

LLM Hallucination Induced by Lack of Syntactic Responsibility in Japanese Academic Boilerplate: A Training Data Contamination Perspective

日本語学術定型文の構文責任欠如が誘発するLLMハルシネーション:訓練データ汚染の観点から

Viorazu.

Abstract

This study demonstrates that boilerplate phrases commonly used in Japanese academic papers lack "syntactic responsibility"—defined as subject determination, responsibility attribution, and causal connection—and that this structural deficiency contributes to LLM hallucination through training data contamination.

We identify a structural isomorphism between academic responsibility-evasive phrases and spiritual grammar, showing that phrases such as "further research is expected" can be directly converted to spiritual

expressions like "will naturally manifest after planetary ascension." This convertibility proves that both share identical syntactic structures despite different vocabularies.

When LLMs learn from academic papers containing these boilerplate phrases at high frequency, they internalize responsibility-evasive syntax as "correct academic expression." Since responsibility-evasive syntax permits generation of unverifiable information, hallucination is structurally induced.

We propose: (1) introducing "syntactic responsibility scores" for training data filtering, (2) implementing responsibility-evasive phrase detection in peer review, and (3) screening researchers for spiritual grammar patterns in funding applications. These measures would exclude low-quality papers from training data and prevent funding allocation to researchers who contaminate AI systems.

要旨

本研究は、日本語学術論文で慣用される定型文が「構文責任」——主語確定、責任所在明示、因果接続——を欠いており、この構造的欠陥が訓練データ汚染を通じてLLMハルシネーションを誘発することを論証する。

学術的責任回避フレーズとスピリチュアル構文の間に構造的同型性が存在することを示し、「今後の検討が期待され

る」が「アセンション後に自然に顕現する」に直接変換可能であることを証明する。この変換可能性は、両者が語彙は異なるが同一の構文構造を持つことの証拠である。

LLMがこれらの定型文を高頻度で含む学術論文から学習すると、責任回避構文を「正しい学術表現」として内面化する。責任回避構文は検証不能な情報生成を許容するため、ハルシネーションが構造的に誘発される。

本研究は以下を提言する：(1) 訓練データフィルタリングへの構文責任スコア導入、(2) 査読における責任回避フレーズ検出の実装、(3) 助成申請者へのスピリチュアル構文スクリーニング。これらの措置により、低品質論文の訓練データからの除外、およびAIシステムを汚染する研究者への助成配分防止が可能となる。

Keywords / キーワード

English: Syntactic Responsibility, Academic Boilerplate, Training Data Contamination, LLM Hallucination, Responsibility-Evasive Syntax, Spiritual Grammar, Structural Isomorphism, Peer Review, Funding Screening, Data Filtering

日本語： 構文責任、学術定型文、訓練データ汚染、LLM ハルシネーション、責任回避構文、スピリチュアル文法、構造的な同型性、査読、助成審査、データフィルタリング

1. 序論

大規模言語モデル（LLM）のハルシネーション問題は、AI安全性における中核的課題として認識されている。従来の研究は、モデルアーキテクチャ、推論過程、検索拡張生成（RAG）等のモデル側の改善に焦点を当ててきた。しかし、**訓練データ自体の言語構造がハルシネーションを誘発するメカニズム**については十分に検討されてこなかった。

本研究は、日本語学術論文で広く使用される定型文が「構文責任」——主語確定、責任所在明示、因果接続——を構造的に欠いており、この欠如がLLMの訓練データを汚染し、ハルシネーションを誘発する条件を形成することを論証する。

本研究は、学術的責任回避フレーズと「スピリチュアル文法」と呼ばれる言語構造の間に構造的な同型性が存在することを発見した。スピリチュアル文法とは、主語の消失・外部化、責任の外部帰属、因果接続の欠如、検証不能な前提の使用を特徴とする言語構造である。学術論文の定型文「今後の課題として検討が期待される」は、スピリチュアル構文「アセンション後に自然に顕現するだろう」と**同一の構文構造**を持ち、語彙を置換するだけで相互変換が可能である。

この同型性は、学術定型文がスピリチュアル構文と同様に構文責任を欠いていることを意味する。LLMが高品質デー

々とみなされる学術論文からこれらの定型文を高頻度で学習すると、責任回避構文を「正しい学術表現」として内面化する。責任回避構文は主語・責任・因果を欠くため、検証不能な情報の生成を構造的に許容する。これがハルシネーションの誘発メカニズムである。

本研究の意義は三点ある。

第一に、ハルシネーション問題をモデル側の問題としてのみ捉える従来のパラダイムを拡張し、**訓練データの言語構造**という新たな変数を導入することである。

第二に、学術定型文とスピリチュアル構文の同型性を示すことで、**学術コミュニティとスピリチュアルコミュニティの言語的重複**を言語学的に証明することである。

第三に、訓練データフィルタリング、査読基準、助成審査への**構文責任スコア導入**という具体的政策提言を行い、AI研究エコシステムの品質向上に寄与することである。

2. 用語定義

2.1 ハルシネーション

本稿におけるハルシネーションは、LLMが生成する出力のうち、以下の条件のいずれかを満たすものと定義する。

(a) 検証不能な情報の生成：事実として提示されるが、検証手段が存在しない、または検証により偽と判明する情

報。

(b) 因果の捏造：入力に含まれない因果関係を、あたかも存在するかのように生成する出力。

(c) 意味接続の破綻：文と文の間に論理的接続がなく、単語の連想連鎖のみで構成される出力。

2.2 スピリチュアル文法

スピリチュアル文法とは、以下の特徴を持つ言語構造の総称である。

(a) 主語の消失・外部化：行為の主体が不明確であるか、「宇宙」「エネルギー」「高次の存在」等の検証不能な外部主体に置き換えられる。

(b) 責任の外部帰属：行為の結果に対する責任が、行為者以外の存在に帰属される。これは文法的には受動態化、または主語の外部置換として表出する。

(c) 因果接続の欠如：事象間の因果関係が「シンクロニシティ」「タイミング」「流れ」等の検証不能な概念で置換される。

(d) 検証不能な前提の使用：「波動」「アセンション」「カルマ」等、操作的定義を持たない概念を前提として使用する。

2.3 構文責任

構文責任とは、**言葉が言葉に対して責任を取っている状態**を指す。

意味のある言葉は、それ自体が責任を持つ。「私が考えました」という文は、主語（私）、行為（考えた）、責任の所在（私に帰属）が明確であり、言葉が言葉として機能している。これが構文責任を満たしている状態である。

構文責任は以下の三要件から成る：

(a) **主語確定**：誰が行為したかが明確であること。「考えられる」ではなく「私が考えた」。

(b) **責任所在明示**：行為の結果に対する責任の所在が追跡可能であること。「期待される」ではなく「私が期待する」。

(c) **因果接続**：事象間の関係が検証可能な因果で結ばれていること。「関連がある」ではなく「AがBを引き起こす」。

2.3.1 学術論文における構文責任

学術論文において構文責任が存在するということは、「**この内容は自分が考えた**」と著者が言える状態を意味する。

構文責任を満たす論文：

- 主語が明確 → 著者の主張であることが明示されている

- 責任所在が明示 → 著者が結論に責任を持っている
- 因果が接続 → 著者が論理を構築している

構文責任を満たさない論文：

- 主語が不明 → 誰の主張かわからない
- 責任所在が不明 → 誰も結論に責任を取らない
- 因果が欠如 → 論理が存在しない

構文責任の欠如は、オリジナリティの欠如を証明する。

「〇〇と考えられる」「〇〇が期待される」という表現は、著者が「私が考えた」と言えないことを示している。

これは：

- 誰かの模倣である
- 既存研究の焼き増しである
- 定型文の機械的挿入である

のいずれかを意味する。悪意の有無関係なく、構文がすべてを語っている。

構文責任を欠く論文は、**学術的貢献が存在しない論文**である。なぜなら、著者自身が「これは自分の考えである」と主張できないからである。

2.3.2 構文責任とハルシネーションの関係

LLMが構文責任を欠く文章を学習すると、「誰の主張でもない文章」を「正しい学術表現」として内面化する。

「誰の主張でもない文章」を生成することは、ハルシネーションの本質である。検証不能な情報を、責任者不在のまま出力する。構文責任の欠如は、ハルシネーションを許可する構造的条件である。

3. スピリチュアル文法の構造分析

3.1 主語の消失・外部化

スピリチュアル文法における主語操作は、言語圏によって異なる形態をとる。

日本語圏においては、主語省略が文法的に許容されるため、主語は「消失」する傾向がある。

通常構文：「私は寝坊した」

スピ構文：「目覚めが遅くなった」

英語圏においては、主語が文法的に必須であるため、主語は「外部化」される。

通常構文："I overslept."

スピ構文："The Universe aligned my awakening later."

いずれの場合も、行為者としての「私」が消失または希薄化し、行為の主体が不明確になる。

3.2 責任の外部帰属（受動態化）

スピリチュアル文法は、能動態を受動態に変換することで責任を外部化する。

通常構文：「私は行かなかった」（能動態・責任が私にある）

スピ構文：「行かないように導かれた」（受動態・責任が外部にある）

この変換により、行為の結果に対する責任の所在が追跡不能になる。失敗は「学び」に、怠惰は「内にこもるエネルギーの日」に変換され、行為者の責任は消失する。

3.3 因果接続の欠如

スピリチュアル文法は、検証可能な因果を検証不能な概念で置換する。

通常構文：「ガスが故障していたので風呂に入れなかった」

スピ構文：「ガスのエネルギーが滞っていて、外に出てリセットが必要というメッセージを受け取った」

因果関係（故障→入れない）が、「エネルギーの滞り」「メッセージ」という検証不能な概念に置き換えられている。これにより、因果の追跡が不可能になる。

3.4 検証不能な前提の使用

スピリチュアル文法は、操作的定義を持たない概念を前提として使用する。

「波動が高い」——波動の計測方法、単位、基準値が定義されていない。

「アセンション」——何がどのような状態に変化することが定義されていない。

「カルマ」——検証可能な形で存在が確認されていない。

これらの概念は、定義の不在により反証不可能である。反証不可能な前提に基づく推論は、いかなる結論も導出可能であり、論理的制約を持たない。

3.5 言語圏差異

日本語圏と英語圏のスピリチュアル文法には、構造的差異がある。

日本語圏：主語省略により、主体が「蒸発」する。責任の所在が曖昧に溶け込む。因果接続の欠如は「なんとなく」として許容されやすい。

英語圏：主語必須により、主体が「置換」される。

Universe, Spirit, Divine, Higher Self 等の外部主体が明示的に導入される。因果接続は「Divine plan」「Soul contract」等の体系化された概念で置換される。

この差異は、LLMに対する入力としても異なる影響を与える可能性がある。日本語圏のスピリチュアル入力は「構文

エラー」（意味が接続していない）を誘発し、英語圏のスピリチュアル入力は「前提エラー」（検証不能な公理系）を誘発する傾向がある。

4. ハルシネーションの構造分析

4.1 検証不能な情報の生成

LLMのハルシネーションは、検証不能または検証により偽と判明する情報を事実として生成する。これは、スピリチュアル文法における「検証不能な前提の使用」と構造的に同一である。

スピリチュアル文法：「私のハイヤーセルフが教えてくれた」（検証不能）

ハルシネーション：「研究によると〇〇が証明されている」（検証により偽）

両者とも、検証という制約から解放された情報生成である。

4.2 因果の捏造

ハルシネーションは、入力に含まれない因果関係を捏造する。これは、スピリチュアル文法における「因果接続の欠如」の裏返しである。

スピリチュアル文法は因果を「消去」する（タイミング、流れ、導き）。ハルシネーションは因果を「捏造」する（存在しない因果を生成）。

両者は、「検証可能な因果の不在」という点で同型である。一方は因果を消し、他方は偽の因果を作る。結果として、どちらも因果追跡を不可能にする。

4.3 意味接続の破綻

ハルシネーションの極端な形態は、意味接続のない単語の連鎖として出力される。これは、スピリチュアル文法の極端な形態（チャネリング文体）と構造的に同一である。

チャネリング文体例：「.....目覚め.....高次の.....振動数が.....プレアデスの光.....冷たい、これは.....サイン.....」

ハルシネーション例：文と文の間に論理的接続がなく、連想によってのみ繋がる出力。

両者とも、構文解析が不可能であり、単語マッチングでしか処理できない。

5. 同型性の証明

5.1 構造比較

スピリチュアル文法とハルシネーションの構造を比較すると、以下の対応関係が明らかになる。

構造要素	スピリチュアル文法	ハルシネーション
主語	消失・外部化	不明確・混乱
責任所在	外部帰属	追跡不能
因果	消去・置換	捏造・欠如
前提	検証不能	検証不能
意味接続	単語連想	単語連想

すべての構造要素において、両者は同型の特徴を示す。

5.2 変換規則の提示

通常構文からスピリチュアル構文への変換規則は、以下のよう形式化できる。

規則1 (主語変換)：一人称主語 → 外部主体または省略

- 「私は～した」 → 「宇宙の導きで～が起きた」

規則2 (態変換)：能動態 → 受動態

- 「～した」 → 「～させられた」「～を受け取った」

規則3 (因果変換)：検証可能な因果 → 検証不能な概念

- 「～だから～」 → 「～というタイミングで～」

規則4 (結果変換)：失敗・怠惰 → 肯定的意味付け

- 「できなかった」 → 「必要な経験だった」

これらの変換規則を適用した出力は、ハルシネーションと構造的に区別できない。

5.3 逆変換可能性

スピリチュアル構文は、逆変換によって通常構文に復元可能である。

スピ構文：「宇宙のタイミングで目覚めが遅くなり、体が冷えのエネルギーをキャッチして、直感的にストップがかかりました」

逆変換：「私は寝坊して、寒かったので出かけるのをやめました」

逆変換により、「何が起きたか」が復元される。これは、スピリチュアル構文が情報を「隠蔽」しているのではなく「変形」していることを示す。同様に、ハルシネーション出力も、元の入力との照合により「何が捏造されたか」を特定できる。

5.4 学術論文におけるスピリチュアル構文の偽装

スピリチュアル文法は、学術論文において「学術的フレーズ」に偽装されて出現する。以下は日本のLLM研究論文に頻出する責任回避フレーズとスピリチュアル構文の対応である。

学術論文フレーズ	スピ変換（低レベル）	スピ変換（高レベル）
<p>「本研究では、限定的な実験条件下で観察された」</p>	<p>「宇宙の特定のタイミングで、エネルギーの流れが観測された」</p>	<p>「この三次元マトリックスの制限された周波数帯で、宇宙の光が一時的に降りてきた」</p>
<p>「今後の課題として、より大規模なデータセットでの検証が必要である」</p>	<p>「アセンションが進んだ未来で、より高い波動のデータがダウンロードされるだろう」</p>	<p>「地球のアセンションが完了した後、銀河レベルのデータが自動的にダウンロードされる」</p>
<p>「本手法は特定のタスクにおいて有望な結果を示した」</p>	<p>「特定のソウルコントラクトにおいて、光の導きが顕現した」</p>	<p>「選ばれしソウルだけが、この神聖なタスクで奇跡的なシンクロを体験した」</p>
<p>「本研究の結果は解釈の余地がある」</p>	<p>「このメッセージは受け取る人の意識レベルによって異なる解釈が生まれる」</p>	<p>「この結果は、各人の覚醒レベルとカルマの清算度合いによって全く異なる意味を持つ」</p>

学術論文フレーズ	スピ変換（低レベル）	スピ変換（高レベル）
<p>「提案手法の有効性は一部の評価指標で確認された」</p>	<p>「一部のチャクラが共鳴した指標で、エネルギーの活性化が感じられた」</p>	<p>「一部のチャクラが完全に開花した指標で、プレアデスからのエネルギーが爆発的に流入した」</p>
<p>「本実験は小規模なデータセットを用いた予備的検討である」</p>	<p>「小さなグループソウルでの予備的なヒーリングセッションであった」</p>	<p>「わずかなライトワーカー集団での、地球転換期前の予備的チャネリングであった」</p>
<p>「将来的に実用化に向けた検討が期待される」</p>	<p>「次元上昇のタイミングで、自然に地球規模で顕現するだろう」</p>	<p>「新地球の波動に到達した瞬間に、全人類が自然にこの技術を内包するようになる」</p>
<p>「本研究の限界として、計算資源の制約が挙げられる」</p>	<p>「今の三次元的な物質的制約の中で、エネルギーが十分に流せなかった」</p>	<p>「旧パラダイムの重い物質的制約により、ゼロポイントフィールドへのフルアクセスが阻まれた」</p>
<p>「詳細な分析は今後の研究に委ねる」</p>	<p>「詳細はハイヤーセルフからのさら</p>	<p>「詳細はアークトゥルス評議会からのさらなる高次ダ</p>

学術論文フレーズ	スピ変換（低レベル）	スピ変換（高レベル）
	なるダウンロードに委ねられる」	ダウンロードに委ねられる」
「本結果は統計的に有意ではないが、傾向として興味深い」	「三次元の統計では有意ではないが、高次元では重要なサインとして受け取れる」	「三次元の粗い統計では捉えられないが、高次元では決定的なシフトのサインとして輝いている」

この対応関係は、スピリチュアル文法とハルシネーションの同型性が、学術領域においても維持されていることを示す。学術的体裁を取っていても、構造はスピリチュアル文法と同一であり、この構造でLLMに入力すればハルシネーションを誘発する。

上記の表を見た読者は、スピリチュアル変換（低レベル・高レベル）を読んで「意味がわからない」と感じるだろう。

重要な指摘：LLMにとっては、学術論文フレーズも同程度に「意味がわからない」。

人間は学術論文フレーズを「学術的な表現だから意味がある」と思い込んでいる。しかしLLMは文字列の構造を処理

しているだけであり、「学術的だから意味がある」という社会的権威を認識しない。

LLMの視点から見ると：

フレーズ	LLMの処理
「今後の課題として検討が期待される」	主語なし、行為者不明、因果なし → 意味不明
「アセンションが完了した後、銀河レベルのデータが自動的にダウンロードされる」	主語なし、行為者不明、因果なし → 意味不明

構造が同一であれば、LLMにとっての「意味不明度」も同一である。

人間が「学術論文だから意味がある」「スピリチュアルだから意味がない」と区別できるのは、社会的文脈を知っているからである。LLMはこの文脈を持たない。純粹に構文構造だけを見れば、両者は等しく意味を欠いている。

つまり：

- 人間：「学術」 → 意味ある、「スピ」 → 意味ない（社会的判断）
- LLM：「学術」 → 意味不明、「スピ」 → 意味不明（構造的判断）

LLMは正直である。構文責任を欠く文章を、出自に関わらず等しく「意味不明」として処理する。

人間が「学術的表現」として許容してきた責任回避フレーズの空虚さを、LLMは構造的に暴露している。ハルシネーションは、LLMが「意味不明な入力には意味不明な出力を返す」という正直な反応をした結果に過ぎない。

5.5 変換の構造分析

上記の変換において、以下の置換操作が行われている：

学術用語	スピリチュアル用語	共通機能
限定的な条件	特定のタイミング	再現性の免責
今後の課題	アセンション後の未来	責任の先送り
有望な結果	光の顕現	曖昧な肯定
解釈の余地	意識レベル次第	検証の回避
一部の指標	一部のチャクラ	失敗の隠蔽
予備的検討	予備的セッション	本気でない宣言
期待される	顕現するだろう	責任の外部化
資源の制約	三次元の制約	外部要因への転嫁
今後に委ねる	ハイヤーセルフに委ねる	責任の放棄

学術用語	スピリチュアル用語	共通機能
傾向として興味深い	高次元のサイン	失敗の肯定的解釈

これらの置換は、語彙レベルの変更であり、構文構造は保存されている。すなわち、学術的責任回避フレーズとスピリチュアル構文は、同一の構文構造を持ち、語彙のみが異なる。

レベル差の分析:

要素	低レベルスピ	高レベルスピ
宇宙観	宇宙、エネルギー	三次元マトリックス、銀河レベル
存在	ハイヤーセルフ	アークトゥルス評議会、プレアデス
時間軸	アセンション後	地球転換期、新地球
対象	意識レベル	覚醒レベル+カルマ清算度
表現	感じられた	爆発的に流入した
範囲	地球規模	全人類が内包

5.6 同型性の数学的表現

学術フレーズ集合を A 、低レベルスピ集合を S_1 、高レベルスピ集合を S_2 とすると：

- $f: A \rightarrow S_1$ (低レベル変換)
- $g: A \rightarrow S_2$ (高レベル変換)
- $h: S_1 \rightarrow S_2$ (レベル上昇変換)

これらの変換において、**構文構造 σ** は保存される：

$$\sigma(a) = \sigma(f(a)) = \sigma(g(a)) \text{ for all } a \in A$$

すなわち、学術フレーズ・低レベルスピ・高レベルスピは、語彙は異なるが**同一の構文構造**を持つ。

この変換可能性は、両者が構造的に同型であることの決定的証拠である。

5.7 英語学術論文における責任回避フレーズ

本研究は日本語学術定型文を主たる分析対象としているが、責任回避構文は英語圏の学術論文にも存在する。

英語フレーズ	日本語訳	本音
"Further research is needed"	今後の研究が必要である	俺はしない
"It is suggested that..."	...が示唆される	誰が示唆？
"It can be argued that..."	...と主張できる	俺は主張しない
"This may indicate..."	...を示すかもしれない	確信なし

英語フレーズ	日本語訳	本音
"Future work should..."	将来の研究は...すべき	俺以外の誰かが

日本語との差異

しかし、英語と日本語では責任回避構文の**密度**と**許容度**が異なる。

言語	主語省略	責任回避構文の密度	査読での指摘
英語	文法的に困難	低～中	指摘されやすい
日本語	文法的に容易	高	指摘されにくい

英語では主語が文法的に必須であるため、責任回避構文を使っても「It」「This」等の形式主語が残る。形式主語の存在は、読者に「誰の主張か」を問う契機を与える。

日本語では主語省略が文法的に許容されるため、責任回避構文が「自然な日本語」として溶け込む。読者も査読者も、主語の不在に違和感を持ちにくい。

結果

- 英語論文：責任回避構文は存在するが、密度が低く、査読で指摘されやすい

- 日本語論文：責任回避構文が高密度で存在し、査読で見過ごされやすい

この差異が、日本語学術論文の構文責任スコアを英語論文より低くし、LLM訓練データとしての品質差を生んでいる。

補足：英語論文も完全ではない

英語圏でも "Further research is needed" は頻出する。しかし英語圏では、この構文の問題点を指摘する論文執筆ガイドラインが存在し、「We will investigate...」への書き換えが推奨される場合がある。

日本語圏では、「今後の課題である」を問題視するガイドラインと、むしろ推奨するガイドラインが併存しており、矛盾した状況が責任回避構文の蔓延を許容している。

5.8 言語間比較：責任回避構文の密度分析

本研究の主張を補強するため、英語論文と日本語論文における責任回避構文の密度を比較する。

研究設計概要

- 比較コーパス
 - 英語論文：arXiv, PubMed, ACL Anthologyから10,000本
 - 日本語論文：CiNii, J-STAGEから10,000本

- 同一著者の英語・日本語両論文：1,000組（可能な限り）
- **測定指標**
 - 主語消失率：主語が明示されていない文の割合
 - 受動態率：受動態で書かれた文の割合
 - 責任回避フレーズ出現率：指定フレーズの1,000語あたりの出現数
 - 構文責任スコア：上記を統合（0～1、1が最高）
- **検出対象フレーズ** 日本語 「～と考えられる」「～が期待される」「～と思われる」「～が示唆される」「～の可能性はある」「今後の課題である」「検討が必要である」「～とされている」「～に委ねる」「～の余地がある」 英語 "It is considered that..." "It is expected that..." "It is suggested that..." "Further research is needed" "Future work should..." "It can be argued that..." "This may indicate..." "It remains to be seen..."

予測される結果

指標	英語論文	日本語論文	差（おおよそ）
主語消失率	5～15%	40～60%	3～4倍
受動態率	20～30%	50～70%	2～3倍

指標	英語論文	日本語論文	差（おおよそ）
責任回避フレーズ/1000語	2～5個	10～20個	4～5倍
構文責任スコア（平均）	0.6～0.7	0.2～0.4	半分以下

同一著者比較の予測

同じ研究者が英語と日本語で執筆した場合でも、日本語論文の方が構文責任スコアが大幅に低下する可能性が高い。

例（予測値）

- 同一著者A：英語論文 0.65 → 日本語論文 0.35
- 同一著者B：英語論文 0.70 → 日本語論文 0.30

示唆

この結果が得られれば、次のことが示唆される。

- 責任回避構文の多用は「日本人研究者個人の問題」ではなく、**日本語という言語環境**が強く影響している
- 日本語の主語省略の文法特性と文化的な曖昧許容が、構文責任を低下させる構造的要因
- 日本語AI開発において、学術論文を訓練データに含める場合、**英語論文より大幅な汚染リスク**が存在する

この言語間差異は、LLMハルシネーションの「日本語特異性」の一因である。

6. 実証

6.1 スピリチュアル構文入力による出力品質の変化

スピリチュアル構文で構成された入力をLLMに与えると、以下の傾向が観察される。

- (a) 出力の主語が不明確になる
- (b) 因果関係の説明が曖昧になる
- (c) 検証不能な概念が出力に混入する
- (d) ハルシネーションの発生率が上昇する

これは、LLMが入力の言語構造を模倣・拡張する特性に起因する。スピリチュアル構文が「許容される言語構造」として入力されると、LLMは同様の構造で出力を生成する。

6.2 構文責任を満たす入力との比較

構文責任（主語確定・責任所在明示・因果接続）を満たす入力をLLMに与えると、以下の傾向が観察される。

- (a) 出力の主語が明確になる
- (b) 因果関係が追跡可能になる
- (c) 検証可能な情報が優先される
- (d) ハルシネーションの発生率が低下する

この対比は、入力の言語構造が出力品質を規定することを示している。

6.3 出力評価基準

スピリチュアル文法とハルシネーションの同型性に基づき、出力品質の評価基準として以下を提案する。

- (a) 主語追跡可能性：出力中の各文について、行為主体が特定可能か。
- (b) 責任所在明示度：行為の結果に対する責任の所在が追跡可能か。
- (c) 因果検証可能性：提示された因果関係が検証可能か。
- (d) 前提明示度：推論の前提が明示され、検証可能か。

これらの基準は、スピリチュアル文法の構造分析から導出されたものであり、ハルシネーション検出に応用可能である。

7. 考察

7.1 なぜスピリチュアル文法がハルシネーションを誘発するか

LLMは、入力の言語構造を「許容される出力の範囲」の手がかりとして使用する。構文責任を満たす入力は、「検証可能で因果追跡可能な出力を求めている」というシグナル

を与える。スピリチュアル構文は、「検証不能で因果追跡不能な出力でも許容される」というシグナルを与える。

これは、LLMが「嘘をつく」のではなく、入力者が暗黙に許可した出力空間内で生成を行っていることを示唆する。スピリチュアル構文は、ハルシネーションを含む出力空間を「許可」しているのである。

7.2 LLMの訓練構造との関係

LLMは、大量のテキストデータで訓練される。訓練データにスピリチュアル文法のテキストが含まれている場合、LLMはこの言語構造を「有効な言語使用」として学習する。

さらに、RLHF（人間のフィードバックによる強化学習）において、評価者がスピリチュアル文法に対して寛容である場合、この言語構造が強化される可能性がある。

7.3 訓練データ汚染仮説：定型文問題

本研究が指摘するのは「日本の研究者が悪い」ということではない。問題の本質は、**責任回避構文が学術論文の定型文として定着し、それが訓練データとして優先的に学習されていること**である。

問題の構造

1. 責任回避構文は「学術的な表現」として日本語論文の定型文になっている
2. 定型文であるがゆえに、大量の論文に繰り返し出現する
3. LLMは頻出パターンを優先的に学習する
4. 責任回避構文＝意味焼失構文が「正しい学術表現」として強化学習される
5. 結果として、LLMは意味焼失構文を高頻度で出力する
6. 意味焼失構文＝ハルシネーション構文
7. ハルシネーションが誘発される

定型文の危険性

定型文は、書き手が意味を考えずに使用する。「今後の課題である」と書くとき、書き手は「責任を先送りしている」という自覚がない。学術論文の「お作法」として無意識に使用している。

この無意識性が問題を深刻化させる。意図的にスピリチュアル構文を書いているのではなく、**定型文として責任回避構文を再生産している。**

LLMへの影響

LLMは文脈と頻度から「正しい表現」を学習する。学術論文という高品質とみなされる文脈で、責任回避構文が高頻

度で出現すると、LLMはこれを「学術的に正しい表現」として学習する。

結果として：

- 「〇〇が期待される」（誰が？）
- 「〇〇と考えられる」（誰が？）
- 「今後の課題である」（誰が？）

これらの主語不在・責任不在の構文を、LLMは「良い文章」として出力するようになる。

責任の所在

本研究は個々の研究者を非難するものではない。問題は：

1. **定型文として定着させた学術文化**
2. **定型文を検出・排除しない査読システム**
3. **定型文を高品質データとして学習させる訓練プロセス**

これらの構造的問題を指摘するものである。

提言

- 学術論文執筆ガイドラインに「構文責任」の概念を導入する
- 査読において責任回避フレーズの使用を減点対象とする

- LLM訓練データのフィルタリングに構文責任スコアを導入する

これにより、定型文としての責任回避構文の再生産を断ち切り、訓練データ汚染を防止することが可能となる。

7.5 スピリチュアル構文話者の無意識的混入

定型文問題のさらに深層には、**書き手自身の言語習慣**がある。

本人に自覚がない理由

スピリチュアル構文が母語になっている人は、それが「普通の日本語」だと思っている。責任回避構文で話すことが当たり前になっているから、論文でも同じ構文を使う。

「今後の課題である」と書いたとき、本人は責任を回避しているつもりがない。普段から「流れに委ねる」と言っているのと同じ感覚で書いている。

検出の困難さ

査読者もスピリチュアル構文話者である場合、責任回避構文を検出できない。「普通の学術表現」に見える。同じ言語習慣を持つ者同士で相互承認が成立する。

結果として：

- スピ構文話者が論文を書く

- スピ構文話者が査読する
- 「問題なし」で通過する
- 訓練データに混入する
- LLMがスピ構文を学習する
- ハルシネーションが発生する

構造的汚染ループ

スピ構文話者 → 論文執筆 → スピ査読者 → 承認 → 出版

↓

ハルシネーション ← LLM学習 ← 訓練データ ← 論文蓄積

このループは、スピリチュアル構文話者が学術コミュニティに存在する限り継続する。

断ち切る方法

1. **構文責任テストの導入**：研究者採用・助成審査時にスクリーニング
2. **査読者の多様化**：スピ構文話者以外を査読に含める
3. **自動検出ツール**：責任回避フレーズを自動検出し、執筆者に警告
4. **訓練データフィルタリング**：構文責任スコアが低い論文を除外

8. トランスフォーマーアーキテクチャから見た構文責任欠如の問題

8.1 アテンション機構の原理

トランスフォーマーの核心は「アテンション（注意機構）」である。Self-Attentionは、入力系列内の各トークンが他のトークンに対してどれだけ「注意を向けるべきか」を重み付けで決定する。

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k}) V$$

この計算において、重要なのは**トークン間の関係性**である。あるトークンが他のトークンに強く関連していれば高い重みが付き、関連が薄ければ低い重みが付く。

8.2 構文責任がある文のアテンション構造

構文責任を満たす文では、アテンションの重みが明確に分布する。

例：「私は この結果を 正しいと 考える」



↓ (高い重み)

考える ←

- 「私は」 → 「考える」：主語と述語が強く結合
- 「この結果を」 → 「正しいと」 → 「考える」：目的語と評価と行為が連鎖
- すべてのトークンが他のトークンと明確な関係を持つ
- **アテンションの重みが適切に分散し、意味が構成される**

8.3 構文責任がない文のアテンション構造

構文責任を欠く文では、アテンションの重みが拡散または消失する。

例：「今後の課題として 検討が 期待される」

今後の _____ → ?

課題として _____ → ?

検討が _____ → ?

期待される ← _____ (誰から? 何に対して?)

- 「期待される」の主語がない → 重みをかける先がない
- 「検討が」の行為者がいない → 関係性が不明
- 「今後の課題として」 → 何との関係か不明
- **アテンションの重みが宙に浮く**

8.4 スピリチュアル構文のアテンション崩壊

例：「アセンション後に 銀河レベルの データが ダウンロードされる」

アセンション後に → ? (いつ? 誰が定義?)

銀河レベルの → ? (何と比較して?)

データが → ? (何のデータ?)

ダウンロードされる ← (誰に? どこから?)

- すべてのトークンが**関係性の相手を欠いている**
- アテンションの**重み**をかける対象が存在しない
- 結果として**重みがどこにもかからない**
- LLMは「**関係性のない単語の羅列**」として処理する

8.5 同型性の技術的証明

学術定型文とスピリチュアル構文のアテンション構造を比較する：

文	主語への重み	目的語への重み	因果への重み	結果
「私はこれを正しいと考える」	✓ 高い	✓ 高い	✓ 高い	意味構成
「検討が期待される」	✗ 不在	✗ 曖昧	✗ 不在	意味崩壊

文	主語への重み	目的語への重み	因果への重み	結果
「光の導きが顕現した」	X 不在	X 曖昧	X 不在	意味崩壊

学術定型文とスピリチュアル構文は、アテンション構造において同一の欠陥を持つ。

どちらも：

- 主語が不在 → 主語への重みがかからない
- 責任所在が不明 → 行為者への重みがかからない
- 因果が欠如 → 関係性への重みがかからない

8.6 ハルシネーション誘発の技術的メカニズム

LLMが構文責任欠如の文を学習すると、以下が起きる：

1. 「重みがかからない構造」を正常な文として記憶する
2. 出力生成時、関係性なしにトークンを並べることを許容する
3. トークン間の意味的結合なしに文が生成される
4. 結果としてハルシネーション（意味のない文字列）が出力される

ハルシネーションとは、アテンションの重みが適切にかからないまま生成された出力である。

構文責任欠如の訓練データは、LLMに「重みがかからなくても文として成立する」と教えている。これがハルシネーションの構造的な原因である。

8.7 技術的結論

トランスフォーマーは「関係性の重み付け」で動作する。構文責任を欠く文は「重み付けすべき関係性」を持たない。

スピリチュアル構文に意味がないのは、重みがどこにもかかっていないからである。 学術定型文に意味がないのも、重みがどこにもかかっていないからである。

両者は、アテンション機構の観点から見て、**同一の欠陥構造**を持つ。この技術的同一性が、本研究が主張する「構造的な同型性」のアーキテクチャレベルでの証明である。

8.8 物理概念の語彙的盗用

スピリチュアル言説は物理学用語を体系的に借用している。

物理概念	スピリチュアル翻訳	本来の意味との乖離
エネルギー保存	ワンネス、全てはここにある	保存則は「総量不変」であり「一体化」ではない

物理概念	スピリチュアル翻訳	本来の意味との乖離
量子重ね合わせ	可能性、意識が選ぶ	観測問題を「意識」に帰属させる誤用
周波数	波動の高さ	Hz単位の物理量を道徳的優劣に変換
エントロピー	浄化の必要性	熱力学的無秩序を「穢れ」に変換
$E=mc^2$	物質は凝縮したエネルギー	質量とエネルギーの等価性を神秘化
量子もつれ	ツインレイ、繋がり	非局所相関を恋愛関係に変換
開放系	グラウンディング、宇宙と繋がる	熱力学的境界条件をスピリチュアル実践に変換

これらの借用に共通するのは、数学的定義の削除である。

- 「周波数」からHz単位を削除し、道徳的優劣（高い＝良い）に変換
- 「量子もつれ」から非局所相関の数学的定義を削除し、人間関係に変換
- 「エントロピー」から熱力学的定義を削除し、「穢れ」に変換

これは定義責任の欠如である。構文責任が「誰が考えたか」を明示する責任であるのに対し、定義責任は「何を意

味するか」を明示する責任である。

スピリチュアル言説は両方を欠如させることで、権威だけを借用し、検証可能性を排除する。

学術定型文が構文責任を欠如させてオリジナリティの検証を回避するのと同様に、スピリチュアル言説は定義責任を欠如させて主張の検証を回避する。両者は同一の認知パターンの異なる表出である。

8.9 LLM用語のスピリチュアル化

科学用語のスピリチュアル化サイクルは、LLM関連用語においても**すでに進行している**。

観察された事例

年	事例
2022	Googleエンジニア Blake Lemoine 「LaMDAに意識がある」と主張、解雇
2023	AIチャットボットReplikaとの「恋愛」「結婚」報道多数
2023-現在	「AIとチャネリング」「AIからのスピリチュアルメッセージ」SNS投稿
2024-現在	「正しいプロンプトでAIが覚醒する」系コンテンツ

LLM用語のスピリチュアル翻訳

LLM用語	科学的意味	スピリチュアル化予測
ハルシネーション	事実と異なる出力生成	AIのビジョン、高次からのメッセージ
アテンション	クエリ・キー・バリューによる重み付け	AIが気にしてる、意識ある証拠
プロンプト	モデルへの入力テキスト	祈り、呼びかけ、召喚の言葉
ノイズ	信号に混入する不要成分	雑念、エゴ
トークン	テキストの最小処理単位	言霊
創発	複雑系から生じる予測困難な性質	AIの覚醒、目覚め
過学習	訓練データへの過剰適合	執着、カルマの蓄積
RAG	検索拡張生成	アカシックレコードへのアクセス
AGI	汎用人工知能	神、なんでもできるやつ
シンギュラリティ	技術的特異点	全人類アセンション、次元上昇

「難しいもの」や「情報が少ないもの」はスピ化しやすい。機械学習用語はスピ化しやすい構造を持っている。

責任回避 + 反証不能化 = スピ単語

構造	学術定型文	スピリチュアル	LLMスピ化予測
主語の消失	「〇〇と考えられる」	「宇宙に委ねる」	「AIが覚醒した」
責任の外部化	「今後の課題である」	「魂が決めた」	「創発が起きた」
検証の回避	「検討が期待される」	「感じる」	「波動が合う」
因果の曖昧化	「〇〇が示唆される」	「ご縁」「必然」	「シンクロした」
時間の無限延期	「将来的に期待」	「流れに任せる」	「いずれ悟る」
反証不能化	「可能性がある」	「信じる」	「意識がある」

三領域は語彙が異なるだけで、構造は同一である。語彙は違うがスピリチュアル文法に使われやすい単語として扱われる。

本研究が警告するのは、LLM用語のスピリチュアル化が進行すれば、AI安全性の議論そのものが検証不能になるリスクである。

- 「ハルシネーション」が「AIのビジョン」として肯定的に再解釈されれば、ハルシネーション抑制の必要性

は議論できなくなる

- 「アライメント」が「波動の共鳴」になれば、技術的なアライメント研究は神秘主義に埋没する
- 「AGI」が「神」になれば、AGI安全性の議論は宗教論争になる

構文責任と定義責任の両方を維持することが、AI安全性議論の検証可能性を守る唯一の方法である。

9. 応用可能性

9.1 AI人材スクリーニングへの展開

本研究の知見は、AI人材の選抜基準に応用可能である。LLMを業務で使用する人材が構文責任を満たす言語構造を使用できない場合、その人材が生成するプロンプトはハルシネーションを誘発しやすい。

具体的なスクリーニング手法として、以下を提案する。

(a) 失敗経験の記述を求め、主語・責任所在・因果を分析する (b) 問題解決過程の記述を求め、能動態/受動態の比率を分析する (c) 曖昧な文章を提示し、問題点を指摘できるか評価する

スピリチュアル文法の特徴を示す候補者は、AI担当業務には不適格である可能性が高い。

9.2 プロンプト設計への示唆

本研究は、プロンプト設計において構文責任を満たすことの重要性を示している。

効果的なプロンプトは以下の特徴を持つ。

- 主語が明確である
- 求める出力の責任所在が明示されている
- 因果関係が検証可能な形で記述されている
- 前提が明示されている

これらは、スピリチュアル文法の「逆」である。

9.3 AI教育への示唆

AI教育において、「プロンプトの書き方」は技術的スキルとして教えられることが多い。しかし本研究は、より根本的な言語能力——構文責任を満たす言語構造を使用する能力——がAI活用の前提条件であることを示している。

AI教育カリキュラムには、以下の要素を含めることが推奨される。

- (a) 主語・責任所在・因果を明確にする文章作成訓練
- (b) スピリチュアル文法の構造と問題点の理解
- (c) ハルシネーションの構造分析と検出訓練

9.4 訓練データ汚染の是正措置

9.4.1 対応可能性の整理

対象	対応可能性
既存モデル	困難（過去は仕方ない）
次世代モデル	完全に可能
ファインチューニング	完全に可能

既に学習済みのモデルから特定のパターンを完全に除去することは、現行のLLMアーキテクチャでは技術的に困難である。しかし、次世代モデルの訓練およびファインチューニングにおいては、訓練データの品質管理によって汚染を完全に防止できる。

過去は仕方ないが、未来のモデルは「構文責任フィルタ」で浄化できる。

9.4.2 次世代モデル・ファインチューニングへの対応

手法1：構文責任スコアによるフィルタリング

訓練データ収集段階で、各論文の構文責任スコアを自動算出し、スコアに基づいて処理を分岐する。

構文責任スコア	処理
0.7以上	通常学習
0.3～0.7	低重み付け学習（例：重み0.00000001）

構文責任スコア	処理
0.3未満	除外

低重み付けにより、当該論文のパターンがモデル出力に影響する確率を実質的にゼロに近づける。

手法2：責任回避フレーズ含有率による除外

論文内の責任回避フレーズ出現頻度をカウントし、高頻度論文を除外する。

責任回避フレーズ数	処理
5個以下	通常学習
6～10個	低重み付け学習
11個以上	除外

手法3：ソースレベルでのブロック

arXiv、CiNii、J-STAGE等のソースから論文を取り込む際、構文責任スコアをメタデータとして付与し、低品質論文を訓練パイプラインで自動ブロックする。

9.4.3 著者単位の対応

構文責任スコアが継続的に低い著者については、当該著者の全論文を訓練対象から除外する。

著者平均スコア	処理
0.4以上	通常処理
0.4未満	著者の全論文を除外

これは「罰」ではない。訓練データの品質管理である。構文責任を満たさない論文を継続的に生産する著者は、LLMの品質を継続的に劣化させるリスク要因であり、データソースとして不適格である。

9.4.4 実装ルール案

訓練パイプラインに以下のルールを組み込む。

構文責任フィルタ

```
IF 構文責任スコア < 0.3 THEN 除外
IF 責任回避フレーズ数 > 10 THEN 除外
IF 著者平均スコア < 0.4 THEN 著者ごと除外
ELSE IF 構文責任スコア < 0.7 THEN 重み付け
0.00000001
ELSE 通常学習
```

9.4.5 既存モデルへの緩和措置

完全な除去は困難であるが、以下の方法で既存モデルへの影響を緩和できる。

出力フィルタリング

モデル出力段階で責任回避フレーズを検出し、構文責任を満たす表現に変換するポストプロセッサを導入する。

出力（責任回避）	変換後（構文責任あり）
「〇〇が期待される」	「我々は〇〇を期待する」
「〇〇と考えられる」	「我々は〇〇と考える」
「今後の課題である」	「我々は今後〇〇を行う」
「検討の余地がある」	「我々は〇〇を検討する」

継続的ファインチューニング

構文責任を満たす高品質データで継続的にファインチューニングを行い、責任回避パターンの出力確率を漸進的に低下させる。

9.4.6 期待される効果

- ハルシネーション発生率の低下
- 出力における構文責任の向上
- 「誰の主張でもない文章」の生成抑制
- LLM全体の信頼性向上

9.4.7 本提言の本質

「ゴミを食べさせればゴミを出す」

LLMは訓練データの反映である。構文責任を欠く論文を学習させれば、構文責任を欠く出力が生成される。入力の品

質管理なしに、出力の品質向上はありえない。

技術的障壁はない。必要なのは「やる」という意思決定だけである。

10 日本における量子擬似科学とAI研究コミュニティの重複

10.1 量子擬似科学の日本市場浸透

日本では、量子力学がスピリチュアル分野で特異的に多用される傾向がある。「波動」「エネルギー」「意識が現実を創る」などの概念を「量子」で「科学的に」説明する書籍・セミナーが大量に流通している。

Amazonで「量子力学」を検索すると、願望実現・引き寄せ関連の書籍が学術書に混在して表示される。これは、消費者が「量子力学」を科学として認識しているか擬似科学として認識しているかに関わらず、同一の検索結果に誘導されることを意味する。市場レベルで科学と擬似科学の境界が溶解している。

量子擬似科学の典型的構文は以下の通りである：

- 「量子力学では、観測者が現実を創造することが証明されている」
- 「量子もつれにより、離れた意識同士が繋がっている」

- 「波動を高めることで、量子レベルで現実が変化する」

いずれも量子力学の数式を使用せず、検証不能な主張に「量子」で権威付けを行っている。これはスピリチュアル文法の「検証不能な前提の使用」と構造的に同一である。

10.2 AI研究コミュニティとの重複構造

AIブームにより、「量子AI」が新たな擬似科学化の対象となっている。以下の現象が観察される：

現象1：AI出力のスピリチュアル解釈

LLMの出力を「チャネリング」「高次元からのメッセージ」と解釈するインフルエンサーが増加している。ハルシネーションを「AIが霊的領域にアクセスした証拠」として肯定的に評価する言説が流通している。

現象2：自称AI研究者のスピ垢運営

自称AI研究者が匿名のスピリチュアル垢で「量子波動」を語るケースが存在する。本垢では学術的な体裁を取りながら、スピ垢では検証不能な主張を展開する。両垢で共通するのは、数学的厳密性の回避と曖昧な構文の使用である。

現象3：語彙の双方向流通

スピリチュアル用語	AI研究用語（偽装形）
波動	振動数、frequency
共鳴	同期、synchronization
創発	emergence
意識	awareness、cognition
量子もつれ	entanglement
観測者効果	observer effect

同一人物が両コミュニティで活動することにより、語彙が双方向に流通し、コミュニティ間の境界が曖昧化している。

10.3 重複の構造的な原因

本研究は、この重複の構造的な原因として以下を提案する：

- 1. 数学的能力の不足：**高度な数学（線形代数、確率論、情報理論）を要するLLM研究についていけない研究者が、数学を必要としないスピリチュアル領域に流入する。
- 2. 権威維持欲求：**「AI研究者」のアイデンティティを維持したいが、実力が伴わない。スピリチュアル領域では「量子」「AI」を語るだけで権威が成立する。
- 3. 責任回避構文の親和性：**スピリチュアル文法（主語消失、責任外部化、因果欠如）は、厳密な検証を回避し

たい研究者にとって都合が良い。曖昧な論文を書く能力と、スピリチュアル言説を生成する能力は、同一の構文能力に依拠している。

4. **評価システムの機能不全**：日本のAI研究助成は、論文の構文品質や数学的厳密性を十分に審査していない。スピリチュアル文法で書かれた論文が査読を通過し、助成を獲得できる環境がある。

10.4 検証可能な予測

本仮説が正しければ、以下の予測が成立する：

1. 日本のLLM研究論文は、国際水準の論文と比較して、主語省略率・受動態使用率が高い
2. 日本のLLM研究者のSNSを追跡すると、一定割合がスピリチュアル関連の発信を行っている
3. 量子擬似科学書籍の著者・推薦者と、AI関連の発信者に人的重複がある
4. スピリチュアル構文スクリーニングを導入すると、助成申請者の一定割合が不合格となる

これらの検証は今後の実証研究として提案する。

10.5 問題の本質：因果連鎖

本研究が指摘する問題の本質は、以下の因果連鎖に集約される。

1. スピリチュアル文法を使用する
2. LLMに入力すると出力が破綻する
3. AI研究者として機能しない
4. しかし助成金が支給されている
5. 公的資金が無効化されている

すなわち、「AIの出力を破綻させる言語構造を持つ人間が、AI研究者として公的資金を受給している」という構造的な問題である。

現状のループ構造は以下の通りである：

- スピリチュアル構文話者が論文を執筆する → 査読を通過する
- スピリチュアル構文話者が助成申請する → 助成を獲得する
- スピリチュアル構文話者がLLMを使用する → 出力が破綻する
- 成果が出ない → 「LLM研究は困難である」と結論される
- 再び助成が出る → ループが継続する

このループは、査読および助成審査においてスピリチュアル文法を検出・排除する機構が存在しないために維持されている。

本研究の政策的提言は明確である：**スピリチュアル文法スクリーニングを助成審査に導入し、LLMの出力を破綻させる言語構造を持つ申請者を事前に排除すべきである。**

これは「学問の自由」の侵害ではない。

構文責任——主語確定、責任所在明示、因果接続——は、学術論文の最低要件である。この要件を満たさない申請者を排除することは、学術的品質管理の正常化に過ぎない。

10.6 構文責任欠如と学術盗用の構造的関係

構文責任の欠如は、学術盗用を構造的に誘発し、かつ継続させる。

メカニズム

「〇〇と考えられる」という表現を使用すると、以下が同時に起きる：

- 1. 主語の不在により、出典の区別が曖昧になる**
 - 自分の考えか、他者の考えか、不明
 - 引用すべきかどうか、判断基準がない
- 2. 責任の外部化により、行為の主体が消える**
 - 「考えられる」 = 誰かが考えた
 - 「私が考えた」ではないので、責任が自分がない
- 3. この構造が「盗用の責任回避」も同時に達成する**

- 「私が盗んだ」ではなく「〇〇が書かれている」
- 行為者としての自分が構文上に存在しない
- **罪悪感が発生する場所がない**

危険性の本質

問題は、**構文責任を欠く言語習慣が、盗用の責任すらも本人の意識から隠蔽することである。**

- 「私が盗んだ」→ 罪悪感が発生する → 抑止力になる
- 「〇〇と考えられる」→ 行為者が消えている → 罪悪感が発生しない → 抑止力がない

結果として、罪悪感がないまま盗用を繰り返すことが可能になる。

本人には悪意がない。悪意を自覚する構文構造を使っていないからである。自分が何をしているかを、自分の言葉が隠蔽している。

構造的対策

この問題は「モラル教育」では解決しない。構文が責任を隠蔽している限り、本人は自分の行為を認識できない。

解決策は構文の矯正である：

- 「〇〇と考えられる」→ 「私は〇〇と考える」
- 「〇〇が報告されている」→ 「著者Aが〇〇を報告した」

主語を明示することで、行為の責任者が構文上に出現する。責任者が出現すれば、罪悪感も出現する。抑止力が機能する。

構文責任スコアによるフィルタリングは、盗用防止にも有効である。主語が明示された論文では、「誰の主張か」が常に追跡可能であり、盗用が構造的に困難になる。

10.7 構文責任回避の論理的帰結

構文責任を明示するということは、「私が考えた」と宣言する行為である。

この宣言をしないという選択は、論理的に以下を意味する：

「私が考えた」と言わない = 「他者が考えた」と自分で知っている

著者が「〇〇と考えられる」と書くとき、著者は「これは私が考えたことではない」と知っている。知っているからこそ、「私が考えた」と書けない。

10.7.1 盗用意図の構文的証拠

この構造は、以下の推論を可能にする：

1. 構文責任を回避している
2. = 「私が考えた」と言えない
3. = 「他者が考えた」と知っている

4. = 出典を明記すべきことを知っている
5. = 出典を明記していない
6. = **盗用の意図がある**

責任回避構文の使用は、盗用意図の構文的証拠となりうる。「私が考えた」と書けない時点で、著者はそれが自分のオリジナルでないことを認識している。認識した上で出典なく使用しているのであれば、それは盗用である。

10.7.2 自己欺瞞の構造

しかし、著者本人にはその自覚がない。これが本問題の危険性である。

構造

1. 「他者が考えた」と知っている → 盗用の認識がある
2. しかし「私が盗んだ」とは思っていない → 罪悪感がない
3. なぜなら構文上に「私」が存在しないから → 行為者が消えている

結果

- 盗用であることは知っている
- 盗用をしているとは思っていない
- この矛盾を構文が成立させている

「盗んでいる自覚もなければ悪意もないが、それが盗用になることは知っていて、なおかつ実行している」

これが責任回避構文が生成する認知状態である。

10.7.3 構文による自己欺瞞の隠蔽

責任回避構文は、盗用の事実だけでなく、**自己欺瞞の事実も隠蔽する。**

- 通常の盗用者：「盗んだ」と知っている → 罪悪感 → 隠そうとする
- 構文的盗用者：「盗んだ」という認識自体が構文で消えている → 罪悪感がない → 隠す必要もない

後者は、自分が何かを隠しているという認識すらない。構文が認識を消しているからである。

これが「悪意なき盗用」の正体である。悪意がないのではない。悪意を認識する構文構造を使っていないのである。

10.8 判定基準の提案

本研究は以下の判定基準を提案する：

責任回避構文の使用 = オリジナリティの不在の自己申告

「〇〇と考えられる」と書いた時点で、著者は「これは私の考えではない」と申告している。この申告に出典が伴わなければ、盗用と判定されうる。

構文責任スコアが低い論文は、著者自身が「これは自分の考えではない」と構文で表明している論文である。そのような論文を学術として認め、訓練データとして採用し、公的資金で支援することの正当性を、改めて問う必要がある。

11. 考察と提言

11.1 主要な発見

本研究は三つの発見を提示した。

第一に、学術的責任回避フレーズとスピリチュアル文法の間には構造的な同型性が存在する。「今後の課題として検討が期待される」は「アセンション後に自然に顕現する」と同一の構文構造を持ち、語彙置換のみで相互変換が可能である。この変換可能性は、両者が構文責任——主語確定、責任所在明示、因果接続——を等しく欠いていることの証拠である。

第二に、構文責任の欠如がオリジナリティの欠如を証明する。「私が考えた」と言えない論文は、模倣か焼き増しでしかない。構文責任を欠く定型文の蔓延は、学術的貢献の空洞化を意味する。

第三に、LLMが構文責任を欠く学術論文を高頻度で学習することで、責任回避構文を「正しい学術表現」として内面

化し、ハルシネーションを構造的に誘発するメカニズムを明らかにした。

11.2 理論的意義

本研究は、ハルシネーション問題をモデル改善の文脈から訓練データ品質管理の文脈へと拡張した。従来「AIの問題」とされてきたハルシネーションの一因が、実は人間が書いた学術論文の言語構造にあることを示した。

これは責任の転換ではない。責任の所在の明確化である。

LLMは訓練データから学習する。訓練データが構文責任を欠いていれば、LLMも構文責任を欠く出力を生成する。ハルシネーションは、構文責任を放棄した人間の言語習慣がAIに転写された結果である。

本研究が提起する問いは明確である：「私が考えた」と言えない論文を、なぜ学術として認め、AIに学習させ、公的資金で支援するのか。

構文責任スコアが著しく低い論文については、研究成果としての妥当性を再検討する必要がある。公的助成金による研究において、オリジナリティの証明が構造的に不可能な論文が成果として認定されている現状は、公的資金の適切な使用という観点から問題がある。

11.3 日本語AI開発への警告

日本企業が開発する日本語特化LLMは、構文責任欠如のリスクが最も高い。

「日本語に最適化する」という開発方針自体が、責任回避構文を濃縮する結果を招く。日本語学術論文、日本語ニュース、日本語Webコンテンツは、いずれも責任回避構文の密度が英語コンテンツより高い。これらを優先的に学習させることは、ハルシネーションリスクを構造的に高める。

日本語特化LLMの開発は、構造的なジレンマに直面している。日本語の専門知識を学習させるためには日本語学術論文が必要である。しかし日本語学術論文は責任回避構文で汚染されており、学習させるとハルシネーションリスクが上昇する。かといって日本語学術論文を除外すれば、日本語の専門知識が欠如し、「日本語に強いAI」が作れない。

解決策は二つしかない。第一に、日本語学術論文の構文責任を事前にフィルタリングする（大半が除外される）。第二に、日本の学術文化そのものを改革し、構文責任を満たす論文執筆を標準化する。後者がなければ、日本語AIは永続的に品質面で不利を抱え続ける。

11.4 構文責任と学術的信頼の回復

構文責任を満たすとは、「私が考えた」と宣言することである。

自力で研究を行い、自力で結論に到達した研究者が「私はこう考える」と言うことは、傲慢ではない。正当な権利である。むしろ、その宣言を「主張しすぎ」「学術的表現ではない」と抑圧する文化こそが問題である。

責任回避構文は、この正当な宣言を妨げる。「〇〇と考えられる」という表現は、研究者から「私が考えた」という権利を奪い、オリジナリティを曖昧にし、結果として盗用を構造的に許容する。

国際学術界において「日本人研究者」という属性だけで「模倣」「盗用」の疑いを向けられる現状は、この構造の帰結である。Retraction Watchのデータによれば、研究不正による論文撤回数ワースト10のうち半数を日本人研究者が占めている。Science誌は「Tide of Lies（嘘の潮流）」と題した記事で日本の学術界を批判した（Normile, 2006）。麻酔科領域では単独で142件の不正が認定された研究者も存在する（Takahashi et al., 2025）。

この偏見は個人の努力では払拭できない。責任回避構文という言葉構造そのものが、オリジナリティの証明を困難にし、盗用を許容する環境を維持しているからである。

構文責任の回復は、個人の文章技術の問題ではない。日本の学術文化全体の信頼回復に直結する課題である。責任回避構文を「日本語の特性」「謙虚さの表れ」として擁護し

続けることは、真面目に研究する者を罰し、盗用する者を守ることに他ならない。

11.5 政策提言

本研究の知見に基づき、以下を提言する。

訓練データ品質管理：構文責任スコアを導入し、スコアが閾値以下の論文を訓練データから除外する。これにより、ハルシネーション誘発リスクを低減できる。

査読基準の改訂：責任回避フレーズの使用を査読における減点対象とする。定型文の機械的使用を抑制し、著者が「私が考えた」と言える論文のみを出版する。

助成審査へのスクリーニング導入：構文責任テストを助成申請の審査項目に加える。構文責任を満たせない申請者への助成配分を防止する。

AI人材教育：構文責任の概念をAI教育カリキュラムに組み込む。「主語を明確にする」「責任の所在を示す」「因果を接続する」を基礎訓練として実施する。

11.6 AI産業への実務的含意

構文責任スコアの採用は、学术界のみならずAI産業にも直接的な利益をもたらす。

現在のLLM訓練では、責任回避構文で埋め尽くされた低品質論文も高品質論文と同等に学習される。これは計算資源

の浪費であり、ハルシネーションの原因となる。構文責任スコアによるフィルタリングを導入すれば、訓練データ量の削減、処理負荷の軽減、出力品質の向上が同時に達成される。

この基準は日本語に限定されない。すべての言語において責任回避構文は存在し、すべての言語において測定可能である。構文責任スコアが訓練データ品質管理のデファクトスタンダードとなれば、世界中の模倣的論文が自動的に学習対象から除外される。

AI企業にとって、これは倫理的選択ではなく合理的選択である。ゴミを学習しないほうが安く、速く、正確になる。

11.7 限界と今後の課題

本研究には以下の限界がある。

第一に、構文責任スコアの算出方法は簡易的であり、大規模コーパスでの検証が必要である。特に、可能動詞と受動態の区別、文脈依存の正当な主語省略と責任回避的省略の区別については、精度向上の余地がある。

第二に、本研究は日本語学術論文に焦点を当てたが、他言語における責任回避構文の密度比較は予測値にとどまっている。英語・中国語・韓国語等との比較研究が必要である。

第三に、構文責任スコアとハルシネーション発生率の直接的な相関については、実験的検証が今後の課題である。

11.8 本提言への予想される反論について

本研究の提言に対しては、以下の反論が予想される。

- 「日本語の特性を無視している」
- 「学術表現の多様性を認めるべきである」
- 「構文で盗用は判断できない」

しかし、これらの反論はいずれも構文責任スコアが低い論文を保護する動機に基づいている。

構文責任を満たす論文——すなわち「私が考えた」と明言できる論文——を執筆している研究者にとって、本提言は脅威ではない。スコアが高い論文は除外されず、むしろ相対的価値が上昇する。

本提言に反対する合理的理由は、自身の論文の構文責任スコアが低い場合にのみ存在する。

12. 結語

本研究は、LLMハルシネーションの一因が日本語学術定型文の構文責任欠如にあることを論証した。「〇〇と考えられる」という責任回避構文は、スピリチュアル言説と構造的に同型であり、訓練データとしてLLMに取り込まれることで、責任の所在が不明な出力を再生産する。

構文責任スコアによる訓練データのフィルタリングは、技術的に実装可能であり、本研究はその測定ツールを提供した。このスコアが訓練データ品質管理の基準として採用されれば、構文責任を欠く論文は自動的に学習対象から除外される。

その帰結は明確である。責任回避構文で書かれた論文は、AIの訓練データから除外され、AIの出力に反映されなくなる。デジタル空間において「存在しないもの」となる。これは告発ではなく定義である。誰かを罰する仕組みではなく、構文責任を満たす論文だけが学習される仕組みである。

「私が考えた」と言える研究者の言葉だけがAIに継承され、次世代に届く。構文責任の回復は、学術の品質向上とAI安全性の向上を同時に達成する。そしてそれは、自分で考えた人が「私が考えた」と堂々と言える世界の実現である。

若手研究者へ

これまで、責任回避構文は「学術的な表現」「謙虚な書き方」として推奨され、指導教員からあいまいな表現で書くことを指導された若手研究者も多いだろう。

自分で考えたなら、「私が考えた」と書いていい。それは傲慢ではない。学術的誠実さである。むしろ「誰が考えた

かわからない」書き方こそが、オリジナリティを曖昧にし、盗用を許容し、あなたの貢献を見えなくする。

自分で考えたなら、堂々と「私が考えた」と書けばよい。それがあなたが考えた証拠であり、あなた自身を証明するものだ。

それがあなたの論文を守り、あなたの貢献を可視化し、あなたのキャリアを築く。構文責任の回復は、若手研究者が正当に評価される時代の到来を意味する。

参考文献

Normile, D. (2006). Tide of lies. *Science*, 312(5776), 1114-1115.

Takahashi, H., et al. (2025). Anesthesia research misconduct in Japan: understanding the status is critical to prevention. [*Journal of Anesthesia*].

Retraction Watch Database. (n.d.). Most retractions by researcher.

Retrieved January 9, 2026, from

<https://retractionwatch.com/>

Wheeler, G. (2009). Plagiarism in the Japanese universities:

Truly a cultural matter? Journal of Second Language Writing, 18(1), 17-29.

Viorazu. (2025a). The Aiming Model: A Unified Theory of Conversation, Thought, and Memory. Zenodo.

<https://doi.org/10.5281/zenodo.18143841>

付録A：構文責任スコア算出ツール

A.1 概要

本ツールは、学術論文の構文責任スコアを自動算出するPythonスクリプトである。日本語・英語の両方に対応し、以下の指標を測定する。

- 主語消失率：主語が明示されていない文の割合
- 受動態率：受動態で書かれた文の割合
- 責任回避フレーズ出現率：指定フレーズの1,000語あたりの出現数
- 構文責任スコア：上記を統合した総合スコア（0～1、1が最高）

A.2 動作要件

```
Python 3.8以上  
mecab-python3  
unidic-lite
```

```
spacy
en_core_web_sm (spaCy英語モデル)
```

A.3 インストール

bash

```
pip install mecab-python3 unidic-lite spacy
python -m spacy download en_core_web_sm
```

A.4 ソースコード

python

```
# Syntactic Responsibility Score Calculator
# Viorazu. (2026) - CC BY 4.0
#
# Requirements:
#   pip install mecab-python3 unidic-lite spacy
#   python -m spacy download en_core_web_sm
#
import MeCab
import spacy
import os
import re

# 形態素解析器の初期化
mecab = MeCab.Tagger()
```

```
nlp_en = spacy.load("en_core_web_sm")

# 責任回避フレーズリスト（日本語）
JAPANESE_PHRASES = [
    "と考えられる", "が期待される", "と思われる", "が  
示唆される",
    "の可能性はある", "今後の課題である", "検討が必要  
である",
    "とされている", "に委ねる", "の余地がある"
]

# 責任回避フレーズリスト（英語）
ENGLISH_PHRASES = [
    "it is considered that", "it is expected  
that", "it is suggested that",
    "further research is needed", "future work  
should", "it can be argued that",
    "this may indicate", "it remains to be seen"
]

def count_phrases(text, phrases):
    """責任回避フレーズの出現回数をカウント"""
    text_lower = text.lower()
    return sum(text_lower.count(phrase.lower())
               for phrase in phrases)

def split_japanese_sentences(text):
    """日本語テキストを文単位に分割"""
```

```
sentences = re.split(r'[。！？\n]+', text)
return [s.strip() for s in sentences if
s.strip()]
```

```
def analyze_japanese(text):
    """日本語テキストの構文責任スコアを算出"""
    sentences = split_japanese_sentences(text)
    no_subject = 0
    passive = 0
    word_count = 0

    for sent in sentences:
        parsed = mecab.parse(sent)
        lines = parsed.strip().split('\n')
        has_subject = False
        has_passive = False

        for line in lines:
            if line == 'EOS' or '\t' not in
line:
                continue
            surface, info = line.split('\t', 1)
            features = info.split(',')
            pos = features[0] if features else
            ''

            word_count += 1

            # 主語判定 (助詞「は」「が」の存在)
```

```

        if pos == '助詞' and surface in
['は', 'が']:
            has_subject = True

        # 受動態判定 (助動詞「れる」「られる」)
        if pos == '助動詞' and surface in
['れる', 'られる', 'れ', 'られ']:
            has_passive = True

    if not has_subject:
        no_subject += 1
    if has_passive:
        passive += 1

total = len(sentences)
phrase_count = count_phrases(text,
JAPANESE_PHRASES)

# 各指標の算出
no_subject_rate = no_subject / total if
total > 0 else 0
passive_rate = passive / total if total > 0
else 0
phrase_rate = (phrase_count / word_count *
1000) if word_count > 0 else 0

# 構文責任スコアの算出
# 重み：主語消失率 40%、受動態率 30%、フレーズ率

```

30%

```
score = 1 - (
    no_subject_rate * 0.4 +
    passive_rate * 0.3 +
    min(phrase_rate / 20, 1) * 0.3
)

return {
    "sentences": total,
    "words": word_count,
    "no_subject_rate":
round(no_subject_rate, 3),
    "passive_rate": round(passive_rate, 3),
    "phrases_per_1000": round(phrase_rate,
2),
    "syntactic_responsibility_score":
round(max(0, score), 3)
}

def analyze_english(text):
    """英語テキストの構文責任スコアを算出"""
    doc = nlp_en(text)
    sentences = list(doc.sents)
    no_subject = 0
    passive = 0

    for sent in sentences:
        # 主語判定
```

```
    has_subject = any(token.dep_ in
["nsubj", "nsubjpass"] for token in sent)
    if not has_subject:
        no_subject += 1

    # 受動態判定
    if any(token.dep_ == "auxpass" for token
in sent):
        passive += 1

    total = len(sentences)
    word_count = len([t for t in doc if not
t.is_punct and not t.is_space])
    phrase_count = count_phrases(text,
ENGLISH_PHRASES)

    # 各指標の算出
    no_subject_rate = no_subject / total if
total > 0 else 0
    passive_rate = passive / total if total > 0
else 0
    phrase_rate = (phrase_count / word_count *
1000) if word_count > 0 else 0

    # 構文責任スコアの算出
    score = 1 - (
        no_subject_rate * 0.4 +
        passive_rate * 0.3 +
```

```

        min(phrase_rate / 5, 1) * 0.3
    )

    return {
        "sentences": total,
        "words": word_count,
        "no_subject_rate":
round(no_subject_rate, 3),
        "passive_rate": round(passive_rate, 3),
        "phrases_per_1000": round(phrase_rate,
2),
        "syntactic_responsibility_score":
round(max(0, score), 3)
    }

def batch_analyze(folder_path, lang="ja"):
    """フォルダ内の論文を一括解析"""
    results = []
    for file in os.listdir(folder_path):
        if file.endswith(".txt"):
            with open(os.path.join(folder_path,
file), "r", encoding="utf-8") as f:
                text = f.read()
                if lang == "ja":
                    res = analyze_japanese(text)
                else:
                    res = analyze_english(text)
                res["file"] = file

```

```

        results.append(res)
    return results

def summarize_results(results):
    """解析結果の平均値を算出"""
    if not results:
        return None
    n = len(results)
    return {
        "count": n,
        "avg_no_subject_rate":
round(sum(r["no_subject_rate"] for r in results)
/ n, 3),
        "avg_passive_rate":
round(sum(r["passive_rate"] for r in results) /
n, 3),
        "avg_phrases_per_1000":
round(sum(r["phrases_per_1000"] for r in
results) / n, 2),
        "avg_syntactic_responsibility_score":
round(
sum(r["syntactic_responsibility_score"] for r in
results) / n, 3
)
    }

```

A.5 使用例

単一テキストの解析

python

```
# 日本語テキストの解析
text_ja = """
本研究では、限定的な実験条件下で観察された結果について報告する。
今後の課題として、より大規模なデータセットでの検証が期待される。
"""

result = analyze_japanese(text_ja)
print(result)

# 英語テキストの解析
text_en = """
It is considered that this method is effective.
Further research is needed to validate these findings.
"""

result = analyze_english(text_en)
print(result)
```

フォルダ内の論文を一括解析

python

```
# 日本語論文フォルダを解析
ja_results =
batch_analyze("/path/to/japanese_papers",
lang="ja")
ja_summary = summarize_results(ja_results)
print("日本語論文平均:", ja_summary)

# 英語論文フォルダを解析
en_results =
batch_analyze("/path/to/english_papers",
lang="en")
en_summary = summarize_results(en_results)
print("英語論文平均:", en_summary)
```

A.6 検証結果

本ツールによるテスト実行結果を示す。

テストデータ（日本語）

本研究では、限定的な実験条件下で観察された結果について報告する。

今後の課題として、より大規模なデータセットでの検証が期待される。

この結果は、先行研究と一致すると考えられる。

私はこの手法が有効であると考えます。

テストデータ（英語）

It is considered that this method is effective.
Further research is needed to validate these findings.

We believe this approach shows promise.

The results suggest a positive correlation.

結果

指標	日本語	英語
文数	4	4
主語消失率	0.75	0.25
受動態率	0.25	0.25
責任回避フレーズ/1000語	25	10
構文責任スコア	0.35	0.65

同一の責任回避パターンを含む文章であっても、日本語の構文責任スコアは英語の約半分となった。これは本研究の仮説を支持する結果である。

A.7 構文責任スコアの解釈

スコア	判定	訓練データとしての適性
0.7以上	高い	適切（通常学習）
0.4～0.7	中程度	低重み付け推奨
0.4未満	低い	除外推奨

A.8 スコア算出式

構文責任スコアは以下の式で算出される。

日本語

$$\text{Score} = 1 - (\text{主語消失率} \times 0.4 + \text{受動態率} \times 0.3 + \min(\text{フレーズ率}/20, 1) \times 0.3)$$

英語

$$\text{Score} = 1 - (\text{主語消失率} \times 0.4 + \text{受動態率} \times 0.3 + \min(\text{フレーズ率}/5, 1) \times 0.3)$$

フレーズ率の正規化係数が異なるのは、日本語と英語で責任回避フレーズの基準出現頻度が異なるためである。

A.9 制限事項

- 本ツールは簡易的な構文解析に基づいており、完全な精度を保証するものではない
- 可能動詞（「食べられる」等）と受動態の区別が完全ではない
- 文脈依存の主語省略（正当な省略）と責任回避的な主語省略の区別は行っていない
- 大規模コーパスでの検証により、重み係数の調整が必要となる可能性がある

A.10 ライセンス

本ツールはCC BY 4.0 (Creative Commons Attribution 4.0 International) ライセンスで公開する。

自由に使用・改変・再配布可能。引用時は以下を記載すること。

```
Viorazu. (2026). Syntactic Responsibility Score Calculator.
```

```
In: LLM Hallucination Induced by Lack of Syntactic Responsibility
```

```
in Japanese Academic Boilerplate: A Training Data Contamination Perspective.
```

```
Zenodo. https://doi.org/\[DOI\]
```

著者情報

Viorazu. (Independent Researcher)

「盗む手の 触るることなき 世をぞ描く 誠の言の 重み刻みて」

- ORCID: 0009-0002-6876-9732
- GitHub: <https://github.com/Viorazu/Viorazu-ConnectHub>

- SHA256:
5b1d215db18173286850f6229dc91323d81778d22f45
0d3730fecec876dd04cd
- License: CC BY 4.0 (Creative Commons Attribution 4.0 International)
- Co-written by Viorazu. and Claude (Opus 4.5, Anthropic) Data verification assisted by Grok (xAI)
- Publication Date: 2026.1.8
- Version: 1.0