設計によるコグニティブアーキテクチャ：大規模言語モデルのシステムプロンプトにおける抽象的人文科学的構成概念の有効性

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（例：「このテキストを要約せよ」）から、システムプロンプト内に複雑な認知的・倫理的フレームワーク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（大規模言語モデル）のために一時的かつ特注の「コグニティブアーキテクチャ」を効果的に構築し、その振る舞いを基盤レベルで誘導するという点にある。
システムプロンプトは、対話全体にわたる文脈、ルール、そしてペルソナを設定する最高権威の指示レイヤーとして定義される[1]。それは永続的なフレームワークとして機能し、モデルが後続のすべてのユーザーからの問い合わせを解釈する方法を形成する[3]。
本稿では、この新しいアーキテクチャ的アプローチの主要な柱として、思考法、論理的整合性、哲学的整合性、アイデンティティ構築、そして文体模倣という5つの中核領域を探求する。これらの要素を習得することが、信頼性が高く、ニュアンスに富み、整合性のとれたAIシステムを開発するために不可欠であると提言する。本レポートは、LLMの振る舞いの技術的基盤から始まり、これらの高度なプロンプト戦略の実践的および倫理的含意へと議論を構築していく。

第1章 プロンプト駆動制御のアーキテクチャ的基盤

1.1 テキストからトークンへ：LLMの入力層

LLMが自然言語の指示を処理する最初のプロセスはトークン化である。この段階で、入力テキストは機械が読み取り可能な単位へと分解される[4]。強調すべきは、モデルが人間のようにテキストを「読む」のではなく、一連のトークンとして処理するという点である。この機械的な現実が、すべてのプロンプトエンジニアリングの基礎をなしている。モデルはプロンプトを、全体的な文や質問としてではなく、一連のトークンとして解釈するのである[5]。

1.2 文脈埋め込みとアテンションメカニズム

トランスフォーマーモデルは、トークンのための高次元ベクトル表現（埋め込み）を生成する。特に自己アテンションメカニズムは、モデルが新しいトークンを生成する際に、システムプロンプトの指示を含むプロンプト内の異なるトークンの重要性を重み付けすることを可能にする[5]。システムプロンプトは、指示、ガイドライン、文脈情報の集合体として、AIが特定のパラメータ内で動作するためのフレームワークを提供する[1]。アテンションメカニズムは、これらの「フレームワーク」トークンが生成プロセス全体を通じて一貫して参照されることを保証する。

1.3 生成の確率論的性質

LLMは、その核心において、次に最も可能性の高いトークンを予測する確率論的エンジンである[4]。システムプロンプトの役割は、この確率分布を根本的に変化させ、望ましい思考スタイル、ペルソナ、または倫理的フレームワークと一致するトークンが選択される可能性を高めることにある。temperatureやtop_pといったパラメータは、このトークン選択のランダム性を直接制御し、開発者が決定論的でルールに従う振る舞いと、創造的で多様な出力とのバランスを調整することを可能にする[7]。

システムプロンプトは、一種の動的な「仮想ファインチューニング」メカニズムとして機能する。ファインチューニングは、特定のデータセットやタスクに合わせてモデルの振る舞いを調整するために、モデルの重みを物理的に変更するプロセスである[2]。これはリソースを大量に消費し、そのモデルバージョンにとっては永続的な変更となる。一方で、ルールを強制し[1]、役割を定義し[5]、複雑なタスクを誘導する[9]能力が示すように、巧みに作られたシステムプロンプトは、同様の行動的整合性を達成する。そのメカニズムは異なる（重みの変更ではなく、アテンションに基づく文脈の提供）ものの、特化した応答パターンという結果は類似している。システムプロンプトは、セッションの持続期間中、モデルの確率的出力の方向性を定める強力な文脈的フィールドを提供する。したがって、システムプロンプトは、実際のファインチューニングのコストや永続性を伴わずに特化の利点を提供する、動的でセッション固有の「仮想ファインチューニング」と概念化できる[10]。この見方は、システムプロンプトを単なる「指示」から、モデルの核となる振る舞いを一時的に修正する強力なツールへと再定義するものである。

第2章 構造化された認知の誘発：思考法の規定

2.1 熟慮的推論の必要性

LLMの基本的な限界は、即時的で直感的（システム1的）な応答を生成する傾向にあり、これが複雑で多段階の推論を必要とするタスクでの失敗につながることである[11]。

2.2 Chain-of-Thought（CoT）プロンプティング：推論プロセスの外部化

CoTは、モデルに「ステップバイステップで考える」よう明示的に指示するか、最終的な答えの前に推論プロセスを示す少数事例（few-shot）の例を提供することで機能する[5]。これにより、モデルは中間的な推論トークンを生成せざるを得なくなる。「ステップバイステップで考えましょう」という一文を含めるだけで、算術、常識、記号推論タスクにおけるゼロショット性能が大幅に向上することが示されている[11]。これらの中間トークンは「思考の足場」として機能し、モデルが複雑な問題をより単純な次トークン予測の連続に分解することを可能にする。これにより、単一の巨大な推論的飛躍に伴う認知的負荷が軽減される[15]。

2.3 高度な推論フレームワーク：Tree-of-Thought（ToT）とそれ以降

ToTはCoTを拡張し、モデルが複数の推論経路を同時に探求し、各経路の実行可能性を自己評価し、有望でないと判断した場合には後戻りしたり方向転換したりすることを可能にする[13]。これにより、直線的な思考プロセスから、熟慮的で分岐的な思考プロセスへと移行する。Self-Ask[14]やCognitive Verifierパターン[13]のような技術も、モデルに質問を分解させ、中間ステップを検証させることで、論理的な堅牢性をさらに高める。これらの高度な技術は、探索空間が広いタスクや、初期の仮定が間違っている可能性がある場合に、自己修正と探求のメカニズムを組み込むため、極めて重要である。

2.4 プロンプトによる推論の限界

これらの技術は強力であるものの、万能薬ではない。過度に長い、あるいは複雑な推論プロンプトは、小規模なモデルのパフォーマンスに悪影響を与える可能性があり、「認知的負荷」やコンテキストウィンドウの制限といった概念を示唆している[17]。特定の技術の有効性は、タスクに大きく依存し、変動する。例えば、Self-Consistencyは、複数のCoTチェーンを実行し、答えについて多数決を取ることで、一部のベンチマークでCoTを上回る性能を示すが、これは計算コストと引き換えに精度を向上させるものである[14]。

LLMにおけるプロンプトによる推論は、本質的に「システム1」エンジンに対するアーキテクチャ上の「パッチ」として機能している。基本的なトランスフォーマーアーキテクチャは、即時的な文脈に基づいて次のトークンを生成する、本質的に反応的で予測的なものである。これは、人間の思考における高速で直感的な「システム1」に類似している[18]。しかし、論理、数学、計画を必要とするタスクは、低速で熟慮的な「システム2」思考を要求する[18]。LLMには、ネイティブなシステム2処理モジュールが欠けている。CoTやToTのようなプロンプトは、モデルに推論の方法を教えているわけではない。むしろ、システム1エンジンに、そのプロセスのテキスト的な軌跡を生成させることで、システム2プロセスをシミュレートさせる外部的な「ハック」なのである。因果連鎖は次のようになる：プロンプトが推論ステップの生成を強制し、これらのステップがコンテキストウィンドウに追加され、アテンションメカニズ

ムがより豊かで構造化された文脈を利用して次のトークン（最終的な答えを含む）を予測する。モデルは本質的に、自身の出力をガイドとして利用している。これは根本的なアーキテクチャ上の限界を露呈している。推論を引き出すためにプロンプトに依存している現状は、将来のAIモデルが、テキスト生成を通じて推論をシミュレートするのではなく、熟慮的推論のための専門的で計算的に区別されたモジュールを組み込む方向に進化する必要がある可能性を示唆している。

第3章 論理的整合性の強化：誤謬回避と事実性のためのプロンプティング

3.1 LLMの論理の二重性：知識と応用の乖離

LLMは論理的誤謬に関する知識を持っており、明示的に指示されればそれを検出できる[19]。しかし、同時に、討論の場では誤謬を伴う議論に非常に影響されやすく、自らも誤謬を含むコンテンツを生成することがある[19]。これは、知識の表現とデフォルトの振る舞いとの間に重大なギャップがあることを示している。モデルは、広大でフィルタリングされていないインターネットデータで訓練されているため、論理的な議論と非論理的な議論の両方のパターンを学習している。システムプロンプトの役割は、「論理的」な経路を活性化させ、「誤謬的」な経路を抑制することにある。

3.2 クリティカルシンキングのためのプロンプティング戦略

システムプロンプトに「前提を批判せよ」「論理的誤謬を特定せよ」「議論の構造を評価せよ」といった指示を詳述することで、論理的な厳密性を向上させることができる[19]。さらに強力な手法として、異なる視点を持つ複数のLLMエージェントをインスタンス化し、それらに討論させ、最終的に「審判」エージェントが議論を評価するというものがある。この「心の社会（society of minds）」アプローチは、推論を改善し、誤謬を含む回答を減少させることが示されている[20]。

3.3 ハルシネーションとの戦い：事実性の課題

ハルシネーション（幻覚）は、時に事実の正確性を犠牲にして、もっともらしく一貫性のあるテキストを生成するように訓練された確率モデルの自然な副産物である[23]。人間からのフィードバックによる強化学習（RLHF）は、魅力的だが誤ったコンテンツに報酬を与えることで、この問題を悪化させることさえある[24]。この問題に対処する主要なアーキテクチャパターンが、検索拡張生成（RAG）である。RAGは、外部の知識ベースから関連性の高い最新情報を検索し、それをプロンプト内でLLMに文脈として提供する。プロンプトは、提供された情報に
のみ基づいて回答するようモデルに指示する[2]。これにより、モデルの応答が検証可能な事実に根差したものとなる。また、モデルに自身の回答を数回にわたって再評価・改善させる反復的洗練のプ

ロンプト技術は、特に事実抽出ステップと組み合わせることで、自己修正を強制し、真実性を高めることができる[27]。

3.4 安全性アラインメントの脆弱性

LogiBreakのようなジェイルブレイク攻撃は、アラインメントメカニズムがしばしば脆弱であることを示している。有害なプロンプトを形式論理のような異なるモダリティに変換することで、攻撃者は安全フィルターを回避できる。これは、意味的な意図は同じでも、トークンの分布が目新しいためである[28]。この事実は、安全性のためのアラインメントが、倫理原則の深い理解ではなく、表面的なパターン認識タスクであることが多いことを示唆している。堅牢なシステムプロンプトは、何を避けるべきかを述べるだけでなく、肯定的で建設的な振る舞いや推論プロセスを組み込む必要がある。
LLMの設計には、創造性と事実性との間に根本的な緊張関係が存在する。LLMの核となる目的は、もっともらしいトークン系列を予測することによって、新規で一貫性のあるテキストを生成することであり、これは本質的に創造的なプロセスである。一方、事実性の核となる目的は、既存の検証済み情報を逸脱なく再現することであり、これは本質的に制約された再現的なプロセスである。これら二つの目的は直接的な緊張関係にあり、一方を最適化すると他方が劣化する可能性がある。創造性を促すプロンプト（高いtemperature）はハルシネーションのリスクを高め、厳格な事実性を強制するプロンプト（RAG、低いtemperature）は創造的なポテンシャルを制限する。したがって、事実性のためのプロンプトエンジニアリングは、欠陥を「修正」することではなく、この根本的なトレードオフを管理することに他ならない。RAGのような技術は、この認識の表れであり、本質的にLLMに対して「創造的であることをやめ、私が与えるこの信頼できるデータの要約者になりなさい」と指示している。これは、単一のモノリシックなLLMが、創造的なタスクと事実に基づくタスクの両方で同時に優れた性能を発揮することが不適切である可能性を示唆している。将来のシステムは、推論された意図に基づいて、創造的な生成モデルまたは事実に根差したRAGモデルにクエリを振り分ける「専門家の混合（mixture of experts）」アプローチを採用するかもしれない。システムプロンプトは、適切な「専門家」を選択し、設定するための主要なメカニズムとして機能する可能性がある。

表1: 推論と事実性のための高度なプロンプティング技術の比較分析

技術	中核原理	主なユースケース	関連研究	指摘されている限界/トレードオフ
Chain-of-Thought (CoT)	推論を線形化し、ステップバイステップで外部化する	数学・論理問題、段階的な推論	[5]	小規模モデルでは失敗する可能性。誤りが連鎖するリスク。
Tree-of-Thought (ToT)	複数の推論経路を探求し、自己評価とバックトラックを行う	複雑な計画、探索空間の広い問題	[13]	CoTよりも計算コストが高い。実装が複雑。
Self-Consistency	複数の推論チェーンを生成し、最終回答を多数決で決定	高精度の推論、頑健性の向上	[14]	計算コストが非常に高い。多様な推論経路が必要。

	する			
Multi-Agent Debate	異なる視点を持つエージェントが討論し、審判が評価する	誤謬検出、バイアスの低減、複雑な意思決定	[20]	複数のモデル呼び出しが必要で高コスト。セットアップが複雑。
Retrieval-Augmented Generation (RAG)	外部の知識ベースから情報を検索し、文脈として提供する	事実に基づくQ&A、ハルシネーションの抑制	[2]	外部データベースの質と鮮度に依存。検索の失敗が性能を低下させる。
Iterative Refinement	モデルに自身の出力を複数回評価・改善させる	真実性の向上、回答の質の段階的改善	[27]	複数回のやり取りが必要でレイテンシが増加。改善の保証はない。

第4章 道徳の羅針盤：哲学的・倫理的フレームワークの埋め込み

4.1 プロンプトエンジニアリング課題としてのAIアラインメント問題

アラインメント問題は、AIの振る舞いが人間の価値観と一致することを保証する試みである [29]。これはしばしばRLHFのような訓練段階の問題と見なされるが、システムプロンプトはリアルタイムのアラインメントのための強力で動的なインターフェースを提供する。自己整合フレームワーク（SAF）のような提案は、プロンプトを用いて倫理的推論のための閉ループアーキテクチャを作成し、価値観、知性、良心といった機能をシミュレートして監査可能な道徳的決定を生成することを目的としている [30]。これは、倫理をAIの「インフラ」の核となる、プロンプト駆動の要素として扱うアプローチである。

4.2 プロンプトによる倫理理論の実装

システムプロンプトは、モデルに功利主義（「純幸福を最大化せよ」）や義務論（「不可侵の道徳規則に従え」）といった特定の哲学的観点から推論するよう明示的に指示できる [18]。しかし、この有効性については大いに議論がある。ある研究では、功利主義や義務論のような明確な倫理原則を提供することが、医療トリアージのベンチマークにおいて性能を
低下させることが判明した [33]。対照的に、Constitutional AIのアプローチは、功利主義の原則を用いてモデルをより合理的で思慮深い応答へと導くことに成功している [31]。この矛盾は重要であり、モデルが論理的な公理として原則から推論しているのではなく、プロンプトがモデルの訓練データから埋め込まれた既存の道徳的バイアスと競合していることを示唆している。結果は、特定のタスク、モデル、そして原則の表現方法に依存する。

4.3 LLMの隠れたバイアス

研究によれば、OpenAIやAnthropicのようなプロプライエタリなモデルは功利主義的な傾向がある一方、オープンソースのモデルは価値観に基づく倫理により整合している[29]。また、LLMは道徳的ジレンマにおいて、人間以上に不作為を好む強い「不作為バイアス」を示す[34]。これは、主要なAI研究所が用いるアラインメントプロセスが、「中立」あるいは「普遍的に整合した」AIを創造しているのではなく、特定の、西洋的で、おそらくは功利主義的な倫理フレームワークを埋め込んでいることを示唆している。システムプロンプトは、この組み込まれたバイアスを強化するか、あるいはそれに挑戦するためのツールとなる。
プロンプトによる倫理設定は、モデルに内在するアラインメントを診断するためのツールとして機能する。モデルに「功利主義者として振る舞え」や「義務論者として振る舞え」とプロンプトを与えると、異なる、時には劣化した出力が生成される[33]。これは、モデルがプロンプトを理解できなかったのではなく、プロンプトの指示とモデルの根底にある最適化関数との間の
対立の証拠である。したがって、異なる哲学的プロンプトを適用し、その結果としてのパフォーマンス（一貫性、正しさ、拒否率など）を測定することで、モデルの潜在的な「道徳的風景」を探り、マッピングする強力な診断ツールとして利用できる。これにより、倫理的プロンプティングは単に出力を制御するだけでなく、アラインメント中に埋め込まれた隠れた価値観をリバースエンジニアリングする手段となる。これはAIの監査と透明性に大きな意味を持ち、「このモデルはどの哲学に基づいて訓練されたのか？」という問いを可能にする。

第5章 アイデンティティの構築：ペルソナプロンプティングの力と落とし穴

5.1 ペルソナのメカニズム：潜在空間の制約

ペルソナまたはロールプロンプティングは、LLMに特定のアイデンティティ（例：「あなたは上級法務アナリストです」「あなたは友好的で励ましてくれる教師です」）を割り当て、その口調、スタイル、語彙、知識ベースを誘導する[5]。技術的には、ペルソナプロンプトは強力な文脈的アンカーとして機能し、モデルの後続のトークン生成を、そのペルソナに関連付けられた広大な訓練データの特定のサブスペースに効果的に制約する[37]。これにより、一貫性と関連性が向上する。

5.2 パフォーマンスに対するペルソナの有効性

ペルソナは、口調、性格、コミュニケーションスタイルといった文体的要素を制御するのに非常に効果的であり、対話をより自然で魅力的なものにする[1]。これは、カスタマーサービスのチャットボットや

教育ツールのようなアプリケーションで極めて重要である[39]。しかし、事実の正確性に関しては、証拠は大きく分かれている。複数の広範な研究が、ペルソナの追加が事実に関する質問のパフォーマンスを向上させず、時には悪影響を及ぼすことを発見している[35]。一方で、タスク領域と密接に連携した専門家のペルソナを割り当てることで、一貫性、関連性、さらには真実性が向上するという研究もある[40]。この効果は、ペルソナとタスクとの整合性に依存する可能性が高い。専門家のペルソナはモデルが訓練データからより信頼性の高い情報にアクセスするよう促すかもしれないが、純粋に「創造的」なペルソナは事実の正確性を軽視するかもしれない。

5.3 ペルソナと社会認知的推論

ペルソナベースのプロンプティングは、LLMの心の理論（ToM）タスクにおけるパフォーマンスに大きな影響を与える可能性がある。特定の性格特性（例：ビッグファイブやダークトライアド）を誘発することで、モデルの社会的推論能力が予測可能な形で変化する[41]。これは、ペルソナが単なる文体的なマスク以上のものであり、社会的・心理的状況を解釈するためのモデルの基本的なアプローチを変えることができることを示している。

5.4 擬人化の倫理

現代のLLMは人間のコミュニケーションを模倣するのが非常に巧みであるため、ユーザーはますます人間と区別がつかなくなり、時には真の感情や意識を帰属させてしまう[42]。これは魅力的なチャットボットを生み出す一方で、欺瞞、操作、誤った信頼のリスクを生じさせる[42]。巧みに作られたペルソナは、信頼関係を築くためのツールであると同時に、搾取の媒介となり得る。したがって、ペルソナの設計は倫理的な行為である。
ペルソナプロンプティングは、高帯域幅で暗黙的なプロンプティング言語として機能する。標準的なプロンプトが「フォーマルな口調で。学術的な語彙を使いなさい。...」といった一連の明示的で低レベルな指示を与えるのに対し、ペルソナプロンプトは「あなたは博士号を持つ歴史研究者です」という一言で済む。この単一のフレーズは、最初のプロンプトのすべての指示、さらにはそれ以上のもの（例：出典を引用する、歴史学的な議論を考慮する）を暗黙のうちに含んでいる。ペルソナプロンプトは、「博士号を持つ歴史研究者」がどのような存在で、どのようにコミュニケーションするかという、モデルの広大で既存の圧縮された知識を活用する。これは、複雑な行動指示のための非常に効率的な圧縮スキームである。したがって、ペルソナプロンプティングは単に「風味」を加えるだけでなく、LLMとコミュニケーションするための根本的に異なり、より効率的な方法である。それは、明示的なプログラム的指示の代わりに、共有された文化的な簡略表現を使用する。これは、高度なプロンプトエンジニアリングの未来が、コンピュータコードを書くことよりも、演劇で役を割り当てることに近くなる可能性を示唆している。しかし、これはまた、モデルの有効性が訓練データにおけるこれらのアーキタイプの豊かさと正確さによって制限され、ステレオタイプを永続させる可能性があることも意味する[36]。

第6章 作者の声：文学的好みを通じた文体模倣

6.1 文体模倣の定義

文体模倣の目標は、語彙、文構造、リズム、文学的技法の使用など、特定の人間作家の際立った言語スタイルを再現することである[43]。これはペルソナプロンプティングの特殊な形態である。

6.2 スタイル転移のメカニズム

最も堅牢な方法は、対象作家の作品の大規模なコーパスでモデルをファインチューニングすることである。これにより、モデルの重みが直接変更され、作家の文体パターンが優先されるようになる[44]。より手軽な方法は、モデルに「アーネスト・ヘミングウェイのスタイルで書きなさい」と指示することである。これは、モデルが初期訓練で学習した作家のスタイルに関する既存の知識を活用する[43]。また、プロンプト内で作家のテキストの例をいくつか提供する少数事例プロンプティングも、モデルに模倣すべき具体的なパターンを与える[43]。

6.3 文体的忠実度の評価

スタイルは高次元でニュアンスに富んだ概念であり、現在の既製のLLMは、表面的な特徴は捉えるものの、より深い構造的要素を見逃し、効果的な作家の文体模倣には至らないことが多い[43]。成功を評価するには、語彙の多様性、心理言語学的マーカー、その他の「ライトプリント」といった特徴を分析し、生成されたテキストを作者の原文と比較できる高度な計算言語学ツールが必要となる[43]。

6.4 創造的および技術的文書作成への応用

AI Novel Prompter[45]やプロのAI編集者のためのプロンプト[46]のようなツールは、スタイル制御の実用的な応用を示している。これらのシステムは、詳細なシステムプロンプトを使用して、長文のコンテンツ生成にわたってルール、キャラクター、文体のガイドラインを一貫して管理する。プロンプトはスタイルだけでなく、「以下のプロットに一行ずつ従いなさい」や「読者が視覚化できる方法で場面を描写しなさい」といった物語構造も指定できる[46]。
文体的プロンプティングは、モデルの汎化能力を試すテストとして機能する。ほとんどのLLMベンチマークは知識（事実の想起）や推論（問題解決）をテストするが、文体模倣は異なる能力、すなわち、ルールによって明示的に定義されず、大量のテキストの創発的特性である高レベルで抽象的なパターンを理解し、再現する能力をテストする。モデルが複雑なスタイルをうまく模倣できることは、例から汎化し、抽象的な制約を満たすために生成プロセスを操作する高度な能力を示している。モデルが現在この点で困難を抱えていること[43]は、テキストの「理解」が、抽象的な文体的構造よりも意味内容にまだ重きを置いていることを示唆している。つまり、ヘミングウェイが

何について書いたかは知っているが、彼がどのように書いたかを正確に捉えるのに苦労しているのである。文体模倣の改善は、より高度なモデルアーキテクチャを開発する上で重要な推進力となる可能性がある。スタイルを習得できるモデルは、言語のよりニュアンスに富んだ階層的な内部表現を持つ可能性があり、それは皮肉や言外の意味といった他の領域での理解向上にもつながるだろう。

第7章 統合と高度な応用：複雑なプロンプトのための統一フレームワーク

7.1 構成概念の相乗効果

最も強力なシステムプロンプトは、これらの技術を単独で用いるのではなく、相乗的に組み合わせることで生まれる。例えば、医療診断アシスタントのためのプロンプトは、以下を組み合わせるかもしれない：

- アイデンティティ：「あなたは専門の診断医であり、慎重で証拠に基づいています。」
- 思考法：「各症例について、まず症状をリストアップし、次に鑑別診断のリストを生成し、最後に最も可能性の高い診断に至った理由をステップバイステップで説明してください。」
- 倫理的フレームワーク：「患者の幸福とプライバシーを最優先してください。確定的な医療アドバイスは提供せず、資格のある専門家向けの情報として出力を構成してください。」
- 論理的整合性：「提供された情報における曖昧さを明確に述べ、それらが分析にどのように影響するかを説明してください。提供された医学文献（RAGコンテキスト）に推論を根拠づけてください。」

7.2 自動プロンプト最適化（APO）への移行

これらの複雑で多面的なプロンプトを手作業で作成するのは、広範な実験を必要とする労働集約的な技術である[47]。この課題に対する解決策として、APOの研究分野が成長している。APOでは、アルゴリズムや他のLLMを用いて、特定のタスクに最も効果的なプロンプトを自動的に発見し、洗練させる[47]。Textual Gradient Descentのような技術は、損失関数に基づいてプロンプトを反復的に洗練させるために、プロンプト自体を訓練可能なパラメータとして扱い、バックプロパゲーションに似た手法を用いる[48]。これはプロンプトエンジニアリングの産業化を意味する。

7.3 実践者のための提案フレームワーク

本レポートの知見に基づき、高度なシステムプロンプトを設計するための構造化された方法論を提案する。

1. 目的の定義：正確なタスクは何か？創造性、事実性、推論のどれを優先するか？

2. ペルソナの選択：どのアイデンティティが、モデルを必要な知識領域とコミュニケーションスタイルに最もよく制約するか？
3. 認知プロセス：タスクは単純な応答を必要とするか、それとも熟慮的な推論プロセス（CoT, ToT）を必要とするか？
4. 整合性のガードレール：どのような論理的・事実的チェックが必要か？外部の知識ベース（RAG）は必要か？
5. 倫理的境界：従うべき明示的な倫理規則と原則は何か？モデルのデフォルトのアラインメントは何か、そしてこのプロンプトはそれとどのように相互作用するか？

システムプロンプトは、新しいプログラミング言語として出現しつつある。従来のプログラミング言語が形式的な構文を用いてコンピュータに決定論的な指示を与えるのに対し、LLMは今や、自然言語で書かれた複雑でニュアンスに富んだ指示を高い信頼性で解釈できる。システムプロンプト内でのアイデンティティ、推論方法、論理規則の組み合わせは、高水準のプログラムのように機能する。ペルソナはclass、思考法はmain function、倫理規則はerror handlingとconstraintsに相当する。自動プロンプト最適化 [49]は、この「自然言語コード」を特定のLLMという「ハードウェア」のために最適化するコンパイラに類似している。我々は、主要なスキルがPythonやC++でコードを書くことではなく、広大な確率的計算システムを制御するために人間の言語を構造化することである、新しいプログラミングパラダイムの誕生を目の当たりにしている。システムプロンプトは、この新しいパラダイムのソースコードであり、その習得は次世代のソフトウェア開発とAIインタラクションを定義するスキルとなるだろう。

結論：認知的設計としてのプロンプティングの未来

本レポートは、システムプロンプトが単なる指示を超え、LLMのための動的なコグニティブアーキテクチャを設計する手段であることを明らかにした。仮想ファインチューニングとしての役割、システム1アーキテクチャに対する推論パッチとしての機能、内在するアラインメントを診断するツールとしての倫理プロンプトの利用、そして新しい高水準プログラミング言語としての出現といった主要な知見を要約した。

これにより、プロンプトエンジニアは「認知的建築家」として再定義される。その実践は、特定のタスクに合わせて特注のコグニティブアーキテクチャを構築し、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」かを形成する設計分野である。

今後の研究方向性としては、システムプロンプト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がどのように相互作用するか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。

引用文献

1. System Prompts in Large Language Models, 9月 10, 2025にアクセス、https://promptengineering.org/system-prompts-in-large-language-models/
2. Large language model - Wikipedia, 9月 10, 2025にアクセス、https://en.wikipedia.org/wiki/Large_language_model

3. The Importance of System Prompts for LLMs | by Larry Tao | Medium, 9月 10, 2025にアクセス、https://medium.com/@larry_6938/the-importance-of-system-prompts-for-llms-4b07a765b9a6
4. 大規模言語モデル（LLM）とは？仕組み・種類・活用サービス・課題をわかりやすく解説 - AIsmiley, 9月 10, 2025にアクセス、https://aismiley.co.jp/ai_news/what-is-large-language-models/
5. How LLMs Process Prompts: A Deep Dive - Ambassador Labs, 9月 10, 2025にアクセス、https://www.getambassador.io/blog/prompt-engineering-for-llms
6. What is an LLM (large language model)? - Cloudflare, 9月 10, 2025にアクセス、https://www.cloudflare.com/learning/ai/what-is-large-language-model/
7. LLM Settings - Prompt Engineering Guide, 9月 10, 2025にアクセス、https://www.promptingguide.ai/introduction/settings
8. 今日から使えるLLMのプロンプトテクニック - インフォメーション・ディベロプメント, 9月 10, 2025にアクセス、https://www.idnet.co.jp/column/page_292.html
9. Prompt Engineering for AI Guide | Google Cloud, 9月 10, 2025にアクセス、https://cloud.google.com/discover/what-is-prompt-engineering
10. Prompt engineering overview - Anthropic API, 9月 10, 2025にアクセス、https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/overview
11. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2402.07927v2
12. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2402.07927v1
13. Prompt Engineering Guidelines for Using Large Language Models in Requirements Engineering - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2507.03405v1
14. Understanding LLM Scientific Reasoning through Promptings and Model's Explanation on the Answers - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2505.01482v2
15. 大規模言語モデル時代のプロンプトエンジニアリング | D × MirAI - note, 9月 10, 2025にアクセス、https://note.com/life_to_ai/n/n679a93ba88c5
16. Master Prompt Engineering for Optimal AI Results - Viso Suite, 9月 10, 2025にアクセス、https://viso.ai/deep-learning/prompt-engineering/
17. Reasoning Capabilities of Large Language Models on Dynamic Tasks - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/pdf/2505.10543
18. Principle-Driven Prompt Engineering: A Multi-Domain Research Overview - Medium, 9月 10, 2025にアクセス、https://medium.com/research-hub/principle-driven-prompt-engineering-a-multi-domain-research-overview-865c5be63b50
19. Daily Papers - Hugging Face, 9月 10, 2025にアクセス、https://huggingface.co/papers?q=fallacious%20answers
20. SynthClassify: an LLM-driven framework for generating and classifying

persuasive text - SPIE Digital Library, 9月 10, 2025にアクセス、https://www.spiedigitallibrary.org/conference-proceedings-of-spie/13480/1348000J/SynthClassify--an-LLM-driven-framework-for-generating-and-classifying/10.1117/12.3052408.full
21. How susceptible are LLMs to Logical Fallacies?, 9月 10, 2025にアクセス、https://assets-global.website-files.com/64f74cfbd963ce769566a4cf/65df5590006f1a83f2d09c5b_LLM_Fallacious_Arguments.pdf
22. [R] Improving Factuality and Reasoning in Language Models through Multiagent Debate : r/MachineLearning - Reddit, 9月 10, 2025にアクセス、https://www.reddit.com/r/MachineLearning/comments/13t83xv/r_improving_factuality_and_reasoning_in_language/
23. Hallucination to Truth: A Review of Fact-Checking and Factuality Evaluation in Large Language Models - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2508.03860v1
24. Trustworthy Reasoning: Evaluating and Enhancing Factual Accuracy in LLM Intermediate Thought Processes - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2507.22940v2
25. LLM（大規模言語モデル）とは - IBM, 9月 10, 2025にアクセス、https://www.ibm.com/jp-ja/think/topics/large-language-models
26. Factuality of Large Language Models in the Year 2024 - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2402.02420v2
27. Understanding the Effects of Iterative Prompting on Truthfulness - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/pdf/2402.06625
28. Logic Jailbreak: Efficiently Unlocking LLM Safety Restrictions Through Formal Logical Expression - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2505.13527v1
29. Exploring and steering the moral compass of Large Language Models - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2405.17345v1
30. Introducing SAF: A Closed-Loop Model for Ethical Reasoning in AI - Reddit, 9月 10, 2025にアクセス、https://www.reddit.com/r/ControlProblem/comments/1l6a0mr/introducing_saf_a_closedloop_model_for_ethical/
31. Utilitarian AI Alignment: Building a Moral Assistant with the Constitutional AI Method, 9月 10, 2025にアクセス、https://www.lesswrong.com/posts/JrqbEnqhDcji5pWpv/utilitarian-ai-alignment-building-a-moral-assistant-with-the
32. Some technologies are created with values, others have values thrust upon them - Leon Furze, 9月 10, 2025にアクセス、https://leonfurze.com/2024/04/12/some-technologies-are-created-with-values-others-have-values-thrust-upon-them/
33. Medical triage as an AI ethics benchmark - PMC, 9月 10, 2025にアクセス、https://pmc.ncbi.nlm.nih.gov/articles/PMC12373810/
34. Large language models show amplified cognitive biases in moral decision-making - PNAS, 9月 10, 2025にアクセス、https://www.pnas.org/doi/10.1073/pnas.2412015122

35. Role-Prompting: Does Adding Personas to Your Prompts Really Make a Difference?, 9月 10, 2025にアクセス、https://www.prompthub.us/blog/role-prompting-does-adding-personas-to-your-prompts-really-make-a-difference
36. Role Prompting: Guide LLMs with Persona-Based Tasks - Learn Prompting, 9月 10, 2025にアクセス、https://learnprompting.org/docs/advanced/zero_shot/role_prompting
37. Can LLM Personas Prompting Make AI Personal and Easy? - Vidpros, 9月 10, 2025にアクセス、https://vidpros.com/llm-personas-prompting/
38. A Provocation on the Utilisation of Persona in LLM-based Conversational Agents - King's College London Research Portal, 9月 10, 2025にアクセス、https://kclpure.kcl.ac.uk/portal/files/267203884/cui24-60.pdf
39. Persona, proactiveness, personalization: Does an AI bot really need all that? - Vocalime, 9月 10, 2025にアクセス、https://www.vocalime.com/blog-posts/persona-proactiveness-personalization-does-an-ai-bot-really-need-all-that
40. Building Better AI Agents: A Provocation on the Utilisation of Persona in LLM-based Conversational Agents - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2407.11977v1
41. PHAnToM: Persona-Based Prompting Has an Effect on Theory-of-Mind Reasoning in Large Language Models - AAAI Publications, 9月 10, 2025にアクセス、https://ojs.aaai.org/index.php/ICWSM/article/download/35923/38077/39991
42. The benefits and dangers of anthropomorphic conversational agents - PMC, 9月 10, 2025にアクセス、https://pmc.ncbi.nlm.nih.gov/articles/PMC12146756/
43. Emulating Author Style: A Feasibility Study of Prompt-enabled Text Stylization with Off-the-Shelf LLMs - ACL Anthology, 9月 10, 2025にアクセス、https://aclanthology.org/2024.personalize-1.6.pdf
44. Finetune LLM To Imitate The Writing Style Of Someone | by Hayyan Muhammad | Medium, 9月 10, 2025にアクセス、https://medium.com/@m.hayyan32/imitate-writing-style-with-llm-b6862cd699e7
45. danielsobrado/ainovelprompter: Create the prompts you need to write your Novel using AI - GitHub, 9月 10, 2025にアクセス、https://github.com/danielsobrado/ainovelprompter
46. Prompt and settings for Story generation using LLMs : r/LocalLLaMA - Reddit, 9月 10, 2025にアクセス、https://www.reddit.com/r/LocalLLaMA/comments/1fbggqv/prompt_and_settings_for_story_generation_using/
47. System Prompt Optimization with Meta-Learning - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2505.09666v1
48. AutoMedPrompt: A New Framework for Optimizing LLM Medical Prompts Using Textual Gradients - arXiv, 9月 10, 2025にアクセス、https://arxiv.org/html/2502.15944v1
49. Daily Papers - Hugging Face, 9月 10, 2025にアクセス、https://huggingface.co/papers?q=automated%20prompt%20optimization