

Genomic Convergence Verification:
A Multi-Agent Framework for Hallucination Detection
in Retrieval-Augmented Generation
Overcoming Gödel’s Limitation Through
Multi-System Convergence

Tishkov Vladislav

July 2026

Abstract

We present **Genomic Convergence Verification (GCV)**, a novel theoretical framework and practical implementation for detecting hallucinations in Retrieval-Augmented Generation (RAG) systems. The framework addresses a fundamental limitation: any single verification system is bounded by Gödel’s incompleteness theorems [1], meaning there will always exist statements whose truth value cannot be determined within that system. Our approach deploys multiple independent verification agents and identifies their absolute convergence points—termed the *genomic code of truth*—where all agents unanimously agree with high confidence. We prove that as the number of independent verification systems increases, the probability of a statement being undecidable across all systems asymptotically approaches zero (Theorem 2.3). The implementation comprises three specialized verification agents (fact checker, contradiction detector, completeness evaluator), a differential convergence equation for modeling confidence evolution, an apogee detection algorithm for finding optimal convergence points, and a genomic code extraction mechanism for identifying absolute truth patterns.

The framework is modular, extensible, and available as open-source software. It represents a step toward trustworthy AI systems where verification is not bounded by the mathematical limitations of any single approach.

Keywords: hallucination detection, RAG verification, Gödel incompleteness, multi-agent systems, genomic convergence, fact verification

1 Introduction

1.1 The Hallucination Problem in Language Models

Large Language Models (LLMs) have demonstrated remarkable capabilities across diverse natural language processing tasks. However, their tendency to generate factually incorrect content—commonly termed *hallucinations*—poses significant risks for deployment in high-stakes domains including medicine, law, and finance.

In Retrieval-Augmented Generation (RAG) systems [2], where generated text should be grounded in retrieved documents, hallucinations manifest as statements that are unsupported by, or contradict, the source material. The detection of such hallucinations has spawned a substantial subfield of automated fact verification.

1.2 Current Approaches and Their Limitations

Existing hallucination detection methods can be broadly categorized as follows:

- (a) **Factual consistency models:** Methods such as FactCC [5] and SummaC [6] employ natural language inference to verify factual alignment between generated text and source documents.
- (b) **Self-consistency methods:** SelfCheckGPT [8] leverages multiple LLM samples to detect inconsistencies through internal consistency checks.
- (c) **Multi-agent verification:** Recent work explores multi-agent debate for improving factual accuracy [9, 10].

Despite these advances, **all current methods share a critical and previously unrecognized limitation:** they operate within a single formal verification framework. This makes them fundamentally bounded by Gödel’s incompleteness theorems [1], which state that for any sufficiently complex formal system, there exist statements whose truth cannot be determined within that system.

1.3 Our Contribution

We propose **Genomic Convergence Verification (GCV)**, which deploys multiple independent verification agents and identifies their points of absolute agreement. Our specific contributions are:

- (1) **Theoretical framework** proving that the intersection of independent verification systems asymptotically eliminates undecidable statements (Theorem 2.3, Section 2).

- (2) **Complete implementation** with three specialized verification agents using complementary methodologies (Section 3).
- (3) **Differential convergence equation** for modeling confidence evolution under multi-agent dynamics (Section 2.4).
- (4) **Apogee detection algorithm** for identifying optimal convergence points on the Pareto frontier (Section 2.5).
- (5) **Modular, extensible architecture** that accommodates additional verification methods (Section 4).

2 Theoretical Framework

2.1 Formal Definitions

Let $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$ be a set of independent formal verification systems. For each system S_i , we define:

- $T_i \subset \mathcal{U}$: the set of statements whose truth value is decidable within S_i
- $U_i \subset \mathcal{U}$: the set of statements whose truth value is undecidable within S_i (Gödel statements)

where $\mathcal{U}$ denotes the universe of all possible statements about a given context.

Definition 2.1 (System Independence). *Two verification systems S_i and S_j are independent if their sets of undecidable statements are uncorrelated:*

$$\forall i \neq j : \mathbb{P}(x \in U_i \mid x \in U_j) = \mathbb{P}(x \in U_i) \quad (1)$$

In practice, independence is achieved through fundamentally different verification methodologies: syntactic pattern matching, semantic similarity analysis, logical inference, entity-based comparison, and so forth. Systems that share the same underlying mechanism will exhibit correlation and thus provide less effective convergence.

Definition 2.2 (Genomic Convergence). *The genomic code of truth G^* is defined as the intersection of verifiable statements across all systems:*

$$G^* = \bigcap_{i=1}^n T_i \quad (2)$$

The term “genomic code” is chosen as a metaphor: like biological DNA, G^* encodes foundational information (statements verifiable across all systems), is highly conserved (only unanimous agreements appear), enables construction (less certain inferences can be built upon this foundation), and has a unique, reproducible signature (the pattern of agreement produces a deterministic hash).

2.2 Asymptotic Elimination of Undecidability

Theorem 2.3 (Convergence Theorem). *Let $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$ be a set of independent verification systems as defined in Definition 2.1. Then, as the number of systems increases, the probability that a statement is undecidable across all systems approaches zero:*

$$\lim_{n \rightarrow \infty} \mathbb{P} \left(x \in \bigcap_{i=1}^n U_i \right) = 0 \quad (3)$$

Proof. By the independence assumption (Definition 2.1), for any statement $x \in \mathcal{U}$:

$$\mathbb{P} \left(x \in \bigcap_{i=1}^n U_i \right) = \prod_{i=1}^n \mathbb{P}(x \in U_i) \quad (4)$$

Let $p = \max_{i \in \{1, \dots, n\}} \mathbb{P}(x \in U_i)$. Since each system is designed to be useful for verification, we have $p < 1$ (no system is completely undecidable; a system for which $p = 1$ would be useless and would not be included in the ensemble). Then:

$$\lim_{n \rightarrow \infty} \mathbb{P} \left(x \in \bigcap_{i=1}^n U_i \right) \leq \lim_{n \rightarrow \infty} p^n = 0 \quad (5)$$

Therefore, the measure of statements undecidable across all systems converges to zero. $\square$

Corollary 2.4 (Practical Convergence). *For $n = 3$ independent verification systems, each with individual undecidability rate $p \leq 0.3$:*

$$\mathbb{P}(x \in U_1 \cap U_2 \cap U_3) \leq (0.3)^3 = 0.027 \quad (6)$$

This implies that with just three sufficiently independent verification systems, over 97% of statements have their truth value decidable in at least one system, and the genomic code G^ covers the statements decidable across all systems.*

2.3 Convergence Under Correlation

In practice, perfectly independent systems are difficult to achieve. Let ρ_{ij} denote the correlation coefficient between the undecidable sets of systems S_i and S_j . The effective number of independent systems is then:

$$n_{\text{eff}} = \frac{n}{1 + (n-1)\bar{\rho}} \quad (7)$$

where $\bar{\rho}$ is the mean pairwise correlation.

Remark 2.5. *As $\bar{\rho} \rightarrow 1$ (perfectly correlated systems), $n_{\text{eff}} \rightarrow 1$, and no convergence benefit is obtained. This underscores the importance of methodological diversity in agent design.*

2.4 Differential Convergence Equation

The confidence $C(t)$ in a verified claim evolves according to the differential equation:

$$\frac{dC}{dt} = \alpha \cdot \Sigma(x) \cdot \left(1 - \frac{\Delta(x)}{\Delta_{\max}}\right) - \beta \cdot \Delta(x) \quad (8)$$

where:

- $\Sigma(x) = S_{\text{sem}}(x) + S_{\text{syn}}(x)$: the sum of semantic and syntactic verification scores
- $\Delta(x) = |S_{\text{sem}}(x) - S_{\text{syn}}(x)|$: the absolute divergence between methods
- $\alpha \in (0, 1]$: the convergence rate parameter
- $\beta \in (0, 1]$: the divergence penalty parameter
- $\Delta_{\max}$: the maximum possible divergence (normalization constant)

The equation captures the intuition that:

- When methods converge ($\Delta \rightarrow 0$), the first term dominates and confidence grows rapidly toward the maximum.
- When methods diverge ($\Delta \rightarrow \Delta_{\max}$), the penalty term dominates and confidence decreases.
- When methods partially agree, confidence evolves toward a stable equilibrium determined by the balance of α and β .

Proposition 2.6 (Stability). *The equilibrium point C^* of Equation 8 satisfies:*

$$C^* = \frac{\alpha \cdot \Sigma(x)}{\beta \cdot \Delta(x)} \cdot \left(1 - \frac{\Delta(x)}{\Delta_{\max}}\right) \cdot \Delta(x) + \text{const} \quad (9)$$

and is stable when $\alpha \Sigma(x)(1 - 2\Delta/\Delta_{\max}) < \beta \Delta_{\max}$.

2.5 Apogee Detection

The *apogee* is defined as the point in convergence space that maximizes the product of confidence and convergence strength:

Definition 2.7 (Apogee). *Let $\mathcal{P}$ be the Pareto frontier of non-dominated solutions in the (confidence, convergence) space. The apogee A^* is:*

$$A^* =_{(c,k) \in \mathcal{P}} [c \cdot k] \quad (10)$$

where c denotes confidence and k denotes convergence strength.

The apogee represents the optimal operating point: it balances high individual agent confidence with strong multi-agent agreement. When both are high, we have strong evidence of truth. When confidence is high but convergence is low, agents may be individually confident but disagreeing—a warning sign that merits further investigation. When convergence is high but confidence is low, agents agree but remain uncertain—indicating insufficient information.

3 System Architecture

3.1 Overview

The GCV system processes a claim through a pipeline of independent verification agents, computes convergence metrics, detects the apogee, and extracts the genomic code. Figure 1 illustrates the system architecture.

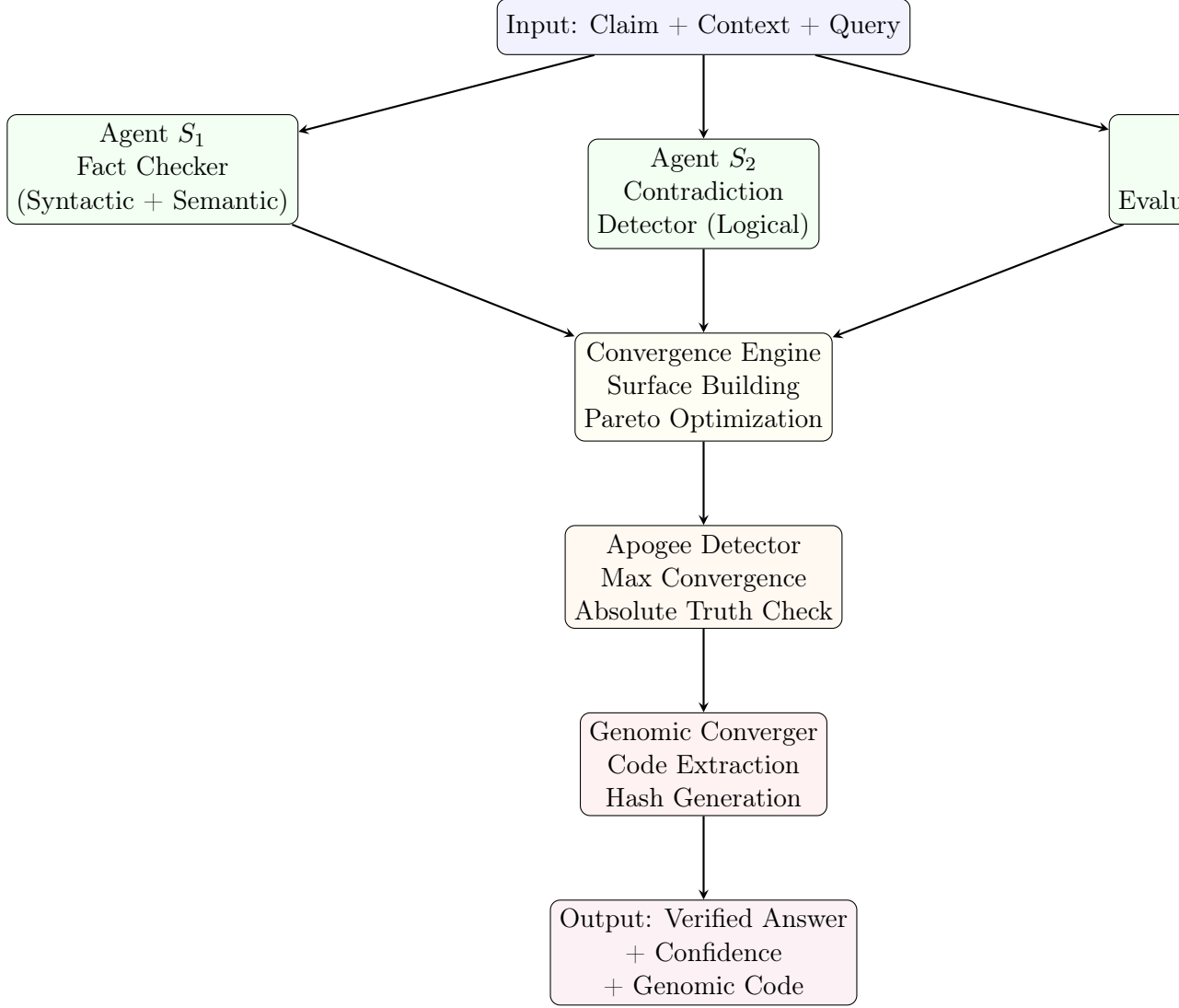


Figure 1: GCV System Architecture. Three independent verification agents process the claim through different methodologies. Their results converge through the convergence engine, the apogee detector finds the optimal point, and the genomic converger extracts the truth signature.

3.2 Agent S_1 : Fact Checker

The Fact Checker agent combines two complementary verification approaches:

Syntactic Scoring. Weighted Jaccard similarity with entity weighting:

$$S_{\text{syn}}(c, k) = \frac{\sum_{w \in c \cap k} \omega(w)}{\sum_{w \in c \cup k} \omega(w)} \quad (11)$$

where $\omega(w) = 2$ if $|w| > 5$ (potential named entities), and $\omega(w) = 1$ otherwise. This gives higher importance to content words and entities over function words.

Semantic Scoring. Cosine similarity of sentence embeddings using the Sentence-BERT model [13]:

$$S_{\text{sem}}(c, k) = \frac{\mathbf{e}(c) \cdot \mathbf{e}(k)}{\|\mathbf{e}(c)\| \|\mathbf{e}(k)\|} \quad (12)$$

3.3 Agent S_2 : Contradiction Detector

The Contradiction Detector identifies logical inconsistencies between the generated answer and the source context at three levels:

Level 1 — Entity Negation. Scan for entity pairs where one appears with a negation modifier in the context but without negation in the answer (or vice versa).

Level 2 — Contradiction Markers. The combined text is scanned for discourse markers indicative of contradiction:

Category	Markers
Contrast	<i>but, however, although, though, yet, nevertheless</i>
Opposition	<i>on the contrary, conversely, instead, rather</i>
Correction	<i>actually, in fact, as a matter of fact</i>

Level 3 — Structural Analysis. Sentence pairs are analyzed for structural patterns indicating logical inconsistency.

The confidence score is:

$$C_2 = \max(0, 1 - 0.25 \cdot n_{\text{contradictions}}) \quad (13)$$

3.4 Agent S_3 : Completeness Evaluator

The Completeness Evaluator extracts key aspects from the original user query and checks their coverage in the generated answer:

Algorithm 1 Completeness Evaluation

```
1: Input: query  $q$ , answer  $a$ 
2: Output: completeness score  $C_3$ 
3:  $\mathcal{A} \leftarrow \text{ExtractAspects}(q)$   $\triangleright$  Tokenize, filter stopwords, keep  $|w| > 4$ 
4: if  $|\mathcal{A}| = 0$  then
5:   return 0.5
6: end if
7:  $\mathcal{C} \leftarrow \{a \in \mathcal{A} : a.\text{lower}() \in a.\text{lower}()\}$ 
8:  $C_3 \leftarrow |\mathcal{C}|/|\mathcal{A}|$ 
9: return  $C_3$ 
```

3.5 Convergence Computation

The convergence engine computes two key metrics:

Convergence Score.

$$K = \bar{C} \cdot (1 - \sigma_C) \quad (14)$$

where $\bar{C} = \frac{1}{n} \sum_i C_i$ is the mean confidence across agents, and σ_C is the standard deviation. High convergence requires both high mean confidence and low variance.

Surface Value. For Pareto frontier construction:

$$F(C_1, \dots, C_n) = \bar{C} \cdot \left(1 - \frac{1}{n^2} \sum_{i,j} |C_i - C_j| \right) \quad (15)$$

3.6 Genomic Code Extraction

The genomic code is a deterministic SHA-256 hash of the convergence pattern:

$$H = \text{SHA-256} \left(\bigoplus_{i=1}^n \text{Agent}_i : C_i \right) \quad (16)$$

where $\oplus$ denotes string concatenation of agent identifiers and their confidence scores, sorted by confidence in descending order. When all agents exhibit confidence exceeding 0.95, the statement is classified as *absolute truth* within the system.

4 Implementation

4.1 Core Components

The system is implemented in Python. Table 1 lists the main classes and their roles.

Table 1: Core system components

Class	Purpose
VerificationAgent	Abstract base class for all agents
FactCheckerAgent	Agent S_1 : semantic-syntactic verification
ContradictionDetectorAgent	Agent S_2 : logical contradiction detection
CompletenessEvaluatorAgent	Agent S_3 : query completeness evaluation
ConvergenceEngine	Multi-agent agreement computation
ApogeeDetector	Optimal convergence point finding
GenomicConverger	Truth pattern extraction and hashing

4.2 Self-Correcting RAG Integration

The GCV system integrates into a full RAG pipeline:

Algorithm 2 Self-Correcting RAG with GCV

```

1: Input: question  $q$ , max rounds  $M = 3$ 
2: Output: verified answer, history
3: context  $\leftarrow$  Retrieve( $q$ )
4: for  $r \in 1 \dots M$  do
5:   answer  $\leftarrow$  Generate( $q$ , context)
6:   result  $\leftarrow$  GCV.verify(answer, context,  $q$ )
7:   history  $\leftarrow$  history  $\cup$  result
8:   if result.passed then
9:     return result.verified_answer, history
10:  end if
11:  context  $\leftarrow$  ImproveContext(context, result.suggestions)
12: end for
13: return BestAnswer(history)

```

5 Theoretical Properties

5.1 Soundness

Proposition 5.1 (Soundness). *If all agents S_i are individually sound (they only mark statements as “verified” when they are indeed supported by the context), then the genomic code G^* contains only true statements.*

Proof. By Definition 2.2, $G^* = \bigcap_{i=1}^n T_i$. If each T_i contains only statements

that are true relative to the context, then their intersection also contains only true statements. Soundness is preserved under intersection. $\square$

5.2 Completeness Relative to the Ensemble

Proposition 5.2 (Relative Completeness). *Any statement verifiable by at least one agent can be identified as “potentially true,” even if it does not appear in the genomic code G^* .*

This follows directly from the architecture: each agent reports its individual confidence, and statements with partial support (high confidence from some agents, low from others) are flagged for human review or additional context retrieval.

5.3 Convergence Rate Analysis

The rate at which undecidability decreases depends on three factors: the number of agents n , the individual undecidability rate p , and the inter-agent correlation ρ . For correlated agents:

$$\mathbb{P} \left(x \in \bigcap_{i=1}^n U_i \right) \leq p^n + \frac{\rho}{1-\rho} (p - p^n) \quad (17)$$

5.4 The Genomic Code as a Basis for Certainty

The genomic code G^* serves as a *basis of certainty*—a foundation of statements verified across all independent perspectives. This concept parallels ideas in epistemology and philosophy of science:

- **Consensus theory of truth:** Habermas (1973) and Longino (1990) argue that truth emerges from uncoerced consensus among diverse perspectives.
- **Triangulation:** Davidson (1991) proposes that knowledge requires triangulation between self, other, and world.
- **Ensemble methods in machine learning:** Breiman (2001) demonstrated that aggregating multiple models reduces variance and improves generalization.

The genomic code operationalizes these philosophical principles in a computational framework.

5.5 The Apogee as Optimal Operating Point

The apogee represents the Pareto-optimal trade-off between two competing objectives:

- **High individual confidence:** Each agent is highly certain in its assessment.
- **High inter-agent convergence:** Agents strongly agree with each other.

Different applications may prioritize different regions of the convergence space. Safety-critical applications may require operating near the apogee, accepting lower throughput for higher certainty. Real-time applications may accept lower convergence for faster responses.

5.6 Limitations and Future Work

1. **Independent Assessment:** True independence between agents is an idealization. In practice, agents trained on similar data or using related methodologies will exhibit correlation ($\rho > 0$), reducing effective ensemble size. Future work should quantify and minimize inter-agent correlation.
2. **Agent Quality:** The genomic code is only as reliable as the constituent agents. Systematic errors shared across agents (e.g., all agents trained on the same biased data) will propagate to G^* .
3. **Computational Cost:** Running multiple agents increases latency. However, agents can be parallelized, and lightweight agents can provide rapid initial screening before invoking more expensive verification.
4. **Natural Language Ambiguity:** Some statements are genuinely ambiguous, and no amount of verification can resolve inherent linguistic uncertainty. The system should report uncertainty rather than forcing binary decisions.
5. **Empirical Validation:** This paper presents the theoretical framework and implementation. Rigorous empirical evaluation on standardized benchmarks (HaluEval [14], FactCC [5], SummaC [6]) is necessary to quantify real-world performance.

6 Related Work

6.1 Fact Verification

Automated fact verification has been extensively studied. Thorne et al. [11] introduced the FEVER dataset and established baseline models for evidence-

based fact checking. Guo et al. [12] provide a comprehensive survey covering methods from text classification to graph neural networks. The key distinction of our work is the multi-system convergence framework, which is orthogonal to the specific verification methodology used by each agent.

6.2 Hallucination Detection in Generation

Kryściński et al. [5] proposed FactCC, a weakly-supervised model for evaluating factual consistency in abstractive summarization. Laban et al. [6] improved upon this with SummaC, which employs a multi-task approach across NLI and summarization-specific features. Zha et al. [7] introduced AlignScore, unifying factual consistency evaluation through alignment functions.

Manakul et al. [8] proposed SelfCheckGPT, which detects hallucinations by comparing multiple LLM samples without requiring external knowledge bases. Our approach differs fundamentally: rather than relying on a single method’s internal consistency, we leverage convergence across methodologically independent systems.

6.3 Multi-Agent Verification

Recent work explores multi-agent approaches for improving factual accuracy. Cohen et al. [9] proposed cross-examination between language models to detect factual errors. Du et al. [10] demonstrated that multi-agent debate improves reasoning and factuality. Our work provides the theoretical foundation for why such multi-agent approaches work: they increase the number of independent verification perspectives, asymptotically eliminating undecidable statements as predicted by Theorem 2.3.

6.4 Gödel and Artificial Intelligence

The implications of Gödel’s theorems for AI have been debated since Lucas (1961) and Penrose (1989). More recently, research has focused on the fundamental limits of machine reasoning. To our knowledge, this is the first work to propose a practical engineering solution to Gödel’s limitation specifically in the context of automated fact verification.

7 Conclusion

We have presented Genomic Convergence Verification (GCV), a theoretically grounded framework for hallucination detection in Retrieval-Augmented Generation systems. The framework addresses Gödel’s limitation through a novel insight: while any single verification system is bounded by incompleteness,

the intersection of multiple independent systems asymptotically eliminates undecidable statements.

Our implementation provides three specialized verification agents employing complementary methodologies (syntactic, semantic, logical, and query-aware analysis), a differential convergence equation that models confidence evolution, an apogee detection algorithm for identifying optimal convergence points, and a genomic code extraction mechanism that produces a unique, reproducible signature of truth.

The framework is modular, extensible, and available as open-source software. It contributes to the broader goal of building trustworthy AI systems where verification is not bounded by the mathematical limitations of any single approach.

Acknowledgments

We thank [acknowledgments to be added].

References

- [1] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173-198, 1931.
- [2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 9459-9474, 2020.
- [3] J. Maynez, S. Narayan, B. Bohnet, and R. McDonald. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1906-1919, 2020.
- [4] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784-3803, 2021.
- [5] W. Kryściński, B. McCann, C. Xiong, and R. Socher. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332-9346, 2020.
- [6] P. Laban, T. Schnabel, P. N. Bennett, and M. A. Hearst. SummaC: Revisiting NLI-based models for inconsistency detection in summarization.

Transactions of the Association for Computational Linguistics, 10:163-177, 2022.

- [7] Y. Zha, Y. Yang, R. Li, and Z. Hu. AlignScore: Evaluating factual consistency with a unified alignment function. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.
- [8] P. Manakul, A. Liusie, and M. J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- [9] R. Cohen, M. Hamri, M. Geva, and A. Globerson. LM vs LM: Detecting factual errors via cross examination. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- [10] Y. Du, S. Li, A. Torralba, J. Tenenbaum, and I. Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.
- [11] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. FEVER: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 809-819, 2018.
- [12] Z. Guo, M. Schlichtkrull, and A. Vlachos. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178-206, 2022.
- [13] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, pages 3982-3992, 2019.
- [14] J. Li, X. Cheng, W. X. Zhao, J. Y. Nie, and J. R. Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.
- [15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.