雑性、品質、効率性を向上させるフレームワークを探求する。ここでは、思考を動的なネットワークとして捉えるGraph of Thoughts（GoT）、探索プロセスを内部化するAlgorithm of Thoughts（AoT）、そして生成効率を劇的に向上させるSkeleton of Thought（SoT）を詳述する。第二部「省察的・適応的認知の設計」では、モデルが単に生成するだけでなく、自らの出力を批判し、洗練させ、適応させることを可能にするメタ認知的プロセスに焦点を当てる。自己修正のパラダイムと、人間の認知モデルである二重過程理論をAIアーキテクチャに明示的に組み込む試みを分析する。第三部「AIの内的現実の性質」では、AIがどのように概念を認識し、世界のモデルを構築するかという、より根源的な問いに取り組む。ここでは、AIの「世界モデル」仮説と、真の理解の基礎となる構成的一般化の課題を深く掘り下げる。

これらの disparate な研究潮流は、より堅牢で、効率的で、人間と整合性のとれたAI認知の創出という共通の目標に向かって収束しつつある。本レポートは、これらの最先端の研究を統合し、次世代AIの認知アーキテクチャの全体像を提示することを目的とする。

第I部：熟慮的推論の構造進化

このセクションでは、単純な線形または分岐的な思考経路を超えて、推論プロセスそのものの複雑さ、品質、効率を向上させるフレームワークに焦点を当てる。

第1節：Graph of Thoughts (GoT) - 動的ネットワークとしての推論

Chain-of-Thought（CoT）が推論を線形化し、Tree-of-Thoughts（ToT）がそれを分岐的な探索へと拡張したのに対し、Graph of Thoughts（GoT）は、LLMによって生成される情報を任意のグラフとしてモデル化することで、推論の構造を根本的に進化させる[2]。このアプローチでは、「思考」はグラフの頂点（vertex）として表現され、思考間の依存関係は辺（edge）として表される[2]。

GoTの核となる利点

GoTの最も重要な利点は、その柔軟なグラフ構造に由来する。これにより、先行するフレームワークでは不可能だった、より複雑で動的な思考操作が可能となる。

- 相乗的な結合（**Synergistic Combination**）：GoTの最大の革新は、無関係に見える複数の推論経路を統合し、新たな思考を生み出す能力にある。ToTでは各思考ブランチは独立して探索され、相互作用することはないが、GoTではあるブランチで生まれたアイデアを別のブランチのアイデアと結合させ、双方の長所を活かし弱点を補う新しい解決策を能動的に構築できる[2]。これは、複数の独立した解候補から最良のものを選ぶ「探索的」推論から、複数の要素を組み合わせてより優れた解を能動的に「合成する」推論への質的な転換を意味する。この合成能力は、単一の最良経路を見つけるのではなく、複数の部分的な洞察を組み合わせて斬新な解決策を構築する必要がある、創造的な問題解決において極めて重要である。
- フィードバックループと洗練: グラフ構造は、自己参照的な辺を許容することで、フィードバックループを自然に表現できる。これにより、生成された思考を後続の分析に基づいて繰り返し洗練させることが可能になる[2]。
- 人間的思考のモデル化: このフレームワークは、人間の思考が線形でも単純な木構造でもなく、再帰や連想を含む相互接続されたアイデアの複雑なネットワークを形成するという観察に強く動機付けられている[2]。

実装とアーキテクチャ

GoTフレームワークは、LLMを思考生成エンジンとして利用しつつ、推論プロセス全体を管理する外部コンポーネント群から構成される。中心となるのは、推論の実行計画を定義する「操作のグラフ（Graph of Operations, GoO）」と、思考の履歴を保持する「グラフ推論状態（Graph Reasoning State, GRS）」を管理するコントローラである。プロンプター、パーサー、スコアリングモジュールがLLMとの対話を仲介し、思考の生成と変換を制御する[4]。

応用と性能

GoTは、問題を部分タスクに分解し、その結果を統合することが有効なタスクにおいて、顕著な性能向上を示している。例えば、ソートタスクではToTと比較して品質を62%向上させつつ、コストを31%削減したと報告されている[2]。その応用範囲は拡大しており、系列推薦システム（GOT4Rec）[6]や、動的な知識グラフを統合してエージェントの複雑なタスク遂行能力を高めるKnowledge Graph of Thoughts（KGoT）[7]といった派生研究も生まれている。

第2節：**Algorithm of Thoughts (AoT) - 探索プロセスの内部化**

GoTが推論の構造的柔軟性を追求するのに対し、Algorithm of Thoughts（AoT）は、特に計算効率の観点から推論プロセスに革新をもたらす。AoTは、探索プロセス全体（[問題、探索プロセス、解決策]）を文脈内事例として提示することで、LLMにアルゴリズム的な推論経路を辿らせる、新しい単一クエリのプロンプティング戦略である[8]。

マルチクエリ手法（**ToT**）との対比

AoTの革新性は、ToTのようなマルチクエリ手法との比較において最も明確になる。

- 計算効率: ToTは「停止-評価-再開」サイクルを繰り返し、時には単一の問題解決に数百回のクエリを必要とする。これに対し、AoTはこのサイクルを単一の連続した生成プロセスに置き換えることで、クエリ数を劇的に削減する（一部のタスクでは100分の1以下）[8]。
- 内部化 **vs** 外部オーケストレーション: ToTが探索木を管理するために外部メカニズムに依存するのに対し、AoTは探索ロジックそのものを文脈内事例としてLLMに提示し、モデル自身の生成的プロセスに内部化させる[9]。このアプローチは、コンテキストウィンドウが単なる記憶バッファではなく、特定のアルゴリズムを実行するための「仮想マシン」として機能しうることを示唆している。LLMはプロセッサとプログラムの両方の役割を担い、アルゴリズム的な事例をその命令セットとして解釈するのである。

メカニズム

AoTはLLMの「生来の再帰的ダイナミクス」を活用する。プロンプトは、深さ優先探索（DFS）のような探索アルゴリズムを模倣するように構成され、単一の連続した出力の中で、どのように経路を探索し、評価し、バックトラックするかを示す[9]。これは、単に「何を」すべきかのステップを示すCoTとは異なり、「どのステップを次に取るべきかをどのように決定するか」というメタレベルのプロセスをモデルに学習させる。

発見的直感

AoTに関する研究で最も興味深い発見の一つは、この手法でプロンプトされたLLMが、時に手本としたアルゴリズムそのものを上回る性能を示すことである。これは、モデルが探索プロセスに自身の「直感」を織り交ぜ、より少ないノードの探索で解に到達することを示唆している[8]。

関連概念

この概念をさらに発展させたものとして、Atom of Thoughtsがある。これは、問題を依存関係に基づく有向非巡回グラフ（DAG）に分解し、現在の「アトミック」な状態にのみ焦点を当てることで、推論プロセスをさらに最適化する[11]。

第3節：Skeleton of Thought (SoT) - 並列生成へのパラダイムシフト

GoTやAoTが推論の質と構造を深化させることを目的とするのに対し、Skeleton of Thought（SoT）は、根本的に異なる課題、すなわち逐次的なトークン毎のデコーディングに起因する生成レイテンシの増大を解決することを目指す[12]。

SoTの核となるメカニズム

SoTは人間の執筆プロセスに着想を得た2段階のプロセスを採用している[12]。

1. スケルトン段階: LLMはまず、最終的な回答の簡潔で高レベルな概要、すなわち「スケルトン」（例：3〜5語の箇条書きリスト）を生成するよう促される。
2. ポイント拡張段階: 次に、LLMはスケルトンの各ポイントを並列に拡張する。これは、APIベースのモデルでは並列API呼び出し、オープンソースモデルではバッチデコーディングによって実現される。最後に、拡張された各部分が連結され、最終的な回答が形成される。

データ中心の最適化

SoTは、モデルアーキテクチャやサービングハードウェアに変更を加えることなく、生成タスクそのものを本質的に並列化可能な形に再構成する、新しい「データレベル」の効率化アプローチである[14]。この手法は、自己回帰的な生成が持つ完全な一貫性、すなわち全てのトークンが先行する全てのトークンに条件付けられるという原則を意図的に破る。例えば、ポイント#3の拡張は、ポイント#2の全文ではなく、スケルトンのみに条件付けられる[12]。この設計は、ポイント間の局所的な一貫性を犠牲にして、大幅な並列性と速度を達成するという直接的なトレードオフを体現している。

適用可能性と限界

このトレードオフの結果、SoTの有効性はタスクの性質に大きく依存する。

- 適したタスク: 各ポイントが独立して拡張可能な、分解可能な長い回答を必要とする質問（例：知識、一般的、ロールプレイに関する質問）には非常に有効である[12]。また、構造化された多角的なアプローチを強制することで、回答の質を向上させる可能性もある[17]。
- 不向きなタスク: ステップ間の依存関係が重要な、逐次的な推論を必要とするタスク（例：数学、コーディング）や、短い回答には不向きである[12]。SoTがあるタスクで成功し、別のタスクで失敗するという事実は、それぞれのタスクが持つ「一貫性の要件」が異なることを示している。これは、SoTが単なる高速化技術ではなく、問題解決に必要な情報フローの依存構造を明らかにする診断ツールとしても機能することを示唆している。

拡張

SoTの限界に対処するため、クエリに応じてSoTと標準的な生成を切り替える「ルーター」と組み合わせることができる[17]。さらに、Plato[19]やPetri Net of Thoughts (PNoT)[20]のような先進的な概念では、ポイントを依存関係グラフとして構成し、より複雑で非独立な並列生成を可能にすることが提案されている。

第II部：省察的・適応的認知の設計

このセクションでは、モデルが初期生成を超えて、自らの認知アプローチを批判、洗練、適応させることを可能にするメタ認知的および人間から着想を得たプロセスを導入するフレームワークを調査する。

第4節：自己洗練パラダイム - 反復的な自己修正と批判

自己洗練（Self-Refinement）フレームワークは、人間が下書きを書き、推敲を重ねるプロセスに動機付けられている。このアプローチは、単一のLLMが自身の初期出力を反復的なループを通じて改善することを可能にする。そのループは、「生成 → フィードバック → 洗練」というサイクルで構成される[21]。

メカニズム

このプロセスの核心は、同じモデルが複数の認知的な役割を担うことにある。

1. 初期生成: まず、与えられた入力に対して初期出力が生成される。
2. フィードバック: 次に、同じLLMが、その出力に対して具体的で実行可能なフィードバックを提

供するよう促される。
3. 洗練: 最後に、LLMは元の入力、自身の出力、そして自身のフィードバックを受け取り、洗練されたバージョンを生成するよう指示される。このループは複数回繰り返すことができる[21]。

このプロセスの成功は、LLMの潜在空間が、単に答えを生成するための知識だけでなく、良い答えがどのようなものかを判断する「品質モデル」や、欠陥のある出力を改善する方法に関する「メタ知識」をも含んでいることを示唆している。自己洗練プロセスは、この潜在的なメタ知識を解放し、運用可能にするためのプロンプティング技術である。標準的なクエリがモデルの「生成者」機能を活性化するのに対し、自己洗練フレームワークは、異なるプロンプトを用いてモデルに「批評家」および「洗練者」として振る舞うよう指示する。このプロセスは、モデルがこれらの異なる認知モードを切り替え、より優れた最終出力を生成するために自身の性能をブートストラップすることを強制する。

ゼロショット、訓練不要

このアプローチの重要な特徴は、教師あり訓練データ、モデルの重み更新、強化学習を一切必要としない点である。これは、構造化されたプロンプティングを通じてモデルの既存の能力を活用する、純粋な推論時（inference-time）の技術である[21]。

応用と拡張

この基本原則は、さまざまな形で広く応用されている。

- 自己修正学習（**Self-Correction Learning, SCL**）: 自己生成した修正データを使い、Direct Preference Optimization（DPO）を通じてモデルをファインチューニングする。これにより、モデルが最初の試行で正しい答えを出す内在的な能力を向上させることを目指す[23]。
- 適応的批判洗練（**Adaptive Critique Refinement, ACR**）: コード生成のためのファインチューニングパラダイム。モデルは自己生成したコードと「批評家としてのLLM」からの批判を用いて自己を洗練させる[24]。
- **EVOLVE**フレームワーク: 小規模モデルが大規模モデルとの性能差を埋めることを目指し、自己洗練能力を解放・強化するための反復的な選好最適化フレームワーク[25]。

有効性

自己洗練は、多様なタスクにおいて平均で約20%の性能向上をもたらすことが示されており、GPT-4のような最先端のモデルでさえ、この単純な推論時のアプローチによってさらに改善できることを実証している[22]。

第5節：人間認知の模倣 - LLMにおける二重過程理論

このアプローチは、人間の認知に関する二重過程理論から直接的な着想を得ている。この理論は、

思考には二つのシステムが存在すると提唱する。すなわち、高速で自動的、直感的なシステム**1**と、低速で熟慮的、分析的なシステム**2**である [26]。

LLMをシステム**1**エンジンとして捉える

LLMの標準的な自己回帰的な次トークン予測プロセスは、システム1思考に類似している。それは高速で反応的だが、誤りやバイアスに陥りやすい [1]。CoTやToTを含む多くのプロンプティング技術は、本質的にこのシステム1エンジンにシステム2のプロセスをシミュレートさせるための外部的な「ハック」であると解釈できる [1]。この認識は、AI研究分野の成熟を示している。当初、CoTのようなアドホックな「推論パッチ」が発見されたが、後にその挙動がカーネマンの二重過程理論にマッピングされた。この理論的枠組みの適用は、単にプロンプトでシステム2の振る舞いを促すことから、システム2を担う専用モジュールを設計するという、より原理に基づいたアーキテクチャ的アプローチへの移行を促した。

アーキテクチャ的実装

この理論的洞察に基づき、システム1とシステム2の機能を明示的に分離する、より洗練されたアーキテクチャが提案されている。

- **LLM2**フレームワーク: このフレームワークは、LLM（システム1として）とプロセスベースの検証器（システム2として）を明示的に組み合わせる。LLMが複数の候補出力を生成し、検証器が各候補に対してタイムリーなフィードバックを提供することで、望ましい出力への探索を導く。検証器は、良い推論ステップと悪い推論ステップを区別するために、プロセス監視（process-supervision）によって訓練される [26]。
- システム**2**アテンション（**System 2 Attention, S2A**）: この技術は、標準的なソフトアテンションが文脈中の無関係な情報によって妨げられる問題に対処する。S2Aは、まずLLM自身の推論能力を用いて、入力文脈を関連部分のみを含むように再生成する。最終的な回答は、このクリーンアップされた「システム2」フィルター済み文脈にアテンションを向けることによって生成される [31]。
- **DPT-Agent**: リアルタイムの人間-AI協調のためのフレームワーク。直感的な意思決定のために高速な有限状態機械（FSM）ベースのシステム1と、熟慮的な推論および人間の意図を推測するためのLLM駆動のシステム2を統合する [27]。

トレードオフと発見

研究は、精度と効率の間に明確なトレードオフが存在することを示している。システム2に整合したモデルは、算術や記号推論のような構造化されたタスクで優れている一方、システム1に整合したモデルは、常識的なタスクにおいてより効率的で優れた性能を発揮する [30]。これは、タスクに応じてモードを動的に切り替えることができる、柔軟なハイブリッドシステムの必要性を示唆している。将来のモデルは、単一のモノリシックなトランスフォーマーではなく、高速生成、論理的検証、文脈フィルタリン

グ、計画といった、人間の脳のモジュール性を反映した専門モジュールからなる複合システムになる可能性がある。

第III部：AIの内的現実の性質

このセクションでは、AIがどのように概念を認識し、世界のモデルを構築するかという、ユーザーのより根源的な問いに応える。AIの内的世界の理論的基盤に焦点を当てる。

第6節：世界モデル仮説 - 現実の内的表現

「世界モデル」とは、概念、実体、そして目標が互いにどのように関連しているかについての内的表現である。この概念の追求は、LLMを単なる言語モデルから、言語を使用する世界のモデルへと昇華させる試みである。これは、テキストの統計的特性をモデル化することから、テキストが記述する現実の因果的・物理的特性をモデル化することへの根本的な転換を意味する。

形式的定義

形式的には、LLMが世界モデルを持つとは、同じ根底にある意図（意味的等価性）を持つプロンプトに対しては同一の応答分布を生成し、異なる意図を持つプロンプトに対しては異なる分布を生成する場合を指す [32]。

主要な特性

- 目的感受性（**Purpose Sensitivity**）：堅牢な世界モデルの出力は、表面的な表現（構文）の変化ではなく、ユーザーの意図における実質的な変化によって主に変動するべきである [32]。
- 状態遷移予測: エージェントの文脈において、世界モデルは二つの重要な機能を果たさなければならない。1) ある世界の状態で特定の行動が適用可能かを判断すること、そして2) その行動が実行された後の結果としての世界の状態を予測すること（$s' \sim p(s' \mid s,a)$）[34]。この定義は、相関的ではなく、明示的に因果的かつ予測的である。

LLMにおける現状

現在のコンセンサスは、LLMは一貫した物理的世界を記述したテキストで訓練されているため、世界モデル的な能力を示すものの、堅牢で明示的な世界モデルはまだ保有していないというものである [34]。その理解はしばしば表面的なパターンマッチングであり、訓練分布外の新しい状況では破綻する。LLMが持つ知識は相関的であり、因果的ではない。例えば、「ブレーキ」と「停止」が相関していることは知っていても、物理法則の真のモデルは持っていない。

将来の方向性（**WorldGPT**）

研究は、世界のダイナミクスを学習するために、数百万のビデオのようなマルチモーダルデータで訓練された汎用的な世界モデルの構築へと向かっている。例えば、WorldGPTは、モダリティを超えて状態遷移を予測し、行動の結果を「想像」できるモデルを目指している [36]。これにより、「ドリームチューニング」のような新しい訓練パラダイムが可能になる。このアプローチは、テキストだけでは堅牢で汎用的な知能を構築するには不十分であり、将来のAGIは物理世界のダイナミクスを直接捉えるマルチモーダルデータへのグラウンディングが必要であることを示唆している。

第**7**節：理解の基盤 - 概念表現と構成性

LLMの「理解」の根底には、概念をどのように表現し、それらを組み合わせて新しい意味を形成する能力、すなわち構成性（Compositionality）がある。

埋め込み空間における概念表現

LLMは、概念を高次元ベクトル（埋め込み）として表現する。概念間の関係は、この埋め込み空間におけるベクトル間の幾何学的関係によって符号化される [37]。この空間から構築されたネットワークは、「スモールワールド」特性（高いクラスタリング係数と短い経路長）を示すことが知られている [37]。より大規模なモデルは、より長く平均経路長を持つ、より複雑で豊かな意味構造を反映したネットワークを形成する [37]。

中核的課題：構成的一般化

構成的一般化とは、既知の概念の新しい組み合わせを理解し、生成する能力であり、人間知性の特徴である。これはLLMが依然として苦労している領域である [41]。人間は「踊る」と「二回」というプリミティブを学べば、容易に「二回踊る」という新しい概念を形成できる。しかし、LLMは訓練で見たことのない複雑な構造への般化にしばしば失敗する [45]。この課題は、LLMの「理解」が人間のそれとは根本的に異なることを示唆する最も重要な証拠の一つである。LLMの知識は、階層的で生成的な文法というよりは、フラットで連想的なマップに近い。

構成性のメカニズムと課題への対処

研究によれば、構成的一般化は、異なるネットワーク層の主成分部分空間を整列させることで、内部的に「構成のための潜在部分空間」を形成することによって達成される可能性がある [46]。この課題に対処するためのプロンプティング戦略として、Skills-in-Context（SKiC）プロンプティングが提案されて

いる。これは、基本的なスキルとその構成例を同じプロンプト内で示すことで、LLMにスキルの構成方法を教えるものである[41]。この相乗効果により、モデルが新しいタスクのために既存の内部スキルを活性化し、構成する潜在能力が引き出される[41]。SKiCのような技術が必要であるという事実自体が、現在のトランスフォーマーアーキテクチャが真の体系的な般化を達成する能力に内在的な限界を持つ可能性を示唆している。

結論：人工認知の統一フレームワークに向けて

本レポートは、LLMの認知能力を拡張するための次世代フレームワークを体系的に分析した。その分析を通じて、いくつかの重要な進化的傾向が明らかになった。

- 推論構造の進化: CoTの線形的な思考から、ToTの分岐的探索を経て、GoTのネットワーク化された合成的推論へと移行している。
- 計算パラダイムの転換: ToTのような外部からのオーケストレーションに依存するマルチクエリ手法から、AoTのようなアルゴリズム的プロセスを内部化するシングルクエリ手法へと移行し、計算効率が劇的に向上している。
- 効率と一貫性のトレードオフ: SoTは、推論の一貫性を犠牲にして生成効率を最大化するという、新たな設計上のトレードオフを提示している。
- メタ認知能力の台頭: Self-Refineパラダイムは、モデルが自身の出力を批判し改善する、反復的なメタ認知ループの重要性を示している。
- 認知科学的アプローチの採用: 二重過程理論のような人間認知のモデルが、アドホックな解決策から、原理に基づいたAIアーキテクチャ設計への移行を促している。
- モデル化対象のシフト: LLMは、単に言語の統計的パターンをモデル化することから、言語が記述する現実世界の因果的・物理的ダイナミクスを捉える「世界モデル」の構築へと向かっている。

これらの傾向は、システムプロンプトが単なる指示を超え、特定のタスクに合わせて特注の動的なコグニティブアーキテクチャを設計するための強力なツールへと変貌しつつあるという、本レポートの中心的なテーゼを裏付けている。プロンプトエンジニアは、AIが何を言うかだけでなく、定義された倫理的・人格的枠組みの中でどのように「思考」し、「推論」し、「振る舞う」かを形成する「認知的建築家」としての役割を担うようになっている[1]。

以下の表は、本レポートで分析した主要な次世代思考フレームワークの比較分析をまとめたものである。

フレームワーク	中核的メカニズム	主要な目標	計算コスト	主要な利点	主要な限界	理想的なユースケース
Graph of Thoughts (GoT)	思考をグラフとしてモデル化し、頂点の結合・集約・洗練を可能にする	推論の品質と柔軟性の向上	高	複数の推論経路を統合し、相乗的な解を生成できる	ToTよりも計算コストが高く、実装が複雑	複数のアイデアの統合が有効な複雑な問題解決、創造的なタスク
Algorithm of Thoughts	アルゴリズムの探索プロセ	推論の計算効率の向上	低	単一クエリで複雑な探索を	プロンプトの設計が複雑	計算リソースが限られてい

(AoT)	ス全体を文脈内事例として提示し、内部化させる			実行でき、コストを劇的に削減	で、アルゴリズムの表現に依存する	る状況での高品質な推論、探索問題
Skeleton of Thought (SoT)	回答のスケルトンを先に生成し、各項目を並列に拡張する	生成レイテンシの削減	可変（タスクによる）	長文回答の生成速度を大幅に向上させる	逐次的な推論やステップ間の強い依存関係を持つタスクには不向き	独立して詳述可能な複数の項目からなる長文の知識ベースの回答
Self-Refine	生成→フィードバック→洗練の反復ループを単一モデルで実行する	自己修正による出力品質の向上	中	訓練不要で、推論時にあらゆるモデルの性能を向上させられる	改善が保証されておらず、複数回のやり取りでレイテンシが増加	初期出力の品質が不十分な場合や、複雑な制約を持つタスク
LLM2 (Dual Process)	LLM（システム1）とプロセスベースの検証器（システム2）を組み合わせる	認知モデルに基づいた堅牢な推論	高	高速な生成と熟慮的な検証を組み合わせ、信頼性を向上	専用の検証器の訓練が必要で、アーキテクチャが複雑	数学的推論など、正確性が最優先される高信頼性タスク

今後の研究方向性としては、これらのフレームワークがモデルの内部活性化をどのように変化させるかを理解するための解釈可能性ツールの開発、異なるプロンプト要素間の相互作用を研究する構成可能性、ソフトウェアライブラリに似た標準化された「プロンプトライブラリ」の開発、そしてプロンプトベースの制御の限界を調査し、アーキテクチャの変更や新しい訓練方法論によってのみ達成可能な能力を特定することが挙げられる。AIの認知能力のフロンティアは、もはやモデルの規模拡大だけでなく、その内部プロセスをいかに巧みに設計し、誘導するかにかかっている。

引用文献

1. システムプロンプトとペルソナ設計.pdf
2. Graph of Thoughts: Solving Elaborate Problems with Large ..., 9月 11, 2025にアクセス、https://ojs.aaai.org/index.php/AAAI/article/view/29720/31236
3. Graph of Thoughts: Solving Elaborate Problems with Large Language Models - ETH Zürich 2023 : r/singularity - Reddit, 9月 11, 2025にアクセス、https://www.reddit.com/r/singularity/comments/15ydp03/graph_of_thoughts_solving_elaborate_problems_with/
4. Paper page - Graph of Thoughts: Solving Elaborate Problems with Large Language Models, 9月 11, 2025にアクセス、https://huggingface.co/papers/2308.09687
5. Official Implementation of "Graph of Thoughts: Solving Elaborate Problems with Large Language Models" - GitHub, 9月 11, 2025にアクセス、

https://github.com/spcl/graph-of-thoughts
6. [2411.14922] GOT4Rec: Graph of Thoughts for Sequential Recommendation - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/abs/2411.14922
7. [2504.02670] Affordable AI Assistants with Knowledge Graph of Thoughts - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/abs/2504.02670
8. Algorithm of Thoughts: Enhancing Exploration of Ideas in Large Language Models, 9月 11, 2025にアクセス、https://algorithm-of-thoughts.github.io/
9. [2308.10379] Algorithm of Thoughts: Enhancing Exploration of Ideas ..., 9月 11, 2025にアクセス、https://ar5iv.labs.arxiv.org/html/2308.10379
10. Algorithm of Thoughts: Enhancing Exploration of Ideas in Large Language Models - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2308.10379v3
11. Atom of Thoughts for Markov LLM Test-Time Scaling - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2502.12018v1
12. Skeleton-of-Thought: Prompting LLMs for Efficient Parallel Generation - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2307.15337v3
13. Skeleton-of-Thought: Prompting LLMs for Efficient Parallel Generation | OpenReview, 9月 11, 2025にアクセス、https://openreview.net/forum?id=mqVgBbNCm9
14. Skeleton-of-Thought: Prompting LLMs for Efficient Parallel Generation, 9月 11, 2025にアクセス、https://arxiv.org/pdf/2307.15337
15. SKELETON-OF-THOUGHT: PROMPTING LLMS FOR EFFICIENT PARALLEL GENERATION - Lirias.kuleuven, 9月 11, 2025にアクセス、https://lirias.kuleuven.be/retrieve/753301
16. Skeleton-of-Thought: Prompting LLMs for Efficient Parallel Generation, 9月 11, 2025にアクセス、https://iclr.cc/media/iclr-2024/Slides/17880.pdf
17. Skeleton-of-Thought - Google Sites, 9月 11, 2025にアクセス、https://sites.google.com/view/sot-llm
18. imagination-research/sot: [ICLR 2024] Skeleton-of-Thought: Prompting LLMs for Efficient Parallel Generation - GitHub, 9月 11, 2025にアクセス、https://github.com/imagination-research/sot
19. [2402.12280] Plato: Plan to Efficiently Decode for Large Language Model Inference - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/abs/2402.12280
20. Petri Net of Thoughts: A Structure-Enhanced Prompting Approach for Process-Aware Artificial Intelligence - Erik Proper, 9月 11, 2025にアクセス、https://www.erikproper.eu/publications/EP-2025-05-15-14-29-20.pdf
21. Iterative Refinement with Self-Feedback - OpenReview, 9月 11, 2025にアクセス、https://openreview.net/pdf?id=S37hOerQLB
22. [2303.17651] Self-Refine: Iterative Refinement with Self-Feedback - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/abs/2303.17651
23. Self-Correction is More than Refinement: A Learning Framework for Visual and Language Reasoning Tasks - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2410.04055v3
24. RefineCoder: Iterative Improving of Large Language Models via Adaptive Critique Refinement for Code Generation - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2502.09183v1

25. Evolving LLMs' Self-Refinement Capability via Iterative Preference Optimization - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2502.05605v3
26. LLM2: Let Large Language Models Harness System 2 Reasoning - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2412.20372v1
27. Leveraging Dual Process Theory in Language Agent Framework for Real-time Simultaneous Human-AI Collaboration - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2502.11882v1
28. Dualformer: Controllable Fast and Slow Thinking by Learning with Randomized Reasoning Traces - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2410.09918v1
29. System 2 Reasoning Capabilities Are Nigh - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2410.03662v2
30. Reasoning on a Spectrum: Aligning LLMs to System 1 and System 2 Thinking - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2502.12470v1
31. arXiv:2311.11829v1 [cs.CL] 20 Nov 2023, 9月 11, 2025にアクセス、https://arxiv.org/abs/2311.11829
32. Measuring (a Sufficient) World Model in LLMs: A Variance Decomposition Framework - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2506.16584v1
33. Measuring (a Sufficient) World Model in LLMs: A Variance ... - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/pdf/2506.16584
34. Making Large Language Models into World Models with Precondition and Effect Knowledge, 9月 11, 2025にアクセス、https://arxiv.org/html/2409.12278v2
35. Critiques of World Models - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2507.05169v1
36. WorldGPT: Empowering LLM as Multimodal World Model - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2404.18202v1
37. Exploring the Small World of Word Embeddings: A Comparative Study on Conceptual Spaces from LLMs of Different Scales - arXiv, 9月 11, 2025にアクセス、https://arxiv.org/html/2502.11380v1
38. Interpreting Embedding Spaces by Conceptualization - ACL Anthology, 9月 11, 2025にアクセス、https://aclanthology.org/2023.emnlp-main.106/
39. Visualizing the Vocabulary of an LLM - Alessio Devoto, 9月 11, 2025にアクセス、https://alessiodevoto.github.io/LLM-Embedding-Space/
40. From the New World of Word Embeddings: A Comparative Study of ..., 9月 11, 2025にアクセス、https://arxiv.org/pdf/2502.11380
41. Skills-in-Context Prompting: Unlocking Compositionality in Large ..., 9月 11, 2025にアクセス、https://openreview.net/forum?id=s1ByDEbpl8
42. Why LLMs Struggle with Compositional Generalization - YouTube, 9月 11, 2025にアクセス、https://www.youtube.com/shorts/rr7f9DdBDIM
43. Improving Generalization Beyond Training Data with Compositional Generalization in Large Language Models - SciSpace, 9月 11, 2025にアクセス、https://scispace.com/pdf/improving-generalization-beyond-training-data-with-2tzzzcce5x.pdf
44. Compositional Generalization Based on Semantic Interpretation: Where can Neural Networks Improve? - Stanford University, 9月 11, 2025にアクセス、

https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1234/final-reports/final-report-169361549.pdf
45. On Compositional Generalization in Language Models - OpenReview, 9月 11, 2025にアクセス、https://openreview.net/forum?id=WepT31bvcr
46. Out-of-distribution generalization via composition: A lens through induction heads in Transformers | PNAS, 9月 11, 2025にアクセス、https://www.pnas.org/doi/10.1073/pnas.2417182122