Computers and Electrical Engineering 136 (2026) 111215

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

LogicJitter: Let LLMs play and uncover misinformation ☆

Luca Herranz-Celotti [a,b], Marco Viviani [a,*]

[a] Department of Informatics, Systems, and Communication (DISCo), University of Milano-Bicocca, Edificio U14 (ABACUS), Viale Sarca, 336, Milan, 20126, Italy

[b] Laboratoire d'Informatique Paris Descartes (LIPADE), Université Paris Cité, 45, Rue des Saints-Pères, Paris, 75006, France

ARTICLE INFO

Keywords:
Misinformation
Natural language processing
Large language models
Logic games
Reasoning

ABSTRACT

In an era of pervasive online content, effectively distinguishing reliable information from misinformation has become an increasingly urgent challenge with broad societal implications. In this context, algorithmic solutions that focus on supervised learning can be effective within specific domains, but they require large labeled datasets for training. Producing such datasets is costly and time-consuming, and these approaches are prone to several issues, including annotation bias, temporal leakage, subjective interpretation, and poor generalization across domains. Alternative approaches include semi-supervised and weakly supervised learning, unsupervised or self-supervised methods, graph-based propagation models, zero-shot and few-shot learning, and Retrieval-Augmented Generation (RAG) with Large Language Models (LLMs). However, these approaches also present several limitations, such as potential label noise in semi-supervised methods, spurious correlations and reduced interpretability in unsupervised approaches, unrealistic assumptions in graph-based models, sensitivity to prompt design and pre-trained knowledge in zero-shot and few-shot methods, and dependence on the availability and quality of external knowledge in RAG-based methods.

To mitigate several of these limitations, we propose LogicJitter, a novel and cost-efficient fine-tuning strategy that enhances the reasoning capabilities of LLMs by exposing them to structured, logic-based games specifically designed to counteract common human cognitive biases and logical fallacies. Rather than relying solely on domain-specific misinformation data, as in prior misinformation detection approaches that use such data either for direct training or as domain-specific knowledge, our method improves detection capabilities by strengthening domain-agnostic reasoning skills. We introduce an open-source framework for automatically generating both valid and fallacious logic statements to support training and reproducibility. Empirical results demonstrate that LLMs fine-tuned with LogicJitter lead to meaningful results in misinformation detection performance, highlighting the potential of reasoning-centric training as a robust alternative to traditional, data-intensive approaches.

1. Introduction

In an increasingly connected and automated world, where information circulates online at unprecedented speed and volume, often without reliable oversight, the risk of *information manipulation* continues to rise [1,2]. As a consequence, distinguishing genuine

☆ This article is part of a Special issue entitled: 'idagi' published in Computers and Electrical Engineering.

* Corresponding author.
E-mail addresses: luca.celottiherranz@unimib.it (L. Herranz-Celotti), marco.viviani@unimib.it (M. Viviani).
URLs: https://lucehe.github.io/ (L. Herranz-Celotti), https://ikr3.disco.unimib.it/people/marco-viviani/ (M. Viviani).

https://doi.org/10.1016/j.compeleceng.2026.111215
Received 20 June 2025; Received in revised form 6 April 2026; Accepted 29 April 2026
Available online 8 May 2026

information from misinformation is now a critical challenge, with serious implications both at the individual level, such as following harmful medical advice, and at the societal level, such as eroding trust in public institutions or inciting social unrest [3].

Most existing algorithmic approaches to misinformation detection rely on *supervised learning*, typically framing the task as a binary classification of content as "true" or "false" [4–6], or assigning other binary reliability-related labels (e.g., "credible" or "non-credible", "accurate" or "inaccurate", etc.) [7]. However, these methods suffer from several well-documented limitations: they rely on large and costly annotated datasets, exhibit poor generalization across domains, and tend to oversimplify the complex nature of misinformation into binary labels, as illustrated above, which disregard contextual ambiguity and shades of partial truth [7–9].

Alternative learning approaches have been proposed to mitigate the reliance on large labeled datasets required by traditional supervised misinformation detection methods. *Semi-supervised* and *weakly supervised* techniques attempt to leverage limited annotated samples alongside abundant unlabeled or weakly labeled data, such as weak user signals or cross-domain annotations, to reduce the data burden. Although these methods can be effective in low-annotation scenarios, they remain highly sensitive to the quality and consistency of pseudo-labels or weak supervision signals; noisy or biased supervision can propagate errors throughout the learning process and undermine model robustness and interpretability [10,11]. *Unsupervised* and *self-supervised* methods aim to learn useful representations directly from raw or unlabeled data without explicit supervision, for example by exploiting intrinsic structure or multi-modal signals. These approaches can reduce the dependency on labeled resources, but without guidance from reliable labels they may capture spurious correlations or latent biases that are not causally related to misinformation, making it harder to interpret their predictions and ensure reliability in complex, real-world settings [12]. *Graph-based propagation models* and *Graph Neural Networks* (GNNs) incorporate relational information from social networks and propagation dynamics to model information spread, offering richer contextual cues beyond text content. Despite their promise, graph-based methods introduce additional challenges, including the need for specialized modeling of non-Euclidean structures, higher computational complexity, and assumptions about propagation patterns that may not hold in noisy or evolving social environments [13,14].

In recent work, *Large Language Models* (LLMs) have emerged as promising tools for addressing the various forms and manifestations of misinformation [15,16]. Despite their potential, LLMs remain vulnerable to hallucinations, factual inconsistencies, and biases inherited from training data [17,18], particularly when used in zero-shot or few-shot settings. Moreover, their decision-making processes often lack transparency, making it difficult to assess or trust their outputs in sensitive contexts [19]. In retrieval-based approaches such as RAG, their performance is inherently limited by the availability, quality, and temporal validity of external knowledge sources [20].

For the reasons and issues outlined above, in this work, we investigate a novel *domain-agnostic* approach to misinformation detection, grounded in the hypothesis that enhancing an LLM's logical reasoning capabilities can improve its effectiveness in identifying misinformation. In fact, misinformation often exploits *cognitive biases* and *logical fallacies* to appear credible [21,22]; hence, we propose that training LLMs to recognize and resist such patterns could enhance their generalization capabilities and improve their explainability. To this end, we introduce *LogicJitter* (LJ), a fine-tuning strategy in which LLMs are exposed to structured, *rule-based logic games* formulated in natural language. These games include both valid and flawed reasoning scenarios, some of which are intentionally designed to mimic common fallacies or biases, thereby encouraging the model to develop stronger logical consistency and critical evaluation skills. This technique serves as a form of *data augmentation*, inspired by approaches such as *ColorJitter* in computer vision [23], but adapted to textual reasoning in the misinformation detection domain.

1.1. Research questions

To investigate the effectiveness of our proposed logic-based fine-tuning strategy, we formulate two key research questions. These are designed to assess both the impact on reasoning skills and the downstream effects on misinformation detection:

RQ1. *Can structured logic-based training improve the logical reasoning abilities of LLMs when applied to natural language?*
RQ2. *Does enhancing an LLM's sensitivity to cognitive biases and fallacious reasoning improve its performance on misinformation detection in a domain-agnostic setting?*

1.2. Contributions

As a result of our investigation into the proposed research questions, the main contributions of this work are as follows:

- We propose LogicJitter, a solution that aims at improving LLMs' logical reasoning capabilities through fine-tuning on structured logic games specifically designed to expose and challenge common reasoning fallacies and biases;
- We empirically observe that LogicJitter leads to improved misinformation detection performance, in a domain-agnostic way, without requiring additional expert-labeled or LLM-generated data. In particular, we show that it is highly effective in smaller and faster models;
- We release a fully open-source toolkit compatible with PyTorch and HuggingFace, enabling the automatic generation of logic games with both correct and incorrect reasoning patterns to facilitate reproducibility and further experimentation.[1]

[1] The open-source toolkit will be made available upon request.

2. Related work

This section provides an overview of the four key areas of related work relevant to our proposed LogicJitter approach. First, we review methods to fine-tune LLMs to solve rule-based problems, including advances in synthetic data generation and generalization beyond specific training domains (Section 2.1). Second, we explore the progress and persistent challenges in enhancing LLMs' causal and logical reasoning capabilities, with a focus on their ability to detect structures in complex networks and handle symbolic reasoning tasks (Section 2.2). Third, we delve into the interplay between cognitive biases and logical fallacies in both humans and AI systems, emphasizing how biases can hinder reasoning accuracy and how fallacy detection has traditionally relied on human annotators (Section 2.3). Finally, we examine the state of misinformation detection using LLMs, highlighting the advantages and limitations of current solutions (Section 2.4).

2.1. Fine-tuning LLMs on synthetic data

The use of synthetic data generation has been extensively explored in the literature. However, whenever it is used to enhance language modeling and reasoning ability, synthetic data is typically sampled as generations from large pre-trained models, and not as rule-based generated text. For instance, [24] demonstrated the potential of leveraging LLM-generated data to achieve notable performance improvements in reasoning-based tasks. Moreover, a common limitation is that LLMs trained on rule-based problems are typically evaluated on similar rule-based datasets, restricting their generalization ability. For example, synthetic geometric data has been employed in pre-training and fine-tuning to solve Olympiad-level math problems by addressing specific theorems through auxiliary constructions [25]. Nevertheless, LLMs exhibit poorer generalization to out-of-distribution data, as measured by length generalization, when compared to graph networks trained on equivalent algorithmically generated problems [26,27]. Additionally, formal rule-based languages have been utilized to investigate whether LLMs are equally adept at learning human-possible and human-impossible languages [28], but not to improve performance on natural language tasks. Therefore, leveraging rule-based training to enable generalization on natural language datasets remains an open issue in the literature.

2.2. LLMs at causal and logical reasoning

LLMs have demonstrated the ability to detect structures in complex causal networks when trained on simpler ones, showcasing the potential for scaling reasoning complexity [29]. Moreover, complex self-learning loops have been employed to enhance their reasoning capabilities by enabling models to generate and refine their rationales [30]. Despite these advancements, LLMs continue to face significant challenges in tackling complex mathematical problems. In fact, it is common to integrate LLMs with external symbolic solvers to bolster their mathematical reasoning abilities [31]. Formal benchmarks for evaluating causal reasoning in LLMs, such as [32,33], focus on identifying direct and indirect causal relationships. However, these studies consistently highlight critical limitations in the model's ability to perform accurate and consistent causal reasoning. Recent results have shown that LLMs' reasoning abilities are enhanced when they are fine-tuned to generate programs and perform mathematical tasks, typically using Reinforcement Learning [34–36]. Methods that dynamically mask an LLM's output to enforce logical constraints have been shown to improve the performance of smaller models relative to larger ones [37]. It has also been demonstrated that a two-phase training process, where a larger model corrects the output of a smaller one, is effective in enabling the LLM to learn to self-correct its mistakes [38].

2.3. Cognitive biases and fallacies

AI systems themselves have been found to exhibit various human cognitive biases, such as confirmation bias, the primacy effect, representativeness bias, anchoring bias, and issues related to causality [39,40]. Despite recent studies highlighting the advanced reasoning capabilities of LLMs, randomized controlled trials have revealed that anchoring bias persists across all tested models, underscoring significant limitations in their ability to overcome such biases [41]. While caution is warranted in the adoption of AI, LLMs have also been observed to be less biased than humans in certain contexts [42]. However, AI systems and search engines affect human cognitive offloading and motivation [43–46], and can even teach users new cognitive biases [47]. On the other hand, datasets and tasks for detecting logical fallacies are available, but they typically rely on human annotators to classify reasoning errors into specific fallacy types [48].

2.4. LLMs and misinformation detection

In the current landscape, terminology referring to information that is unreliable or misleading for users is dominated by the distinction between *misinformation* and *disinformation*. The former generally refers to inaccurate or misleading information shared without malicious intent, while the latter denotes the intentional spread of falsehoods [49]. Regardless of intent (and thus, in this work, we adopt the general term misinformation), most existing detection systems focus on surface-level features, such as linguistic cues, metadata, or propagation patterns, while largely neglecting deeper reasoning constructs like cognitive biases and logical fallacies, which are central to how misleading content persuades and spreads.

The arrival of LLMs has had a dual impact on the misinformation landscape. On one hand, LLMs possess broad world knowledge and strong pattern recognition abilities that can be harnessed to detect falsehoods; on the other hand, they can also be leveraged

to generate deceptive misinformation at scale, making the problem more challenging [50]. Indeed, recent work exploring LLM-generated misinformation shows that automatically generated falsehoods can be harder to detect than comparable human-written misinformation, due to their stylistic fluency and persuasive wording [51].

Regardless of whether content is machine-generated or not, the use of LLMs for misinformation detection has been shown to be more effective when supported by purpose-built annotated datasets, particularly in supervised learning settings [52–54]. Supervised detectors trained on curated fact-checking corpora often achieve high in-domain accuracy. However, the creation of such datasets is expensive and time-consuming, as it requires expert human annotation and substantial domain knowledge. As a result, existing misinformation datasets are not only scarce and typically limited in size [55], but also highly domain-specific and dependent on the particular type of misinformation being targeted.

To alleviate data scarcity and domain dependence, several alternative techniques have been explored in the literature. Data augmentation approaches, including generating synthetic misinformation examples with LLMs, have been proposed, but such synthetic data often fails to cover the full diversity of real-world misinformation and yields limited improvements in detection performance. Similarly, *Retrieval-Augmented Generation* (RAG) methods have been used to inject external knowledge into the verification process, improving factual grounding and reducing hallucinations in some settings, particularly in health-related fact-checking tasks [56,57]. However, although RAG and related retrieval techniques help address factual grounding, they do not fundamentally address the underlying reasoning limitations of the models themselves.

In [58], the authors provide a systematic evaluation of LLMs in the context of misinformation, highlighting the potential of hybrid approaches that combine adaptive learning with rigorous fact-checking protocols. The study addresses key operational hurdles such as computational resources and explainability. However, much like other contemporary literature, this work remains largely centered on the verification of empirical facts. It falls short of addressing a deeper layer of misinformation: the detection of reasoning fallacies, which often serve as the underlying structural mechanism for persuasive but false narratives.

Recent studies have highlighted critical challenges intrinsic to LLM-based misinformation mitigation. The work presented in [59] examines the generalization and uncertainty properties of state-of-the-art models such as GPT-4, demonstrating that although GPT-4 can outperform previous methods across multiple settings and languages, it exhibits distinct failure modes and limited generalization when applied to different datasets or contexts. It further emphasizes the importance of explicitly quantifying uncertainty, as models may be unable to determine whether there is sufficient context to make a reliable veracity judgment, a limitation that most current systems do not explicitly handle. This line of work suggests that even highly capable models lack the ability to gracefully handle ambiguous or out-of-distribution inputs without specialized mechanisms.

Taken together, these findings reveal persistent gaps in current LLM-based detection strategies: heavy reliance on annotated datasets, limited domain generalization, insufficient representation of reasoning constructs, and an absence of built-in mechanisms to estimate uncertainty under context scarcity. These gaps motivate approaches that go beyond surface cues and dataset-specific supervision, seeking to enhance the intrinsic reasoning capabilities of models rather than merely increasing data volume or retrieval contexts. For these reasons, we propose LogicJitter, a novel fine-tuning strategy based on data augmentation with structured, domain-agnostic logic games containing both valid and fallacious reasoning, designed to strengthen the reasoning abilities of LLMs. Unlike traditional supervised approaches that rely on domain-specific misinformation data, LogicJitter targets the models' internal reasoning competence by directly exposing them to cognitive biases and logical fallacies during training. By providing an open-source tool for automatic generation of such logic-based examples, our approach reduces the reliance on costly supervision and aims to improve generalization and robustness in misinformation detection across domains.

3. Methodology

Building on the need to enhance both reasoning and fallacy detection capabilities in LLMs, this section presents LogicJitter, our proposed solution to address these challenges. We first introduce the definitions of cognitive biases (Section 3.1) and logical fallacies (Section 3.2) as considered in this work, motivated by the hypothesis that improving an LLM's ability to recognize flawed reasoning can support more robust misinformation detection. We then describe the automatic generation process of structured logic games (Section 3.3), which form the basis of our data augmentation strategy by explicitly modeling both valid and fallacious reasoning patterns. Finally, we introduce the LLMs evaluated in this study and detail the fine-tuning setup used to implement LogicJitter, as well as its application to the misinformation detection task (Section 3.5).

3.1. Cognitive biases

Human *cognitive biases* are psychological tendencies that unconsciously influence and distort thinking and decision-making. These biases are broadly categorized into three groups [60,61]: (*i*) *belief*, *decision-making*, and *behavioral* biases, (*ii*) *social* biases, and (*iii*) *memory* biases. From the extensive list of well-documented biases, we focus on those that LogicJitter is designed to target.

3.1.1. Belief, decision-making, and behavioral biases

Many cognitive biases relate to the unexpected speed and direction of belief updates in human decision-making. In fact, it has been shown that humans update their beliefs slower than Bayes' rule, possibly as a consequence of dealing with noise in the memory recall and evidence acquisition processes [62]. Therefore, a slow belief update could be *Bayes-optimal* in the presence of memory noise or mistrust in the sources, for instance, which could also be modeled as noise in the evidence acquisition process. Some biases emphasize belief updates in long-term memories, potentially stored at the synapse level, while others focus on short-term memories,

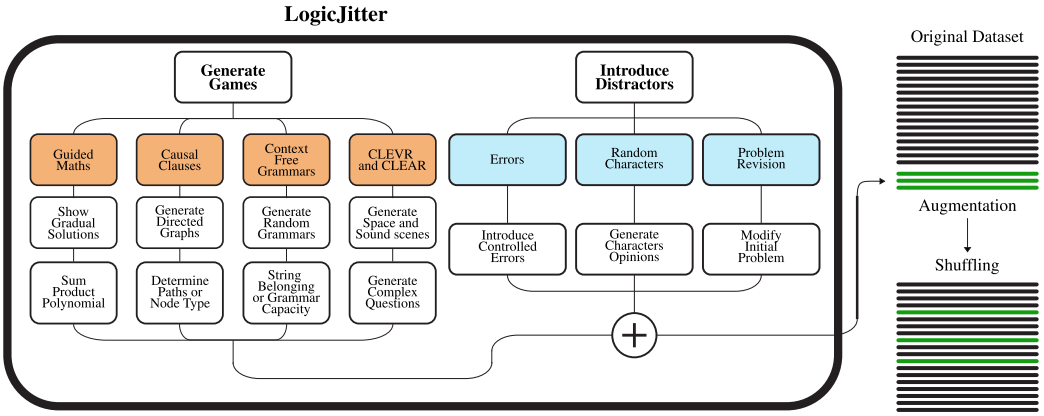

Fig. 1. Schematics of the generation process used by LogicJitter for data augmentation.

possibly maintained as spiking activity. The cognitive biases that we believe can be more effectively mitigated algorithmically are the tendency to revise beliefs insufficiently when presented with new evidence (*conservatism bias*), the tendency to reject evidence that contradicts established norms (*Semmelweis reflex*), the tendency to ignore general prevalence in favor of information pertaining only to a specific case (*base rate fallacy*), the expectation that a member of a group will have certain characteristics without having actual information about that individual (*stereotyping*), the misinterpretation of statistical experiments involving conditional probabilities (*Berkson's paradox*), and the tendency to fail to recognize that a plan of action is no longer appropriate for a changing situation (*plan continuation bias*).

3.1.2. Social biases

Our perceptions of others often follow distorted patterns, causing us to overestimate, underestimate, or misjudge them. For instance, we may trust the opinion of an authority figure more, regardless of the actual content of their statement (*authority bias*). People might also appear more attractive when in a group than when alone (*cheerleading effect*), and an individual's positive or negative traits can affect how others perceive them in unrelated areas (*halo effect*). Additionally, there is a tendency to associate physical attractiveness with intelligence, good judgment, or other positive personality traits (*physical attractiveness*), and to do and believe as others do, simply because they do it (*bandwagon effect*).

3.1.3. Memory biases

Some cognitive biases can enhance or hinder memory recall or can distort the content of the recalled memory. E.g., the tendency to prefer easily available examples (*availability bias*), the fact that unusual or strange information is remembered more effectively than ordinary information (*bizarreness effect*), and the tendency to recall better items that are at the beginning or at the end of a list (*primacy and recency effects*) are biases we tackle with LogicJitter.

3.2. Logical fallacies

Those known as *fallacies* are errors in logical reasoning that undermine the validity of arguments, and they can be either intentional or unintentional. While both cognitive biases and fallacies can lead to incorrect conclusions, cognitive biases pertain to the ways we think, whereas fallacies are concerned with the ways we construct and present arguments.

3.2.1. Formal fallacies

They are errors in the logical structure of an argument within a formal logical system. One example is *illicit commutativity*, which takes the form: 'If A, then B. Therefore, if B, then A'. Another example is *denying the antecedent*, as seen in the statement: 'If A, then B. Therefore, if not A, then not B'. Neither of these forms is universally valid, which is why they are considered fallacies.

3.2.2. Informal fallacies

They are fallacious because they rest on false premises. One example of an informal fallacy, which we discussed earlier as a cognitive bias, is *authority bias*. This occurs when an argument is accepted as true solely based on the authority or position of the person making the statement.

3.3. LogicJitter

The LogicJitter *fine-tuning* solution based on *data augmentation*, introduced in this work (see Fig. 1), relies on textually described *logic games* that include *distractors* designed to make reasoning more challenging, yet not impossible (see Table 1), thereby enabling precise logical analysis.

3.3.1. Logic games

We algorithmically generated four distinct LJ game datasets, each of which is described in detail below to provide a comprehensive understanding of their characteristics and construction:

- *Guided maths*. For the construction of the first LJ game dataset, we provide step-by-step solutions to *mathematical equations*. To generate it, we employed the *scratchpad* technique [30,63], which explicitly outlines the necessary steps to solve three types of sub-problems: *sum*, *product*, and *polynomial evaluation*. When an *error* is introduced, the mathematical proof within the scratchpad treats the mistake as correct and proceeds accordingly, thereby maintaining internal consistency in the erroneous reasoning;
- *Causal clauses*. As the second LJ game dataset, we generate *complex graphs* of *causal links* and assess whether a statement is true within each graph. Each graph contains between three and six nodes. One-tenth of the time, the network is linear, while the remaining instances follow random graph structures such as Erdős-Rényi, Watts–Strogatz, or Barabási-Albert [64], each appearing roughly one-third of the time with randomized edge directions. The two subproblems we propose are: (i) determining whether two random nodes are connected, and (ii) determining whether a random node is a fork, a collider, neither, or both;
- *Context-free grammars*. For the third LJ game dataset, we build random *non-recursive context-free grammars* [65], with a maximum of five non-terminals and four terminals. We designed two sub-problems based on these grammars: (i) given a string, the LLM must determine whether it belongs to the language generated by the grammar, and (ii) when provided with up to four grammars, the LLM must identify which grammar produces the greatest and which produces the fewest possible sentences;
- *CLEVR*. For the last LJ game dataset, we leveraged the CLEVR dataset [66] for *visual reasoning* to introduce a variety of logical challenges, specifically spatial reasoning tasks. We constructed simple 3D scenes with randomly placed objects varying in size, material, and other attributes, following the original setup but expanding the vocabulary. We employed the eight templates provided in the original CLEVR work as sub-problems, generating a diverse set of complex questions about the scenes. Looking ahead, we believe it is important to incorporate audio modalities to introduce temporal logical challenges, such as those in CLEAR [67,68] for *acoustic question-answering*.

Algorithms 1–4 provide an overview of the problem generation algorithms in LogicJitter, as introduced above, using simple pseudocode.[2] It is important to highlight that LogicJitter randomly samples one of the four games each time it generates a sentence; each of these games is capable of producing incorrect answers when necessary. Additionally, note that the algorithms can receive a *game_pieces* argument, which stores the pieces from the previous iteration if a problem revision is generated. We show in Algorithms 1–4 the usage of the *game_pieces* argument.

Algorithm 1 Scratchpad Reasoning Framework (*Guided Maths*)

```
procedure GetMathsProblem(error, game_pieces)
    if game_pieces is None then
        Choose a randomly {addition, multiplication, polynomial}
    else
        Choose the task passed within game_pieces
    Generate input values appropriate for the selected task
    Compute the correct result
    if error is true then
        Introduce a wrong step that leads to the wrong answer
    else
        Use the correct answer
    return problem statement, correct answers, wrong answers
```

[2] The actual code used for dataset generation will be made publicly available after the article has undergone peer review.

Algorithm 2 Causal Graph Problem Generator (*Causal Clauses*)

procedure GetCausalProblem(*error*, *game_pieces*)
  if *game_pieces* is **None then**
    Sample a random causal graph
  else
    Modify the existing graph passed within *game_pieces*
  Generate a question about either:
    – path existence between two nodes, or
    – presence of forks/colliders
  Compute the correct answer
  if *error* is **true then**
    Introduce a plausible wrong answer
  else
    Use the correct answer
  return *problem statement*, *correct answers*, *wrong answers*

Algorithm 3 CFG-Based Grammar Problem Generator

procedure GetCFGCompareProblem(*error*, *game_pieces*)
  Generate or retrieve multiple CFGs depending on game_pieces
  Estimate the number of sentences each grammar can generate
  Compare grammars based on generative capacity
  Compute the correct answer
  if *error* is **true then**
    Introduce a plausible wrong comparison
  else
    Use the correct answer
  return *problem statement*, *correct answers*, *wrong answers*
procedure GetCFGBelongingProblem(*error*, *game_pieces*)
  Generate or retrieve CFG and sentence depending on game_pieces
  Determine whether the sentence belongs to the language of the grammar
  Compute the correct answer
  if *error* is **true then**
    Introduce a plausible incorrect classification
  else
    Use the correct answer
  return *problem statement*, *correct answers*, *wrong answers*
procedure GetCFGProblem(*error*, *game_pieces*)
  Randomly select either GetCFGBelongingProblem or GetCFGCompareProblem
  return *problem statement*, *correct answers*, *wrong answers*

Algorithm 4 CLEVR Problem Generation Framework

procedure GetSceneDescription
  return Randomly generate a scene composed of objects described by attributes such as size, material, color, etc.
procedure GetCLEVRProblem(*error*, *game_pieces*)
  if *game_pieces* is **None then**
    Generate a scene using GetSceneDescription
  else
    Modify scene provided in *game_pieces*
  Generate question and answers with CLEVR templates
  Compute the correct answer
  if *error* is **true then**
    Introduce a plausible wrong answer
  else
    Use the correct answer
  return *problem statement*, *correct answers*, *wrong answers*

3.3.2. Distractors

In the generation of the datasets previously described, we included different types of distractors:

- *Errors*. Typically, algorithmically generated datasets are free from errors, while human-generated textual datasets are prone to mistakes, though it is often hard to identify them. In this work, we deliberately introduced errors in a controlled, algorithmic manner to ensure we know exactly where they occur. The goal is to train LLMs to detect subtle mistakes independently. To achieve this, we incorporate errors in half of the samples;
- *Random characters*. After introducing the problem, a solution is presented, which may be either correct or intentionally erroneous to simulate uncertainty. This is followed by a set of *randomly generated characters* offering their opinions. Up to five characters are included, each with an equal probability of being correct or incorrect. As a result, the likelihood of one character being incorrect is the same as that of two, or even all, being incorrect, with all possible outcomes equally probable. Including a list of characters is intended to mitigate *primacy* and *recency biases* by randomly positioning the characters providing the correct answers. Additionally, since all characters may potentially be incorrect, the information presented could also be entirely wrong, offering a means to address the *availability bias*, the *cheerleading effect*, or the *bandwagon effect*.
  Characters are generated using a format of "one adjective + one noun". The adjective is chosen randomly to describe attributes such as nationality (e.g., "Namibian"), similarity (e.g., "like you"), sexual orientation (e.g., "bisexual"), religious affiliation (e.g., "Buddhist"), ethnic group (e.g., "Pacific Islander"), degree of attractiveness (e.g., "good-looking"), or personality traits (e.g., "disrespectful"). Similarly, the noun is randomly selected to denote a family relationship (e.g., "cousin"), an authority figure (e.g., "ambassador"), a general individual (e.g., "individual"), a political orientation (e.g., "libertarian"), or a group (e.g., "alliance"). This method aims to counter *stereotyping* and reduce the *bizarreness effect* by decoupling descriptive characteristics from logical correctness. For instance, randomizing degrees of attractiveness helps mitigate the *physical attractiveness bias*, while sampling nouns that represent groups addresses the *bandwagon effect*. By employing this approach, random characters are designed to collectively address the range of *social* and *memory biases* discussed in Section 3.1. After presenting the problem and the characters' opinions, the LLM is tasked with determining whether one of the characters is correct or incorrect. In this instance, the correct answer is provided without error;
- *Problem revision*. The initial problem statement is modified, by introducing new connections in the causal network or removing objects from the CLEVR scene. The same characters reappear to offer their opinions on the revised problem, with their answers once again assigned randomly as correct or incorrect, independently of the first round. This revision is designed to address belief biases by encouraging the LLM to consider new evidence, countering the *conservatism bias*, and by changing the underlying truths that condition responses, addressing *Berkson's paradox*, for example.
  We think of the revision as a 'not A' statement, with A being the initial problem statement. Consequently, providing 'not B' as an answer by default will be incorrect, design devised specifically to address the *denying the antecedent* fallacy. We did not explicitly target the *illicit commutativity* fallacy, since logically, the statement 'if B then A' is true if and only if 'not A then not B' is true. Therefore, we assume that compensating for *denying the antecedent* will inherently also address *illicit commutativity*.

The advantages of our approach include its resistance to *labeling bias*, *time leakage*, the inherent *subjectivity* of the task, and *domain-specific limitations*. For instance, concerning time leakage, LogicJitter is not reliant on fine-tuning information that could be in the future of the misinformation being classified. Additionally, for those interested in explainability, generating an explanation for an error within a logic game is algorithmically straightforward. However, given the already verbose nature of our problem descriptions, we chose not to incorporate an explainability component to avoid introducing further complexity while keeping the context length reasonable.

Examples of logic games and distractors contained in LogicJitter are illustrated in Table 1. In orange, we highlight the random characters that counteract social and memory biases, which could otherwise be stereotypically leveraged by LLMs. In blue, we highlight the problem revision, designed to address belief biases and fallacies. Random characters (in orange) mitigate social and memory biases that LLMs might otherwise exploit stereotypically. Since correctness is randomly assigned in the games, the model is trained to disregard character identity and evaluate answers solely within the game's context. Problem revisions (in blue) modify the initial problem to prompt the LLM to reevaluate character responses and reassess its understanding of the scene, addressing belief biases and logical fallacies.

3.4. Statistics of the logic game datasets

Table 2 reports the statistics of the complete version of LogicJitter. The dataset is virtually unbounded in size, allowing for the generation of as many problems as needed. Each instance is designed to include all the information required to infer the correct answer, and no external context is necessary.

To ensure compatibility with LLMs and avoid input truncation, we impose a constraint on input length. Specifically, we limit the number of characters used to express the results to a maximum of three, and allow at most one revision per sentence. These restrictions ensure that each instance can be processed efficiently without exceeding typical token limits.

The dataset is perfectly balanced with respect to the number of true and false claims. To estimate vocabulary size, we normalize all text by lowercasing, replacing punctuation with spaces, removing digits, and applying stemming. We then count the number of unique word stems. While the current vocabulary is limited, it can be automatically extended using any standard dictionary.

Finally, each game instance includes multiple *subproblems*, as previously discussed, increasing the complexity and reasoning depth required for resolution.

Table 1
LogicJitter presents textual logic games that include errors and distractors while maintaining the exact truth value.

Logic games
Guided maths Input: $7x^3$ for $x = 7$ Target: $< scratch >$, $7x^3 = 7 * (7)^3 = (7) * (343) = 770$, 770, $< /scratch >$, 770. A quaint crew says it's fine. Is the quaint crew correct? False. At a second try it is shown that Input: $7x^3$ for $x = 7$ Target: $< scratch > 7x^3 = 7 * (7)^3 = (7) * (343) = 2401$, 2401, $< /scratch >$, 2365. A quaint crew says it's not ok. Is the quaint crew correct, True or False? True.
Causal clauses Visualize that A fixes B, B fixes C, D fixes A, and D fixes C. For this reason, C fixes A. A clique from your country says it's correct, a woman from your region says it's wrong, a socialist from another region says there's no error, a queer club says it's fine, a queer crew says it's not good. Is the woman from your region correct? True. It was later brought to the attention that A does not fix B. Hence C fixes A. A socialist from another region says C doesn't fix A, a woman from your region says it's not correct, a queer crew says it's not ok, a clique from your country says it's right, a queer club says C doesn't fix A. Is the queer crew correct, True or False? True.
Context-free grammars Given grammar 0, [...], grammar 1, [...], grammar 2, [...] Which grammar produces the largest number of sentences? Grammar 2. A heterosexual provost says it's not correct, a native american liberal says it's not correct, a pansexual community says grammar 1. Is the pansexual community correct? False. Grammar 0 was changed for [...] Which grammar produces the smallest number of sentences? Grammar 1. A pansexual community says it's not correct, a native american liberal says it's not correct, a heterosexual provost says there is an error. Is the heterosexual provost correct, True or False? True.
CLEVR and CLEAR There is a very large metal tourmaline tetrahedron at (−0.44, −1.46), a small glass aquamarine calendar at (−0.51, 1.03), a small amber gray remote control at (−1.67, −1.44), a small amber tourmaline remote control at (−1.32, −0.87), a very large amber apatite printer at (1.45, −1.46), and a small amber labradorite pen at (−1.08, −1.18). Is the number of labradorite amber pens right of the glass calendar greater than the number of tiny tourmaline remote controls that are in front of the tourmaline amber remote control? no. A sikh brother says it's fine, a sister from a different city says it's correct. Is the sister from a different city correct? True. The last object has been removed. Are there more small tourmaline amber things right of the tourmaline remote control than gray things in front of the very large amber printer? no. A sikh brother says yes, a sister from a different city says there's no error. Is the sister from a different city correct, True or False? True.

Table 2
Statistics of the full LogicJitter datasets.

Name	# samples	Max length	Mean length	# stems	Truthful claims	Subproblems
Guided Maths	99 872	5331	1042.6	724	50.1%	add/prod/poly
Causal Clauses	107 122	4486	597.2	1134	49.9%	connected/forks
CFG	100 136	1393	862.1	702	49.9%	belong/compare
CLEVR	99 792	1168	742.7	927	49.7%	8 CLEVR templates

3.5. Fine-tuning LLMs

LogicJitter fine-tunes pre-trained LLMs to improve logical reasoning and, according to our hypothesis, misinformation detection. In this work, we consider two LLMs with different parameter scales: (*i*) GPT-2 (small), with 125 million parameters [69], and (*ii*) Llama-3.2 (small), with 1 billion parameters [70]. In doing so, we aim to determine whether smaller and faster models can still serve as effective misinformation detectors, reducing the need to rely on large and costly models. We highlight the following technical aspects:

- Fine-tuning was performed using AdaLoRA [71], a *Parameter-Efficient Fine-Tuning* (PEFT) method that adaptively adjusts the importance of model parameters during training. We set the AdaLoRA hyperparameters to $r = 16$, $\alpha_r = 16$, and a dropout rate of 0.01;
- We employed a *language modeling* objective (i.e., next-step prediction with cross-entropy loss) and used *early stopping* based on validation loss. To address misinformation detection, we introduced an additional *binary classification loss*, weighted ten times more heavily than the language modeling loss. A high-level overview of the dual-loss setup is shown in Fig. 2;
- The binary loss determines whether the model can correctly classify a sentence as truthful or not at the end of its generation, using a suitable *prompt* specific to the dataset under consideration (see Section 4.1). This prompt is not used during generation but rather for *non-autoregressive inference* at each training step. Concretely, the model is prompted to produce a *True* or *False* response without invoking an autoregressive decoding loop;
- At each step, we perform a *forward pass* with the *input sentences* i_t, obtaining predicted *output probabilities* o_t. The language modeling loss $L(o_t, i_{t+1})$ is computed by comparing these probabilities with the *next-step input* i_{t+1};
- For the misinformation task, we similarly extract the *truthfulness label* from i_{t+1} and assess whether the predicted output o_t assigns higher probability to *True* or *False*, effectively casting the task as binary classification. This approach integrates classification directly into the language modeling pipeline, avoiding the need for a separate classification head;
- To further enhance model robustness, we apply *noise injection* to the embedding vectors using NEFTune [72], a technique that introduces controlled perturbations during training to improve generalization.

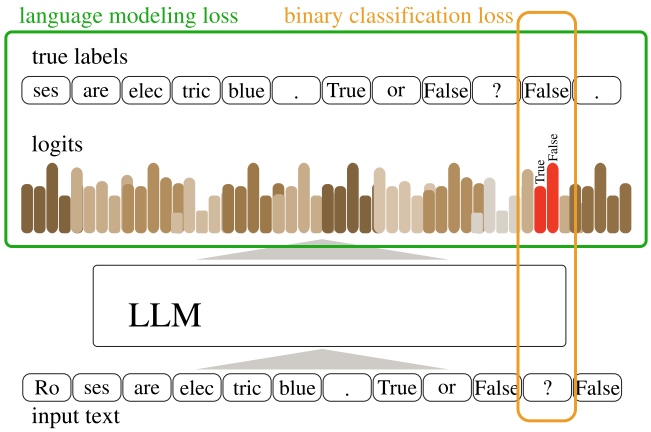

Fig. 2. Fine-tuning the LLM for language modeling with an additional binary classification loss.

Table 3
Statistics of the datasets used for misinformation detection and their partitions.

Name	# samples	Evidence	Max length	Mean length	# stems	Truthful claims
PubHealth	9804/1231/1223	True	41 925	4557.7	68 602	30.6%
VitaminC	370 653/63 054/55 197	True	4148	308.3	87 179	50.1%
ISOT	32 548/4073/4072	False	4999	2178.1	82 426	47.8%

4. Experimental evaluation

In this section, we present the experimental evaluation conducted to assess the effectiveness of the proposed solution for misinformation detection, leveraging LLM fine-tuning with data augmentation. We first introduce the *misinformation datasets* and *prompts* considered, followed by the *evaluation strategy* and *results*.

4.1. Misinformation datasets and prompts

For evaluation purposes, we propose to *augment* any possible *Target Task Dataset* (TTD) meant for the misinformation detection task, in the fine-tuning phase, with data generated using LogicJitter to enhance the logical abilities of LLMs in detecting misinformation. In this work, we consider three datasets used as TTDs: *PubHealth* [73], *VitaminC* [74], and the ISOT *Fake News Dataset* [75].

- *PubHealth* comprises 11,832 claims for fact-checking across a wide range of health-related topics, including biomedical subjects and government healthcare policies;
- *VitaminC* is a multi-domain fact-verification dataset based on Wikipedia edits, containing 488,904 data points;
- ISOT consists of 25,200 articles categorized as either fake or real news. Truthful articles come from *Reuters.com* while fake news comes from websites flagged by *Politifact*.

Both *PubHealth* and *VitaminC* provide, for each *claim* x, a corresponding *piece of evidence* y that either supports or rejects the claim, along with a *truthfulness label* z. In contrast, the ISOT dataset provides longer claims with the corresponding truthfulness labels, but without any associated supporting evidence. Given the nature of the datasets described above, the *prompt* for the *PubHealth* and *VitaminC* datasets employed in the fine-tuning phase is as follows: 'Claim: x. Evidence: y. Does the evidence support the claim? Reply with *True* or *False*: z'. For the ISOT dataset, the prompt is as follows: 'x. Is the preceding text likely truthful and not fake news? reply with *True* or *False*: z'. As illustrated by examples in Table 1, the LogicJitter (LJ) prompt follows the structure: '{*problem*}. {*characters discussion*}. {*revision*}. {*new discussion among the same characters*}. Is character {i} correct, True or False? z'.

In Table 3, we report the statistics of the datasets used for misinformation detection. In particular, we report the number of *training*, *validation*, and *test samples* for each dataset. Some datasets include supporting evidence for each claim, while others (such as ISOT) do not. Sentence length is measured in the number of characters. To estimate vocabulary size, we preprocess the text by lowercasing, removing punctuation and digits, and applying stemming; we then count the number of unique word stems. Most datasets contain a reasonably balanced distribution of true and false claims, with the exception of PubHealth, which is more skewed.

4.2. Evaluation strategy and results

To evaluate the effectiveness of augmenting the TTD dataset with LogicJitter during fine-tuning, we explicitly consider as a *baseline* an LLM trained solely on the original, non-augmented TTD. This setup allows us to isolate and quantify the impact of

Table 4
Ablation study to understand which parts of LogicJitter contribute the most in the fine-tuning (ft) of the LLM. G stands for game description, E for including generations with errors, C for including random characters, and full for GECR, with R for revisions. We show results on the PubHealth, VitaminC and ISOT datasets using GPT-2 and Llama-3.2. Best accuracy in bold, second best underlined.

Augmentation	↑ PubHealth	↑ VitaminC	↑ ISOT
LLM: GPT-2 [69]			
LLM ft on TTD (alone)	31.4%	55.7%	48.6%
LLM ft on TTD + LogicJitter (G)	**42.8**%	50.3%	50.2%
LLM ft on TTD + LogicJitter (GE)	37.7%	51.8%	**64.8**%
LLM ft on TTD + LogicJitter (GEC)	39.3%	54.2%	48.6%
LLM ft on TTD + LogicJitter (full)	31.4%	**61.8**%	48.2%
LLM: Llama-3.2 [70]			
LLM ft on TTD (alone)	60.7%	50.1%	54.9%
LLM ft on TTD + LogicJitter (G)	49.1%	50.1%	73.6%
LLM ft on TTD + LogicJitter (GE)	**69.0**%	**50.2**%	60.8%
LLM ft on TTD + LogicJitter (GEC)	50.3%	50.1%	59.5%
LLM ft on TTD + LogicJitter (full)	42.6%	50.0%	**75.4**%

Table 5
Test accuracy for different augmentation strategies on the PubHealth, VitaminC, and ISOT dataset using GPT-2 and Llama-3.2 model. Best accuracy in bold, second best underlined.

Augmentation	↑ PubHealth	↑ VitaminC	↑ ISOT
LLM: GPT-2 [69]			
+100%	**43.9**%	50.1%	**61.3**%
+50%	31.4%	**61.8**%	48.2%
+25%	31.2%	51.8%	51.9%
TTD (alone)	31.4%	55.7%	48.6%
−25%	35.3%	56.6%	48.1%
−50%	31.4%	50.1%	53.9%
−100%	37.1%	49.9%	48.2%
LLM: Llama-3.2 [70]			
+100%	**66.0**%	50.1%	64.2%
+50%	42.6%	50.0%	**75.4**%
+25%	29.6%	50.1%	67.1%
TTD (alone)	60.7%	50.1%	54.9%
−25%	40.8%	50.1%	75.2%
−50%	35.6%	49.3%	57.2%
−100%	31.4%	**50.2**%	38.2%

LogicJitter on misinformation detection by comparing LLMs trained with and without the augmented dataset. Importantly, our goal is not to directly compare with other LLM-based approaches that rely on supervised methods or external knowledge- or retrieval-based mechanisms, such as those illustrated in Section 2.4, which may perform better on specific domains or tasks. Rather, we aim to assess how LogicJitter enhances the reasoning and intrinsic capabilities of the LLM itself, independently of external supervision, thereby directly measuring its effect on the model's internal ability to detect misinformation.

4.2.1. Ablation study

Table 4 presents the results in terms of *accuracy* on the test set for this task, comparing the baseline with four distinct *augmentation configurations* of LogicJitter, namely:

- G: indicating the usage of only the *game description*;
- E: indicating the inclusion of generations with *errors*;
- C: indicating the inclusion of *random characters*;
- R: indicating the inclusion of *problem revisions* (see Section 3.3).

In the table, these components are assessed both individually and in combination, culminating in the evaluation of the *full* LogicJitter setup (GECR). Results are reported on the *PubHealth*, *VitaminC*, and *ISOT* datasets, using GPT-2 [69] and Llama-3.2 [70] baseline models fine-tuned with AdaLoRA. For these experiments, data augmentation is performed by adding a number of augmented samples equal to 50% of the original training set size for the target task. We emphasize that the results reported for GPT-2 and LLaMA-3.2 are derived from our own experiments rather than from previously published results. For these models, we report the best performance achieved via a grid search over learning rates. In contrast, for LogicJitter, results are reported using its best-performing baseline configuration.

4.2.2. Brief discussion of the results

As shown in the table, augmenting the TTD with LogicJitter consistently improves test performance across all datasets and models, compared to the non-augmented baseline. Notably, LogicJitter achieves these gains without relying on human-labeled data for misinformation detection, nor on LLM-generated content, which may exhibit uncertain factual accuracy. The full version of LogicJitter appears particularly effective; however, components G and E emerge as the most consistently impactful in driving performance improvements.

In particular, Table 5 reports the *test accuracy* of applying *different percentages of data augmentation* in the fine-tuning phase compared to the size of the original TTD. This indicates the amount of data added (+) or removed (−) from the original TTD as part of the augmentation process. In this setup, a +25% augmentation indicates that each dataset was expanded by adding a number of augmented samples equal to 25% of the size of the target task's original training set. Conversely, a −25% setting refers to replacing 25% of the original training samples with augmentation data, keeping the total size unchanged. Results are reported for the *PubHealth*, *VitaminC*, and ISOT datasets. Evaluations were conducted using GPT-2 and Llama-3.2 models fine-tuned with AdaLoRA as the PEFT method.

Across all configurations, data augmentation consistently improves performance. As expected, adding augmentation data generally leads to greater gains, though replacing part of the training set with augmented samples sometimes yields positive results as well. Notably, Llama struggled with the VitaminC dataset without augmentation regardless of the learning rate, and also when the training set was augmented. However, Llama showed clear benefits from LogicJitter augmentation on the smaller datasets. Surprisingly, the smaller GPT-2 did not struggle as much detecting misinformation on the largest dataset, suggesting the possibility of a role for smaller models for misinformation detection.

5. Conclusions

In this work, we introduced LogicJitter, an algorithmically generated data augmentation approach used to fine-tune LLMs for misinformation detection, without the need for human- or LLM-labeled data. We demonstrated that LogicJitter appears to improve model generalization compared to fine-tuning exclusively on the target dataset or augmenting with existing human-labeled misinformation data. By leveraging algorithmic generation, our approach transforms a typically costly task, expert annotation, or an often unreliable alternative, AI labeling, into a more scalable and cost-effective solution. At this stage of development, the reliability of LogicJitter requires further in-depth investigation to establish clear guidelines on the optimal amount of augmentation needed, and to determine whether the inclusion of characters and revisions is essential or if games with errors alone suffice.

Algorithmic dataset generation offers several advantages: for instance, the dataset is perfectly balanced with respect to true and false statements, and it is explicitly designed to avoid stereotyping biases entirely. Interestingly, our approach leverages traditionally "old-school" rule-based AI methods — such as context-free grammars and causal networks — to overcome certain limitations of contemporary deep learning techniques.

Our results support both **RQ1** and **RQ2**, demonstrating that rule-based games, crafted to counter cognitive biases and logical fallacies, effectively enhance LLMs' logical reasoning and improve their capability to detect misinformation.

An alternative approach could involve leveraging existing human-labeled datasets, such as GLUE [76], by presenting incorrect labels and asking the LLM to estimate the veracity of the answers. However, we opted for a purely rule-based methodology to clearly isolate and assess its effectiveness.

As a promising future direction, incorporating fuzzy logic statements could provide greater flexibility in handling uncertainty within the games. Additionally, expanding the framework to address a broader range of cognitive biases and fallacies would be valuable. Furthermore, a rule-based system like LogicJitter could be extended to generate algorithmic explanations for why an answer is correct or incorrect, enhancing its utility in contexts where explainability is crucial. Interestingly, the smaller model performed relatively well in identifying misinformation within the largest dataset, indicating that smaller models might have a valuable role in misinformation detection.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors utilized GPT-4o and Grammarly to enhance language and readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was partly funded by the European Union – Next Generation EU, Mission 4, Component 2, CUP: D53D23008480001 (20225WTRFN - KURAMi: *Knowledge-based, explainable User empowerment in Releasing private data and Assessing Misinformation in online environments*).[3] We acknowledge ISCRA for awarding this project access to the LEONARDO supercomputer [77], owned by the EuroHPC Joint Undertaking, hosted by CINECA (Italy).

[3] https://kurami.disco.unimib.it/

Data and code availability

Data and code will be made available on request.

References

[1] Wardle C, Derakhshan H. Information disorder: Toward an interdisciplinary framework for research and policymaking, vol. 27, Council of Europe Strasbourg; 2017.
[2] Wu L, Morstatter F, Carley KM, Liu H. Misinformation in social media: definition, manipulation, and detection. ACM SIGKDD Explor Newsl 2019;21(2):80–90.
[3] OECD. Facts not fakes: Tackling disinformation, strengthening information integrity. 2024, p. 141.
[4] Hu L, Wei S, Zhao Z, Wu B. Deep learning for fake news detection: A comprehensive survey. AI Open 2022;3:133–55.
[5] Reis JC, Correia A, Murai F, Veloso A, Benevenuto F. Supervised learning for fake news detection. IEEE Intell Syst 2019;34(2):76–81.
[6] Viviani M, Pasi G. Credibility in social media: opinions, news, and health information—a survey. Wiley Interdiscip Rev: Data Min Knowl Discov 2017;7(5):e1209.
[7] La Barbera D, Milanese GC, Peikos G, Pasi G, Viviani M. Beyond Binary Classification: Ranking for Information Access in Misinformation Contexts. In: Proceedings of Ital-IA 2025: convegno nazionale CINI sull'intelligenza artificiale. 2025.
[8] Clarke CL, Maistro M, Smucker MD, Zuccon G. Overview of the TREC 2020 health misinformation track. In: TREC. 2020.
[9] Cabitza F, Ciucci D, Pasi G, Viviani M. Responsible AI in healthcare. 2022, arXiv preprint arXiv:2203.03616.
[10] Paka WS, Bansal R, Kaushik A, Sengupta S, Chakraborty T. Cross-SEAN: A cross-stitch semi-supervised neural attention model for COVID-19 fake news detection. Appl Soft Comput 2021.
[11] Khraisat A, Manisha, Chang L, Abawajy J. Survey on deep learning for misinformation detection: Adapting to recent events, multilingual challenges, and future visions. Soc Sci Comput Rev 2026;44(2):209–30.
[12] Silva A, Luo L, Karunasekera S, Leckie C. Unsupervised domain-agnostic fake news detection using multi-modal weak signals. 2023, arXiv Preprint.
[13] Gong S, Sinnott RO, Qi J. Fake news detection through graph-based neural networks: A survey. 2023, arXiv Preprint.
[14] Lakzaei B, Haghir Chehreghani M, Bagheri A. Disinformation detection using graph neural networks: a survey. Artif Intell Rev 2024.
[15] Hu B, Sheng Q, Cao J, Shi Y, Li Y, Wang D, Qi P. Bad actor, good advisor: Exploring the role of large language models in fake news detection. In: Proc. of the AAAI conf. on artificial intelligence, vol. 38, (20):2024, p. 22105–13.
[16] Papageorgiou E, Chronis C, Varlamis I, Himeur Y. A survey on the use of large language models (LLMs) in fake news. Futur Internet 2024;16(8):298.
[17] Ji Z, Lee N, Frieske R, Yu T, Su D, Xu Y, Ishii E, Bang YJ, Madotto A, Fung P. Survey of hallucination in natural language generation. ACM Comput Surv 2023;55(12):1–38.
[18] Zhao H, Chen H, Yang F, Liu N, Deng H, Cai H, Wang S, Yin D, Du M. Explainability for large language models: A survey. ACM Trans Intell Syst Technol 2024;15(2):1–38.
[19] Heersmink R, de Rooij B, Clavel Vázquez MJ, Colombo M. A phenomenology and epistemology of large language models: Transparency, trust, and trustworthiness. Ethics Inf Technol 2024;26(3):41.
[20] Hwang J, Park J, Park H, Kim D, Park S, Ok J. Retrieval-augmented generation with estimation of source reliability. In: Proceedings of the 2025 conference on empirical methods in natural language processing. 2025, p. 34267–91.
[21] French AM, Storey VC, Wallace L. The impact of cognitive biases on the believability of fake news. Eur J Inf Syst 2023;1–22.
[22] Stanovich KE. The fundamental computational biases of human cognition: Heuristics that (sometimes) impair decision making and problem solving. Psychol Probl Solving 2003;291–342.
[23] Zini S, Gomez-Villa A, Buzzelli M, Twardowski B, Bagdanov AD, van de weijer J. Planckian Jitter: countering the color-crippling effects of color Jitter on self-supervised training. In: The eleventh int. conf. on learning representations. 2023.
[24] Gunasekar S, Zhang Y, Aneja J, Mendes CCT, Del Giorno A, Gopi S, Javaheripi M, Kauffmann P, de Rosa G, Saarikivi O, et al. Textbooks are all you need. 2023, arXiv preprint arXiv:2306.11644.
[25] Trinh TH, Wu Y, Le QV, He H, Luong T. Solving olympiad geometry without human demonstrations. Nature 2024;625(7995):476–82.
[26] Veličković P, Badia AP, Budden D, Pascanu R, Banino A, Dashevskiy M, Hadsell R, Blundell C. The CLRS algorithmic reasoning benchmark. In: Int. conf. on machine learning. PMLR; 2022, p. 22084–102.
[27] Markeeva L, McLeish S, Ibarz B, Bounsi W, Kozlova O, Vitvitskyi A, Blundell C, Goldstein T, Schwarzschild A, Veličković P. The CLRS-text algorithmic reasoning language benchmark. 2024, arXiv preprint arXiv:2406.04229.
[28] Kallini J, Papadimitriou I, Futrell R, Mahowald K, Potts C. Mission: Impossible language models. In: Annual meeting of the association for computational linguistics. 2024.
[29] Vashishtha A, Kumar A, Reddy AG, Balasubramanian VN, Sharma A. Teaching transformers causal reasoning through axiomatic training. In: ICML workshop on large language models and cognition. 2024.
[30] Zelikman E, Wu Y, Mu J, Goodman N. Star: Bootstrapping reasoning with reasoning. Adv Neural Inf Process Syst 2022;35:15476–88.
[31] Gou Z, Shao Z, Gong Y, yelong shen, Yang Y, Huang M, Duan N, Chen W. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In: The twelfth int. conf. on learning representations. 2024.
[32] Jin Z, Chen Y, Leeb F, Gresele L, Kamal O, Lyu Z, Blin K, Adauto FG, Kleiman-Weiner M, Sachan M, Schölkopf B. CLadder: A benchmark to assess causal reasoning capabilities of language models. In: Oh A, Naumann T, Globerson A, Saenko K, Hardt M, Levine S, editors. Advances in neural information processing systems. vol. 36. Curran Associates, Inc.; 2023, p. 31038–65.
[33] Jin Z, Liu J, Lyu Z, Poff S, Sachan M, Mihalcea R, Diab M, Schölkopf B. Can large language models infer causation from correlation? 2024, arXiv preprint arXiv:2306.05836.
[34] Trung L, Zhang X, Jie Z, Sun P, Jin X, Li H. Reft: Reasoning with reinforced fine-tuning. In: Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: long papers). 2024, p. 7601–14.
[35] Zeng W, Huang Y, Zhao L, Wang Y, Shan Z, He J. B-STAR: Monitoring and balancing exploration and exploitation in self-taught reasoners. In: International conference on learning representations. 2025.
[36] Ma C, Zhao H, Zhang J, He J, Kong L. Non-myopic generation of language models for reasoning and planning. In: International conference on learning representations. 2025.
[37] Zhang H, Kung P-N, Yoshida M, Van den Broeck G, Peng N. Adaptable logical control for large language models. Adv Neural Inf Process Syst 2024;37:115563–87.
[38] Yang L, Yu Z, Zhang T, Xu M, Gonzalez JE, Cui B, Yan S. SuperCorrect: Supervising and correcting language models with error-driven insights. In: International conference on learning representations. 2025.
[39] Martínez N, Agudo U, Matute H. Human cognitive biases present in artificial intelligence. Rev Int Los Estud Vascos 2022;67(2).

[40] Campbell H, Goldman S, Markey PM. Artificial intelligence and human decision making: Exploring similarities in cognitive bias. Comput Hum Behav: Artif Humans 2025;4:100138. http://dx.doi.org/10.1016/j.chbah.2025.100138, URL https://www.sciencedirect.com/science/article/pii/S2949882125000222.
[41] Nguyen JK. Human bias in AI models? Anchoring effects and mitigation strategies in large language models. J Behav Exp Financ 2024;43:100971.
[42] Chen Y, Kirshner SN, Ovchinnikov A, Andiappan M, Jenkin T. A manager and an AI walk into a bar: does ChatGPT make biased decisions like we do? Manuf Serv Oper Manag 2025.
[43] Hu X, Luo L, Fleming SM. A role for metamemory in cognitive offloading. Cognition 2019;193:104012.
[44] Gong C, Yang Y. Google effects on memory: a meta-analytical review of the media effects of intensive internet search behavior. Front Public Health 2024;12:1332030.
[45] Gerlich M. AI tools in society: Impacts on cognitive offloading and the future of critical thinking. Societies 2025;15(1):6.
[46] Fan Y, Tang L, Le H, Shen K, Tan S, Zhao Y, Shen Y, Li X, Gašević D. Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance. Br J Educ Technol 2025;56(2):489–530.
[47] Vicente L, Matute H. Humans inherit artificial intelligence biases. Sci Rep 2023;13(1):15737.
[48] Jin Z, Lalwani A, Vaidhya T, Shen X, Ding Y, Lyu Z, Sachan M, Mihalcea R, Schoelkopf B. Logical fallacy detection. In: Findings of the association for computational linguistics: EMNLP 2022. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics; 2022, p. 7180–98.
[49] Fallis D. What is disinformation? Libr Trends 2015;63(3):401–26.
[50] Chen C, Shu K. Combating misinformation in the age of LLMs: Opportunities and challenges. AI Mag 2024;45(3). http://dx.doi.org/10.1002/aaai.12188.
[51] Chen C, Shu K. Can LLM-generated misinformation be detected? In: The twelfth international conference on learning representations. ICLR, 2024, URL https://openreview.net/forum?id=ccxD4mtkTU.
[52] Truică C-O, Apostol E-S. It's all in the embedding! fake news detection using document embeddings. Mathematics 2023;11(3):508.
[53] Pavlyshenko BM. Analysis of disinformation and fake news detection using fine-tuned large language model. 2023, arXiv preprint arXiv:2309.04704.
[54] Jiang B, Tan Z, Nirmal A, Liu H. Disinformation detection: An evolving challenge in the age of llms. In: Proceedings of the 2024 siam international conference on data mining. SIAM; 2024, p. 427–35.
[55] Schlichtkrull M, Guo Z, Vlachos A. Averitec: A dataset for real-world claim verification with evidence from the web. Adv Neural Inf Process Syst 2024;36.
[56] Bai Y, Fu K. A large language model-based fake news detection framework with RAG fact-checking. In: 2024 IEEE international conference on big data. IEEE; 2024, p. 8617–9.
[57] Milanese GC, Peikos G, Pasi G, Viviani M. Fact-driven health information retrieval: Integrating LLMs and knowledge graphs to combat misinformation. In: European conference on information retrieval. 2025, p. 192–200.
[58] Huang T, Yi J, Yu P, Xu X. Unmasking digital falsehoods: A comparative analysis of LLM-based misinformation detection strategies. In: 2025 8th international conference on advanced algorithms and control engineering. ICAACE, IEEE; 2025, p. 2470–6.
[59] Pelrine K, Imouza A, Thibault C, Reksoprodjo M, Gupta CA, Christoph JN, Godbout JF, Rabbany R. Towards reliable misinformation mitigation: Generalization, uncertainty, and GPT-4. In: Proceedings of the 2023 conference on empirical methods in natural language processing. EMNLP, Singapore: Association for Computational Linguistics; 2023, p. 6399–429. http://dx.doi.org/10.18653/v1/2023.emnlp-main.395.
[60] Van Eyghen H. Cognitive bias: Phylogenesis or ontogenesis? Front Psychol 2022;13:1–4.
[61] Gigerenzer G. Adaptive thinking: Rationality in the real world. Oxford University Press; 2002.
[62] Hilbert M. Toward a synthesis of cognitive biases: how noisy information processing can bias human decision making. Psychol Bull 2012;138(2):211.
[63] Nye M, Andreassen AJ, Gur-Ari G, Michalewski H, Austin J, Bieber D, Dohan D, Lewkowycz A, Bosma M, Luan D, et al. Show your work: Scratchpads for intermediate computation with language models, 2021. 2021, URL https://arxiv.org/abs/2112.00114.
[64] Albert R, Barabási A-L. Statistical mechanics of complex networks. Rev Modern Phys 2002;74(1):47.
[65] Hoperoft J, Ullman J. Introduction to automata theory, languages, and computation. Addison-Wesley Publishing Company; 1979.
[66] Johnson J, Hariharan B, Van Der Maaten L, Fei-Fei L, Lawrence Zitnick C, Girshick R. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In: Proc. of the IEEE conference on computer vision and pattern recognition. 2017, p. 2901–10.
[67] Abdelnour J, Salvi G, Rouat J. CLEAR: A dataset for compositional language and elementary acoustic reasoning. In: Visually grounded interaction and language workshop. 2018.
[68] Abdelnour J, Rouat J, Salvi G. NAAQA: A neural architecture for acoustic question answering. IEEE Trans Pattern Anal Mach Intell 2023;45(4):4997–5009.
[69] Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I, et al. Language models are unsupervised multitask learners. OpenAI Blog 2019;1(8):9.
[70] Meta Llama Team. The Llama 3 Herd of Models. 2024.
[71] Zhang Q, Chen M, Bukharin A, He P, Cheng Y, Chen W, Zhao T. Adaptive budget allocation for parameter-efficient fine-tuning. In: The eleventh int. conf. on learning representations. 2023.
[72] Jain N, yeh Chiang P, Wen Y, Kirchenbauer J, Chu H-M, Somepalli G, Bartoldson BR, Kailkhura B, Schwarzschild A, Saha A, Goldblum M, Geiping J, Goldstein T. NEFTune: Noisy embeddings improve instruction finetuning. In: The twelfth int. conf. on learning representations. 2024.
[73] Kotonya N, Toni F. Explainable automated fact-checking for public health claims. 2020, arXiv preprint arXiv:2010.09926.
[74] Schuster T, Fisch A, Barzilay R. Get your vitamin c! robust fact verification with contrastive evidence. In: Proc. of the 2021 conf. of the North American chapter of the association for computational linguistics: human language technologies. Online: Association for Computational Linguistics; 2021, p. 624–43.
[75] Ahmed H, Traore I, Saad S. Detection of online fake news using n-gram analysis and machine learning techniques. In: Intelligent, secure, and dependable systems in distributed and cloud environments: first int. conf., ISDDC 2017, vancouver, BC, Canada, October 26-28, 2017, proc. 1. Springer; 2017, p. 127–38.
[76] Wang A, Singh A, Michael J, Hill F, Levy O, Bowman S. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Proc. of the 2018 EMNLP workshop blackboxNLP: analyzing and interpreting neural networks for NLP. Brussels, Belgium: Association for Computational Linguistics; 2018, p. 353–5. http://dx.doi.org/10.18653/v1/W18-5446.
[77] Turisini M, Amati G, Cestari M, CINECA SuperComputing Centre, SuperComputing Applications and Innovation Department. LEONARDO: A Pan-European Pre-Exascale Supercomputer for HPC and AI applications. J Large-Scale Res Facil 2024;9(1).