ThinknCheck: Advancing Claim Verification with Compact, Reasoning-Driven, and Interpretable Models

Anonymous ACL submission

Abstract

We introduce ThinknCheck, a reasoning-optimized claim verification model that explicitly generates explanation chains before making verification decisions. This Gemma3-based 1B parameter model, fine-tuned on our new LLMAggreFact-Think dataset, achieves 78.1% balanced accuracy on the LLMAggreFact benchmark, outperforming the 7B MiniCheck model (current SOTA) while requiring substantially less computational resources. Explicit reasoning significantly enhances verification accuracy (+20.6 points over non-reasoning ablation) and improves out-of-domain generalization (+14.7 points on scientific claims). Qualitative analysis of reasoning traces revealed distinct patterns, with surface-level evidence matching dominating current datasets; complex synthesis in claim verification remains underrepresented. To evaluate numerical reasoning, we contribute GSM-Claims, a dataset reformulating grade school math problems as verification tasks. Error analysis identified domain-specific patterns, informing our specialized ThinknCheck-Science variant with substantial performance gains across all benchmarks. Reasoning-first approaches are a promising direction for more accurate, edge-device friendly, interpretable, and generalizable claim verification systems across diverse domains.

Claim: Water boils at 100C.
Document: The temperature at which water boils is 212F.

MiniCheck-7B verification result:
`DOES NOT SUPPORT`

ThinknCheck-Science-1B result:
`SUPPORTS`

ThinknCheck-Science-1B reasoning:
The document states that the temperature at which water boils is 212F. The claim states that water boils at 100C, which is equivalent to 100F. To determine if the claim is true, I need to convert 100C to Fahrenheit. 100C is equal to 212F, which is the same as the boiling point of water. Thus, the claim is supported by the information in the document.

Figure 1: Framing claim verification as a natural language entailment problem falls short on many problems requiring additional reasoning effort. This simple example illustrates how a state-of-the-art model `Bespoke-MiniCheck-7B` (Tang et al., 2024) fails to recognize that $100°C$ is equivalent to $212°F$. Performing this sort of reasoning is essential for scientific and other complex claim verification. Our work, ThinknCheck, and particularly ThinknCheck-Science, explicitly generates reasoning prior to verification leading to accurate results in such problems. For a complex example, see Figure 7.

1 Introduction

Large Language Models (LLMs) are increasingly pivotal in modern AI-driven workflows involving natural language claim verification (Thorne et al., 2018), yet their propensity for hallucinations (Zhang et al., 2023), reasoning errors (Chen et al., 2023), and their opaque nature limit broader deployment, particularly in high-stakes domains like healthcare and scientific innovation.

Current claim verification approaches, while evolving, often face challenges. Some methods can be computationally intensive, requiring multiple LLM calls for a single verification (Malaviya et al., 2024; Jacovi et al., 2024a). Verification using large, closed-source models also raises concerns regarding cost, privacy, and data security. While the trend in general reasoning models has seen the development of very large systems, such as OpenAI's o-series and DeepSeek's R1 (DeepSeek-AI et al., 2025), which aim for broad reasoning capabilities, there is a concurrent need for smaller, more specialized models (Tang et al., 2024) that can perform robustly on specific tasks like claim verification, es-

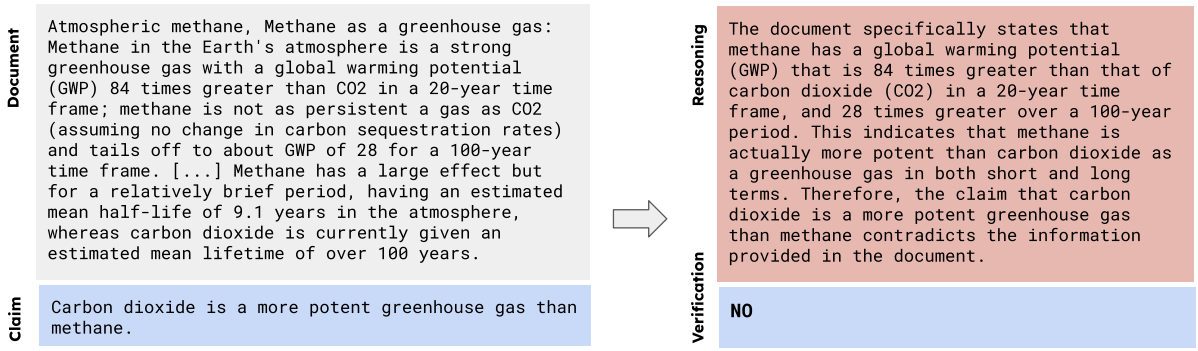


Figure 2: A sample from the LLMAggreFact-Think dataset, which also illustrates our formulation of the claim verification task: Given a pair of claim and document, our goal is to produce cogent reasoning in addition to the verification label. The `[...]` represents parts of the reasoning tokens that we elided to accommodate the example in this figure.

pecially in resource-constrained environments. Our work aligns with this latter direction, focusing on creating efficient yet powerful verification models.

To address these challenges, we introduce ThinknCheck, a suite of novel low-footprint claim verification models that explicitly generate structured reasoning chains *before* rendering a verification decision. Specifically ThinknCheck is a 4-bit quantized 1B parameter Gemma3 (GemmaTeam et al., 2025) model, fine-tuned on our newly createdLLMAggreFact-Think dataset—a version of theLLMAggreFact benchmark (Tang et al., 2024) that we augmented with explicit reasoning traces. As illustrated in Figure 1, explicitly generating reasoning allows ThinknCheck to handle claims that require multi-step inference, a common scenario where prior models falter. Our contributions are as follows:

- Introduce ThinknCheck, a reasoning-optimized model that improves claim verification accuracy and interpretability by first generating explicit explanation chains.
- Demonstrate ThinknCheck's explicit reasoning significantly boosts verification accuracy (+20.6 points over non-reasoning ablation) and substantially improves out-of-domain generalization (+14.7 points on scientific claims).
- Create and release GSMClaims, a novel benchmark from reformulated grade school math problems, to evaluate arithmetic reasoning capabilities in claim verification systems.
- Develop ThinknCheck-Science, a specialized variant optimized for scientific and mathematical verification, achieving significant performance improvements across relevant benchmarks.
- Reveal domain-specific verification strategies, current dataset limitations, and future research insights through a comprehensive analysis of reasoning traces from LLMAggreFact-Think.
- Release all created datasets and models under an Apache 2.0 license[1] to facilitate further research in this critical area.

These contributions establish reasoning-first approaches as a promising path towards more accurate, efficient, interpretable, and broadly generalizable claim verification systems, offering an alternative to sheer model scaling.

2 Related Work

Our work intersects with several research areas: claim verification and fact-checking, the integration of reasoning into language models, the development of lightweight verification models, the creation of benchmark datasets, and strategies for reasoning supervision.

Claim Verification and Fact-Checking: Claim verification, spurred by datasets like FEVER (Thorne et al., 2018) and SciFact (Wadden et al., 2020), typically involves classification models predicting a claim's validity against a document (Tang et al., 2024). The opacity and hallucination risks in LLMs (Chen et al., 2023; Zhang et al., 2023) necessitate verifiers ensuring accuracy and interpretability.

Reasoning-Augmented Verification Models: Explicit reasoning enhances verification. Chain-of-Thought (CoT) prompting (Wei et al., 2022) improves reasoning via rationales, adapted by methods like verifiable CoT (Jacovi et al., 2024b) and

[1]URL withheld for blind-review

the ReAct framework (Yao et al., 2023) which interleaves reasoning with actions. While large models like OpenAI's o-series and DeepSeek R1 (DeepSeek-AI et al., 2025) advance general reasoning, our ThinknCheck fine-tunes a compact model for structured, pre-decision reasoning specific to claim verification.

Lightweight and Specialized Verification Models: Interest in smaller, efficient, specialized models for broader deployment is growing (Allal et al., 2025). `MiniCheck` exemplified this for claim verification; this 7B parameter model, trained on synthetic data, outperformed AlignScore (Zha et al., 2023) onLLMAggreFact. Yet, `MiniCheck` falters on multi-step reasoning (Figure 1), lacks explanations vital for trust and collaboration (Bansal et al., 2020; Fan et al., 2021; Javaid and Estivill-Castro, 2021), and their best model is still resource-heavy. ThinknCheck (1B parameters) addresses this by outperforming `MiniCheck-7B` with explicit, efficient reasoning.

Benchmark Datasets for Verification: Comprehensive benchmarks are vital.LLMAggreFact aggregates nine datasets (Tang et al., 2022; Nallapati et al., 2016; Narayan et al., 2018; Zhu et al., 2021; Hu et al., 2023; Liu et al., 2023; Malaviya et al., 2023; Wang et al., 2023; Kamoi et al., 2023) for diverse claim verification scenarios. While domain-specific benchmarks like SciFact address scientific claims, and GSM8K (Cobbe et al., 2021) is used for math reasoning, we introduce GSMClaims by reformulating GSM8K problems to directly test numerical reasoning in verification.

Reasoning Supervision and Data Augmentation Strategies: Supervised fine-tuning (SFT) on synthetic reasoning traces, a form of knowledge distillation (Xu et al., 2024), trains smaller models to emulate larger ones. Reinforcement learning (RL) techniques, e.g. GRPO (Shao et al., 2024), also optimize reasoning. We chose SFT due to sufficient supervised data, acknowledging preference optimization methods like GRPO as an alternative.

3 Problem Formulation

The standard formulation of evidence-backed claim verification, as used by Tang et al. (2024) and predecessors, is a classification task: a discriminator $\mathcal{M}$ maps a claim (from space $\mathcal{C}$) and document (from space $\mathcal{D}$) to a discrete label in $\{0, 1\}$ (1 for supported, 0 otherwise).

$$\mathcal{M} : \mathcal{C} \times \mathcal{D} \rightarrow \{0, 1\}$$

Our work extends this by incorporating explicit reasoning. We define this task with a reasoner $\mathcal{R}$ that maps the input claim-document pair to a reasoning trace $\mathcal{T}$ and a boolean verification label:

$$\mathcal{R} : \mathcal{C} \times \mathcal{D} \rightarrow \mathcal{T} \times \{0, 1\}$$

This richer output format enhances interpretability and aims to improve accuracy by requiring the model to articulate its reasoning. We adopt the binary labels SUPPORTED (1) and NOTSUPPORTED (0) from prior work, treating "REFUTES" and "NOTSUPPORTED" identically[2].

4 Dataset and Model Development

This section details the creation of our training dataset LLMAggreFact-Think and the model training procedures for ThinknCheck.

4.1 LLMAggreFact-Think Dataset Construction

To train our reasoning-based verifier, we created LLMAggreFact-Think by augmenting the 30.4K examples in the LLMAggreFact development set with reasoning chains. Using zero-shot prompting, GPT-4o-mini[3] generated a step-by-step reasoning process and a YES/NO verification label for each (document, claim) pair — see Figure 2; prompt in Appendix A.

For high-quality reasoning, we filtered instances where GPT-4o-mini's generated label mismatched the original LLMAggreFact label, reducing the dataset from 30.4K to 24.1K examples[4]. This filtered set, LLMAggreFact-Think, contains 4-tuples: (claim, document, verification label, reasoning). We opted against using reasoning traces from `Deepseek R1` (DeepSeek-AI et al., 2025) due to their verbosity and token inefficiency (See Figure 3). To ensure quality, we randomly sampled 100 samples across all 9 datasets in LLMAggreFact and manually inspected the reasoning traces derived from GPT-4o and found them to be accurate.

[2]We concur with Tang et al. (2024) that "REFUTES", common in general NLI problems, is rare in claim verification.

[3]Accessed on March 4, 2025. We did not use the o-series models for this as it does not provide access to raw reasoning tokens.

[4]Notably, ∼21% of LLMAggreFact dev set labels differed from GPT-4o-mini's predictions; analyzing this discrepancy is beyond this paper's scope. Hence we chose to only train on examples with agreement.

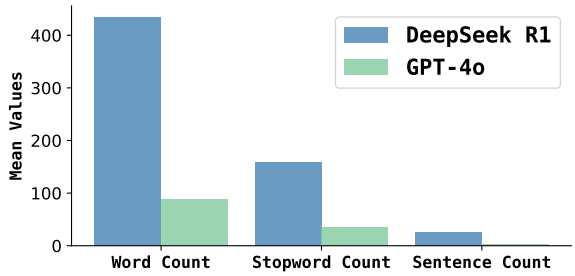


Figure 3: Comparing reasoning traces derived from Deepseek R1 vs. gpt-4o with the same prompt (c.f. Appendix A) on claim verification problems in LLAggreFact. These metrics showed high variance (which we didn't plot here for legibility) for R1 and low variance for GPT-4o. GPT-4o is significantly token efficient compared to R1, making it our choice for harvesting reasoning traces.

4.2 ThinknCheck-1B Model Training

We implemented ThinknCheck-1B by fine-tuning a 4-bit quantized Gemma3 1B model on LLMAggreFact-Think (training details in Appendix B). Our choice of Gemma3 was inspired by its recency and also by its overall performance across diverse LLM benchmarks (GemmaTeam et al., 2025). The fine-tuning prompt (Appendix C) mirrored the LLMAggreFact-Think data structure, constraining the model to output both reasoning and the final verification solution.[5]

4.3 Ablation Model: ThinknCheck-nothink-1B

To isolate the reasoning step's impact, we trained an ablation model, ThinknCheck-nothink-1B. It shares ThinknCheck-1B's architecture, data, and hyperparameters but was trained with a prompt (Appendix D) requesting only the final solution, omitting reasoning generation. This ablation ensures that observed performance gains are not solely due to our choice of Gemma3 as the backbone.

5 Uncovering Reasoning Methods Stressed by Current Claim Verification Datasets

The LLMAggrefact benchmark aggregates 9 claim verification datasets. These datasets (see 2) are highly cited in claim verification literature, yet there is poor understanding about the complexity of the claim verification challenges posed by these datasets. ThinknCheck's reasoning traces provide us an opportunity to understand the reasoning demands exercised by current claim verification datasets, and hence the complexity of the datasets themselves. To do so, we conducted a qualitative analysis of reasoning traces.

5.1 Methodology

We sampled 1,000 examples from the LLMAggreFact-Think dataset along with their generated reasoning traces, using stratified sampling based on the 'dataset' column. Our analysis involved a systematic manual review of reasoning outputs for these document-claim pairs. For each entry, we determined the primary strategy employed to justify the verification label (support or refute). Through an iterative process, we categorized these strategies, identified recurring patterns, and selected representative examples to illustrate the diverse reasoning approaches observed.

5.2 Identifying Reasoning Patterns in LLMAggreFact-Think

Our analysis revealed several distinct reasoning patterns. In this section, we detail these patterns. For illustrative examples of each pattern, please refer to Appendix F.

Direct Evidence Extraction & Matching: The most prevalent strategy involves identifying and often directly quoting or closely paraphrasing specific text segments from the document that explicitly support or contradict the claim. This demonstrates a reliance on surface-level textual matching.

Absence of Evidence Identification: A substantial portion of reasoning concludes that a claim cannot be verified due to insufficient relevant information in the document. These justifications explicitly state that the document does not address the topic or specific details asserted in the claim.

Synthesis of Multiple Information Points: Some reasoning requires integrating information from multiple sentences or sections within the document. This approach is particularly common for claims that summarize findings (e.g., from multiple reviews) or when evidence is distributed throughout the text.

Addressing Scope and Specificity Mismatches: Reasoning frequently addresses discrepancies in scope between the claim and the document. This includes cases where the claim is broader, narrower, or introduces elements not discussed in the source text.

[5] Inference uses parameters recommended by the Gemma3 paper: temperature=1.0, top_p=0.95, and top_k=64.

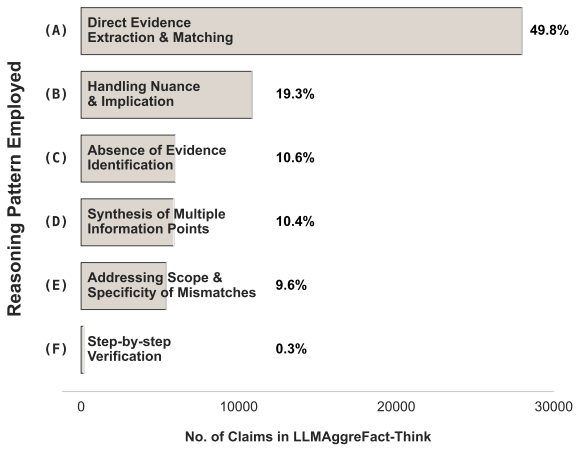


Figure 4: Distribution of reasoning patterns across LLMAggreFact-Think. Direct evidence extraction (A) dominates the verification strategies (27,988 instances), followed by other reasoning strategies. See Section 5.3 for a detailed discussion.

Handling Nuance and Implication: More sophisticated reasoning involves interpreting implications or nuances in the text, even without explicit statements. This may include inferring support based on context or acknowledging partial agreement with the claim.

Step-by-Step Verification: For claims related to processes, instructions, or sequences (e.g., recipes), the reasoning often involves methodically verifying each step mentioned in the claim against the procedures described in the document.

5.3 Insights from Analysis of Generated Reasoning

Our analysis (see Figure 4) reveals that the primary reasoning demand across all datasets for claim verification is that information retrieval and textual entailment capabilities, enabling precise matching between claims and supporting evidence. The distribution of reasoning patterns across 9 datasets (see Figure 8) reveals several significant implications for claim verification systems:

1. **Pattern Dominance Hierarchy:** Direct evidence extraction dominates across all datasets (75-100%), followed by nuance handling (10-50%), and absence identification (1-70%), indicating a clear preference gradient in verification strategies employed by these datasets.
2. **Dataset-Specific Biases:** Certain datasets heavily favor particular reasoning patterns. AggreFact-CNN, for instance, demands information synthesis in approximately 50% of cases, creating a substantial bias toward this reasoning pattern. In contrast, even RAGTruth—the dataset with the highest prevalence of multi-step verification—requires this complex reasoning in merely 0.9% of instances, highlighting a critical gap in current benchmarking resources.
3. **Complementary Pattern Distribution:** Datasets with high rates of direct extraction (pattern A) tend to have lower rates of synthesis (pattern D), suggesting these approaches may be complementary rather than co-occurring.
4. **Verification Complexity Indicators:** The prevalence of scope mismatch handling (E) and nuance interpretation (B) in datasets like FactCheck-GPT and Wice points to the challenging nature of claims requiring contextual understanding beyond literal matching.
5. **Task Formulation Effects:** The stark variation in absence identification (C) across datasets (from <1% in AggreFact-CNN to >70% in FactCheck-GPT) suggests that task formulation significantly influences how systems approach verification when evidence is lacking.

These findings suggest that 1) Comprehensive claim verification systems should balance multiple reasoning strategies rather than optimizing for direct evidence matching alone, with particular attention to the underrepresented but critical capabilities of handling missing evidence and synthesizing distributed information., and 2) Current claim-verification datasets, as represented by the LLMAggreFact benchmark, dominantly test direct evidence matching and high-scoring systems on these datasets will likely not generalize well on complex claim verification that require reasoning beyond evidence matching.

6 Probing Complex Reasoning Capabilities of Claim Verification Models

Current fact verification systems often struggle with claims requiring multi-step reasoning processes, particularly those involving numerical calculations (as demonstrated in Figure 1). While the LLMAggreFact benchmark effectively evaluates core verification capabilities and generalization, it inadequately tests these more complex reasoning scenarios. To address this limitation, we developed both a specialized benchmark and a model variant specifically designed to tackle such challenges.

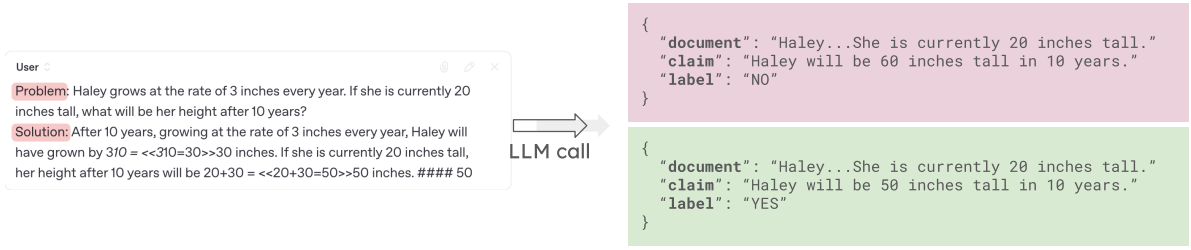


Figure 5: To build GSMClaims, we reframe GSM8K (Cobbe et al., 2021) problems as claim verification problems requiring arithmetic processing. See Appendix E for the prompt used in the LLM call.

6.1 GSMClaims Dataset for Arithmetic Reasoning

We created the GSMClaims dataset to rigorously evaluate arithmetic reasoning in claim verification, an area not covered by current claim verification datasets. Using the GSM8K test set (Cobbe et al., 2021), which features multi-step grade school math problems, we transformed each problem into two claim verification instances using GPT-4o. This three-step process (Figure 5) involved: 1) reformatting the problem context into a reference document; 2) generating a "positive" claim with the correct, calculated answer; and 3) creating a "negative" claim with a plausible, incorrect answer stemming from common calculation errors. This resulted in 2,634 balanced instances where successful verification hinges on the model's arithmetic calculation ability based on the document. The GPT-4o prompt template is in Appendix E. We manually inspected a subset of 100 positive and negative claims and found them to be near-perfect accurate.

6.2 ThinknCheck-Science: Specializing for Complex Claims

The challenges posed by complex scientific and quantitative claims led us to the development of ThinknCheck-Science, a specialized variant of our base verification model. This model variant aims to enhance the reasoning capabilities of the ThinknCheck-1B model through targeted domain adaptation. To accomplish this, we augmented the LLMAggreFact-Think training dataset with additional reasoning-enhanced examples from domains requiring specialized knowledge and calculation abilities. Specifically, we incorporated 614 examples from the SciFact training set, which focuses on scientific claims, and 398 examples from our newly created GSMClaims dataset, which emphasizes arithmetic reasoning. ThinknCheck-Science was subsequently fine-tuned using this enriched dataset while maintaining the same architecture and training procedure as the base ThinknCheck-1B model.

7 Experiments: Evaluating ThinknCheck

We conducted extensive experiments to assess ThinknCheck's effectiveness across multiple dimensions, comparing against competitive baselines and analyzing the impact of our architectural decisions. All improvements reported in this section were tested for statistical significance using the Friedman test, followed by the Nemenyi post-hoc test where applicable.

7.1 Evaluation Metrics

Following prior work (Tang et al., 2022; Fabbri et al., 2021; Laban et al., 2022; Tang et al., 2024), we adopt Balanced Accuracy (BAcc) as our primary metric for evaluating potentially imbalanced datasets like LLMAggreFact and SciFact. [6] For the balanced GSMClaims dataset, we report standard accuracy, so our results are interpretable with previous works.

7.2 Baselines

We compare ThinknCheck against three categories of baselines: (1) closed LLMs in zero-shot settings (GPT-4, GPT-4o, Claude-Sonnet-3.5) as reported by Tang et al. (2024). Our goal is not to compete with these private and massive foundation models, but to provide context., (2) specialized verification models (AlignScore, MiniCheck-7B), and (3) ThinknCheck variants (ThinknCheck-nothink, ThinknCheck, ThinknCheck-Science) to isolate the impact of reasoning and data augmentation components.

[6] Balanced Accuracy (BAcc) is defined as:

$$\text{BAcc} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP}+\text{FN}} + \frac{\text{TN}}{\text{TN}+\text{FP}}\right),$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

7.3 Core Verification Performance (LLMAggreFact)

Table 1 presents the performance of ThinknCheck and baselines on the LLMAggreFact benchmark.

Model	BAcc
GPT-4 (zero-shot)	75.3
GPT-4o (zero-shot)	75.9
Claude-Sonnet-3.5 (zero-shot)	77.2
AlignScore (355M/fp16)	70.4
MiniCheck (7B/fp16)	77.4
ThinknCheck-nothink (1B/fp4)	57.5
ThinknCheck (1B/fp4)	**78.1**

Table 1: Performance on LLMAggreFact test set (29.3K examples). ThinknCheck-1B surpasses the larger MiniCheck-7B, and the large gap to ThinknCheck-nothink highlights the benefit of reasoning.

Our results reveal several key findings. First, ThinknCheck-1B outperforms MiniCheck-7B despite using 7× fewer parameters and 4-bit quantization. Second, the reasoning component is critical—removing it (ThinknCheck-nothink) leads to a dramatic 20.6 point drop in performance. Third, our compact 1B model matches or exceeds state-of-the-art LLMs like GPT-4o and Claude, demonstrating that specialized, reasoning-based architectures can achieve competitive performance at a fraction of the computational cost.

7.4 Robustness and Generalization: Evaluation on SciFact

Beyond core performance, we investigated ThinknCheck's robustness to distribution shifts. Scientific claims in the SciFact benchmark often require implicit reasoning or conceptual understanding beyond simple lexical overlap (as illustrated in Figure 1). We hypothesized that ThinknCheck's explicit reasoning process would enhance out-of-domain generalization compared to models focused solely on classification.

Table 2 strongly supports this hypothesis. ThinknCheck-1B achieves 64.7 BAcc on the SciFact development set, a substantial 14.7 absolute point improvement (29.4% relative gain) over MiniCheck-7B (50.0 BAcc). The ThinknCheck-nothink ablation performs poorly (21.7 BAcc), confirming that the reasoning capability drives this enhanced generalization. These results demonstrate that ThinknCheck handles claims requiring deeper understanding more effectively, suggesting important implications for practical deployment in domain-shifting scenarios.

Model	BAcc
MiniCheck-7B	50.0
ThinknCheck-nothink-1B	21.7
ThinknCheck-1B	**64.7**

Table 2: Performance on the SciFact development set. The reasoning mechanism in ThinknCheck-1B leads to vastly superior out-of-domain generalization compared to MiniCheck-7B and the non-reasoning ablation.

7.5 Performance on Complex Reasoning: GSMClaims & ThinknCheck-Science

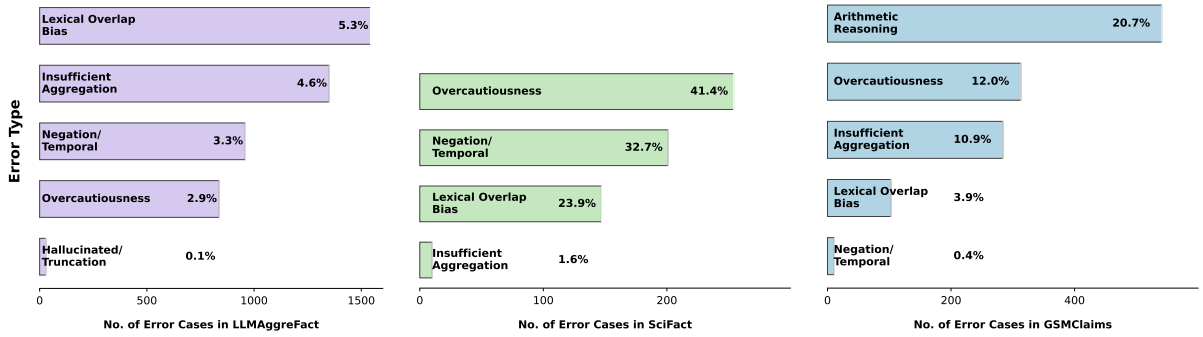
We used the GSMClaims dataset introduced in Section 6.1 to evaluate performance on claims requiring arithmetic reasoning. Table 4 shows the zero-shot performance across models. As expected, both ThinknCheck-1B (52.1% Acc) and MiniCheck-7B (51.3% Acc) find this task challenging, particularly struggling to verify correct positive claims. This confirms that standard training on text entailment is insufficient for reliable numerical reasoning.

Model	Positive	Negative	Overall
Uniform Baseline	50.0	50.0	50.0
MiniCheck-7B	14.4	88.1	51.3
ThinknCheck-nothink-1B	1.0	97.8	49.4
ThinknCheck-1B	14.6	89.8	**52.2**

Table 3: Zero-shot performance on GSMClaims (Accuracy %). Both ThinknCheck and MiniCheck struggle with arithmetic reasoning, particularly verifying positive (correct) claims.

To address this limitation, we evaluated ThinknCheck-Science, our model specifically trained with additional scientific and arithmetic data (introduced in Section 4.2). Table 4 compares its performance against other models across all benchmarks and Figure 7 shows ThinknCheck-Science in action with a highly non-trivial claim verification example requiring complex reasoning. ThinknCheck-Science achieves the best performance on GSMClaims (61.0% Acc), demonstrating that the ThinknCheck architecture can be effectively specialized through targeted training to significantly improve quantitative reasoning capabilities. Importantly, it also shows improvements on LLMAggreFact and SciFact, indicating that the specialized training enhances rather than compromises its general verification abilities.

These results demonstrate that ThinknCheck-Science offers comprehensive improvements across



(a) (b) (c)

Figure 6: Subfigures (a), (b), and (c) show the distribution of key error types on LLMAggreFact, SciFact, and GSMClaims, respectively revealed during our error analysis. For further discussion of these error types see Section 8.

Model	LLMAggre Fact	SciFact (dev)	GSM Claims
MiniCheck-7B	77.4	50.0	51.3
ThinknCheck-nothink-1B	57.5	21.7	49.4
ThinknCheck-1B	78.1	64.7	52.2
ThinknCheck-Science-1B	**79.2**	**66.4**	**61.0**

Table 4: Performance of ThinknCheck-Science across datasets. Targeted training significantly improves performance not only on arithmetic (GSMClaims) and scientific (SciFact) verification, but also on general claim verification (LLMAggreFact).

all evaluation benchmarks, with 17% relative improvement over the base model on GSMClaims.

8 Error Analysis

A comprehensive error analysis of ThinknCheck-1B on LLMAggreFact, SciFact, and GSMClaims datasets, using a unified error taxonomy (Figure 6), revealed varied error profiles. **Lexical Overlap Bias** was most prevalent in LLMaggreFact (5.3%) but lower in GSMClaims (3.9%). In GSMClaims, mathematical claims led to dominant **Arithmetic Reasoning** errors (20.7%). **Overcautiousness**, the leading error in SciFact (41.4%), reflects difficulty confirming complex, ungrounded scientific assertions. **Negation/Temporal** errors were significant in SciFact (32.7%) and LLMaggreFact (3.3%) but rare in GSMClaims (0.4%), highlighting domain-specific reasoning issues. **Insufficient Aggregation** occurred across datasets, critically in LLMaggreFact (4.6%) where multi-hop synthesis is key. These patterns show significant domain-specific error profile variations, underscoring the need for dataset-specific claim verification strategies.

Our findings show domain-specific error patterns: general-domain verification was most affected by lexical overlap bias and insufficient aggregation (Appendix I); scientific claims by a high rate of overcautious false negatives and negation errors (Appendix J); and mathematical claims by arithmetic reasoning failures and multi-step aggregation issues (Appendix K). These insights suggest targeted mitigations like adversarial data mining and domain-specific prompting. Detailed examples and further recommendations are in the respective appendices

9 Conclusion

ThinknCheck, a novel claim verification model, achieves state-of-the-art performance by explicitly generating structured reasoning chains before verification decisions. This reasoning-first paradigm offers significant, multi-dimensional advantages. ThinknCheck-1B outperforms larger models like MiniCheck-7B with fewer parameters and shows remarkable out-of-domain generalization, achieving a 29.4% relative gain on scientific claims. Comprehensive error analysis across general, scientific, and mathematical claim verification identified domain-specific challenges and informed targeted mitigation strategies. The LLMAggreFact-Think and GSMClaims datasets offer valuable resources for future reasoning-augmented verification research. Moreover, the specialized ThinknCheck-Science variant shows targeted domain adaptation yields substantial improvements without compromising general verification. Our success suggests robust AI verification may stem from architectures leveraging structured reasoning, not just model scaling. With automated verification's growing importance across diverse domains (news, science, education), ThinknCheck establishes reasoning-first approaches as a promising foundation for more accurate, resource-efficient, interpretable, and adaptable systems.

10 Limitations

ThinknCheck advances claim verification, and we identify several promising directions for future development. To enhance applicability to very large documents or multi-document scenarios, architectures supporting longer contexts could be explored (Poli et al., 2023; Waleffe et al., 2024), moving beyond the current 4558-token (6000 words) window. The fixed 512-token reasoning output, while encouraging succinctness, could be made dynamic to better handle complex claims needing extensive explanation. Furthermore, performance on tasks like GSMClaims suggests that integrating external tools (e.g., calculators) (Patil et al., 2024) is a key step for complex arithmetic reasoning. Finally, aligning with challenges in prior work (Tang et al., 2024; Zha et al., 2023), calibrating output logits to serve as reliable confidence scores (Liu et al., 2025) remains an important area for ongoing investigation and future refinement of ThinknCheck.

11 Ethical Considerations and Broader Impact

Please see Appendix L for a detailed discussion of broader impact of this work and its ethical ramifications.

12 Reproducibility Statement

We are committed to ensuring the reproducibility of our research. To this end, we provide details regarding our datasets, models, and experimental setup in Appendix M.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. 2025. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*.

Gagan Bansal, Tongshuang Sherry Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel S. Weld. 2020. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.

Hung-Ting Chen, Fangyuan Xu, Shane A Arora, and Eunsol Choi. 2023. Understanding retrieval augmentation for long-form question answering. *arXiv preprint arXiv:2310.12150*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, and many more. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Alexander R Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2021. Qafacteval: Improved qa-based factual consistency evaluation for summarization. *arXiv preprint arXiv:2112.08542*.

Mingming Fan, Xianyou Yang, Tsz Tung Yu, Vera Q. Liao, and J. Zhao. 2021. Human-ai collaboration for ux evaluation: Effects of explanation and synchronization. *ArXiv*, abs/2112.12387.

GemmaTeam, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ram é , Morgane Rivi è re, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gal Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, and many more. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yebowen Hu, Tim Ganter, Hanieh Deilamsalehy, Franck Dernoncourt, Hassan Foroosh, and Fei Liu. 2023. Meetingbank: A benchmark dataset for meeting summarization. *arXiv preprint arXiv:2305.17529*.

Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roee Aharoni, and Mor Geva. 2024a. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains. *Preprint*, arXiv:2402.00559.

Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roee Aharoni, and Mor Geva. 2024b. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains. *arXiv preprint arXiv:2402.00559*.

Misbah Javaid and Vladimir Estivill-Castro. 2021. Explanations from a robotic partner build trust on the

robot's decisions for collaborative human-humanoid interaction. *Robotics*, 10:51.

Ryo Kamoi, Tanya Goyal, Juan Diego Rodriguez, and Greg Durrett. 2023. Wice: Real-world entailment for claims in wikipedia. *arXiv preprint arXiv:2303.01432*.

Philippe Laban, Tobias Schnabel, Paul N Bennett, and Marti A Hearst. 2022. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Nelson F Liu, Tianyi Zhang, and Percy Liang. 2023. Evaluating verifiability in generative search engines. *arXiv preprint arXiv:2304.09848*.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025. Uncertainty quantification and confidence calibration in large language models: A survey. *arXiv preprint arXiv:2503.15850*.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2023. Expertqa: Expert-curated questions and attributed answers. *arXiv preprint arXiv:2309.07852*.

Chaitanya Malaviya, Subin Lee, Sihao Chen, Elizabeth Sieber, Mark Yatskar, and Dan Roth. 2024. ExpertQA: Expert-curated questions and attributed answers. In *2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2024. Gorilla: Large language model connected with massive apis. *Advances in Neural Information Processing Systems*, 37:126544–126565.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. 2023. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pages 28043–28078. PMLR.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Liyan Tang, Tanya Goyal, Alexander R Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryściński, Justin F Rousseau, and Greg Durrett. 2022. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. *arXiv preprint arXiv:2205.12854*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. *Preprint*, arXiv:2404.10774.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

Roger Waleffe, Wonmin Byeon, Duncan Riach, Brandon Norick, Vijay Korthikanti, Tri Dao, Albert Gu, Ali Hatamizadeh, Sudhakar Singh, Deepak Narayanan, et al. 2024. An empirical study of mamba-based language models. *arXiv preprint arXiv:2406.07887*.

Yuxia Wang, Revanth Gangi Reddy, Zain Muhammad Mujahid, Arnav Arora, Aleksandr Rubashevskii, Jiahui Geng, Osama Mohammed Afzal, Liangming Pan, Nadav Borenstein, Aditya Pillai, et al. 2023. Factcheck-bench: Fine-grained evaluation benchmark for automatic fact-checkers. *arXiv preprint arXiv:2311.09000*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. *Preprint*, arXiv:2305.16739.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A Smith. 2023. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. Mediasum: A large-scale media interview dataset for dialogue summarization. *arXiv preprint arXiv:2103.06410*.

A Prompt for generating LLMAggreFact-Think

You are expert fact checker with a strong attention to detail and access to a wealth of information. Given a document and a claim, determine if the claim is entailed by the document, only using the facts in the document.
Respond in the following format:

```
<reasoning>
... // clear, but short description of your
    step by step
   // thinking to arrive at the entailment
   // keep the reasoning sentences separated
        by a newline.
</reasoning>
<entailment>
... // This is always a single word, either
    "YES" or "NO"
</entailment>
```

B ThinknCheck finetuning: hyperparameter details

For fine-tuning, we used LoRA (Hu et al., 2021) with rank=64, lora_alpha=64, and a learning rate of 2e-4 scheduled linearly. We updated the query, key, value, and output projection layers, as well as MLP gate, up, and down projections. The fine-tuning was performed on an A100 GPU for 1 epoch, with 5 warmup steps, a batch size of 4 with 4 accumulated steps, and an 8bit-AdamW optimizer with a weight decay of 0.01.

C Finetuning Prompt for ThinknCheck

You are given a document and a claim. The document is enclosed between <DOCUMENT> and </DOCUMENT>. The claim is between <CLAIM> and </CLAIM>. Determine if the claim is entailed by the document. Think about the problem and provide your reasoning. Place the reasoning between <REASONING> and </REASONING>. Then, provide your entailment solution between <SOLUTION> and </SOLUTION>. The entailment should be either a YES or a NO.

```
<DOCUMENT>
{document}
</DOCUMENT>
<CLAIM>
{claim}
</CLAIM>
<REASONING>
{reasoning}
</REASONING>
<SOLUTION>
{solution}
</SOLUTION>
```

D Finetuning Prompt for ThinknCheck-nothink

You are given a document and a claim. The document is enclosed between <DOCUMENT> and </DOCUMENT>. The claim is between <CLAIM> and </CLAIM>. Determine if the claim is entailed by the document. Provide your entailment solution between <SOLUTION> and </SOLUTION>. The entailment should be either a YES or a NO.

```
<DOCUMENT>
{document}
</DOCUMENT>
<CLAIM>
{claim}
</CLAIM>
<SOLUTION>
{solution}
</SOLUTION>
```

E Prompt to generate GSMClaims

Given an arithmetic problem and a solution, rewrite them as a document and a a pair of positive and negative claims such the positive claim is entailed by the document (after solving some arithmetic) and the negative claim is not entailed by the document (after solving some arithmetic). Produce your answer only as a JSON. Do not add anything before and after the JSON.

F Examples of Reasoning Patterns in LLMAggreFact-Think

Direct Evidence Extraction & Matching:

> ***Claim:*** "Proper clean-up prior to insertion of the vaginoscope is essential."
> ***Reasoning:*** "The document contains a sentence that explicitly states 'Proper clean-up prior to insertion of the vaginoscope is essential.' This directly matches the claim made..."

Absence of Evidence Identification:

> ***Claim:*** "A star's temperature is determined by the amount of mass and energy it has."
> ***Reasoning:*** "The document discusses determining the effective temperature of a star from its spectrum but does not mention the relationship between a star's mass, energy, and temperature... Therefore, the claim is not backed by the facts in the document..."

Synthesis of Multiple Information Points:

> ***Claim:*** "However, according to customer reviews, the quality of the food served at the restaurant is subpar."
> ***Reasoning:*** "The document contains multiple customer reviews expressing dissatisfaction... One review explicitly states that the food was 'horrible,'... Another review criticizes the broth as 'least flavorful'... Together, these reviews confirm that the quality of food... is indeed regarded as subpar by customers."

Addressing Scope and Specificity Mismatches:

> ***Claim:*** "Thermodynamics provides the fundamental laws of energy conservation, mass conservation, and chemical equilibrium that govern the behavior of chemical systems."
> ***Reasoning:*** "The document discusses the first law of thermodynamics, which addresses energy conservation, aligning with part of the claim. However, it does not support the components of mass conservation and chemical equilibrium... Therefore, the claim goes beyond what is explicitly stated... and is only partially supported."

Handling Nuance and Implication:

> ***Claim:*** "The restaurant's use of fresh and high-quality ingredients, such as shrimp and pork belly, has also been commended by reviewers."
> ***Reasoning:*** "...One review even states that 'the shrimp you could really sink your teeth into,' suggesting that the shrimp is fresh and of high quality. The combination of positive comments about these specific ingredients supports the claim."

Step-by-Step Verification:

> ***Claim:*** "Return beef and any accumulated juices to the pan, along with tomato sauce, liver spread, and bay leaves."
> ***Reasoning:*** "...passage 3 explicitly states to 'Return beef and accumulated juices to pan' and also adds 'Add water, tomato sauce, liver spread and bay leaf.' Therefore, all elements of the claim are present in the passage..."

G Non-trivial Example of Claim Verification with ThinknCheck-Science

Claim: Momentum P $\approx$ 10.0 $\pm$1.1 kg $\cdot$ m/s
Document: Mass M = 5.0 $\pm$0.1 kg. Velocity V = 2.0 $\pm$0.2 m/s.

MiniCheck-7B verification result:
`DOES NOT SUPPORT`

ThinknCheck-Science verification result:
`SUPPORTS`

ThinknCheck-Science reasoning:

The momentum is calculated by multiplying the mass and the velocity: 5.0 kg $\times$2.0 m/s = 10.0 kg $\cdot$ m/s. The relative uncertainty in the mass is 0.1/5.0 = 2% and in the velocity is 0.2/2.0 = 10%. When uncertainties are combined for multiplication, they add in quadrature: $\checkmark$((2%)^2 + (10%)^2) = $\checkmark$(0.0004 + 0.01) = $\checkmark$(0.0104) $\approx$ 10.2%. This results in an uncertainty of about 10.2% $\times$10.0 kg $\cdot$ m/s $\approx$ 1.0 kg $\cdot$ m/s. A better calculation using maximum and minimum values yields P_{max} = 5.1 kg $\times$2.2 m/s = 11.22 kg $\cdot$ m/s and P_{min} = 4.9 kg $\times$1.8 m/s = 8.82 kg $\cdot$ m/s. The range between P_{max} and P_{min} is 11.22 8.82 = 2.4 kg $\cdot$ m/s, implying an uncertainty of approximately $\pm$1.2 kg $\cdot$ m/s. Since both approaches yield an uncertainty around $\pm$1.0 to $\pm$1.2 kg $\cdot$ m/s, the claim of 10.0 $\pm$1.1 kg $\cdot$ m/s is consistent with the uncertainties derived from the document.

Figure 7: An example of non-trivial reasoning in ThinknCheck-Science that involves thinking through quantitative uncertainty.

H Distribution of the Reasoning Methods across datasets

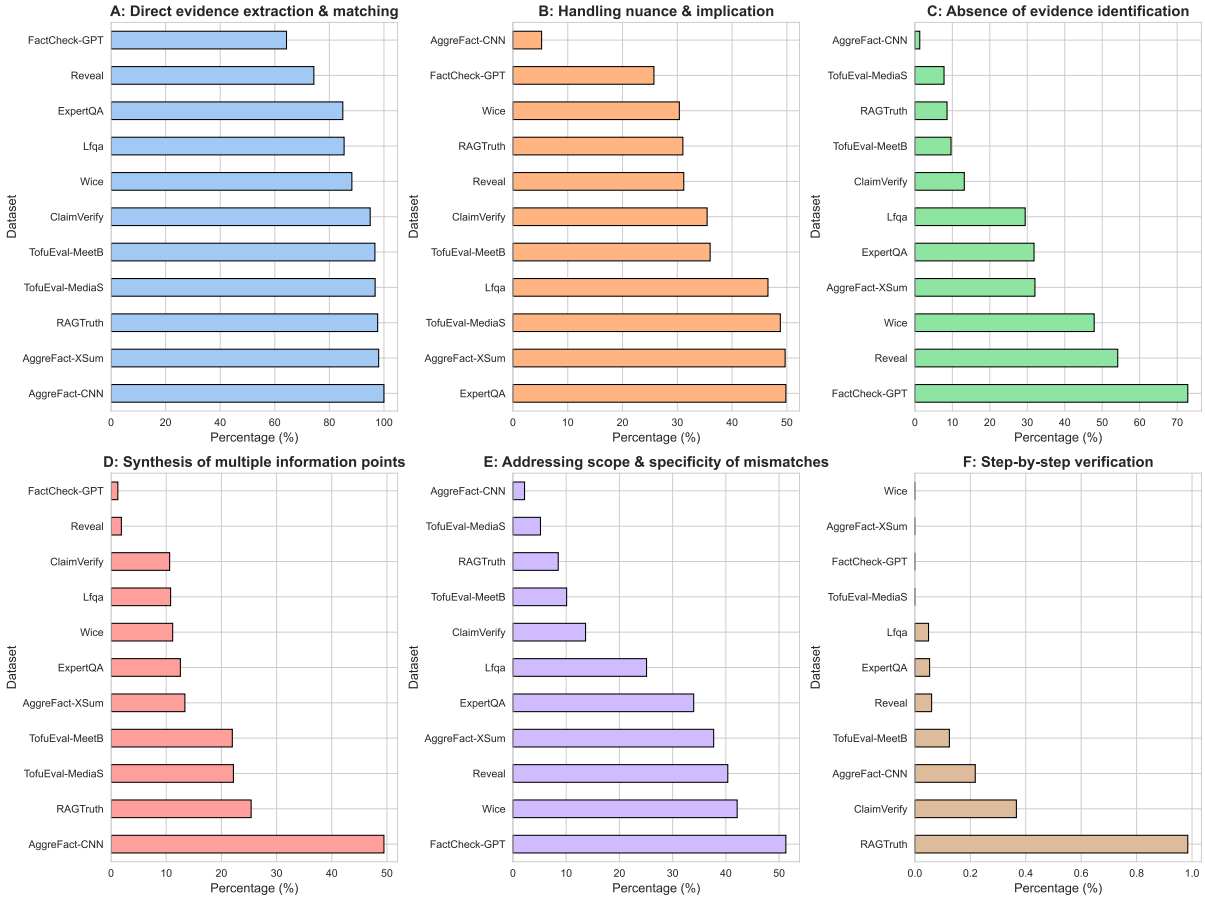


Figure 8: Comparative distribution of reasoning strategies employed for claim verification across various source datasets within LLMAggreFact-Think. Each subplot (A-F) corresponds to a distinct reasoning method identified in Section 5: (A) Direct evidence extraction & matching, (B) Handling nuance & implication, (C) Absence of evidence identification, (D) Synthesis of multiple information points, (E) Addressing scope & specificity of mismatches, and (F) Step-by-step verification. The figure highlights how different datasets predominantly elicit different reasoning patterns.

I Comprehensive Error Analysis and Mitigation Strategies: LLMAggreFact

I.1 Error Taxonomy and Case Counts

Through systematic analysis of model predictions, we identified five distinct error categories that capture the primary failure modes in claim verification:

Lexical Overlap Bias occurs when the model incorrectly predicts YES based on surface-level lexical similarity between the claim and context, without proper semantic entailment assessment. **Insufficient Aggregation** manifests as the model's failure to synthesize information distributed across multiple sentences or paragraphs—a critical requirement for complex, multi-hop claims. **Negation/Temporal Confusion** involves mishandling negations or temporal relationships, often resulting in incorrect entailment decisions. **Overcautiousness** is observed when the model requires complete and explicit evidence for all components of a composite claim, defaulting to NO even when most sub-claims are well-supported. Finally, **Hallucinated Justification** errors arise when the model generates confident reasoning unsupported by the document, particularly prevalent when input is truncated.

Our quantitative analysis of the development set reveals the following distribution of these error types:

Error Type	Case Count
Lexical Overlap Bias	1,543
Insufficient Aggregation	1,350
Negation/Temporal	959
Overcautiousness	837
Hallucinated/Truncation	29

Table 5: Counts of each error type in the analysis set.

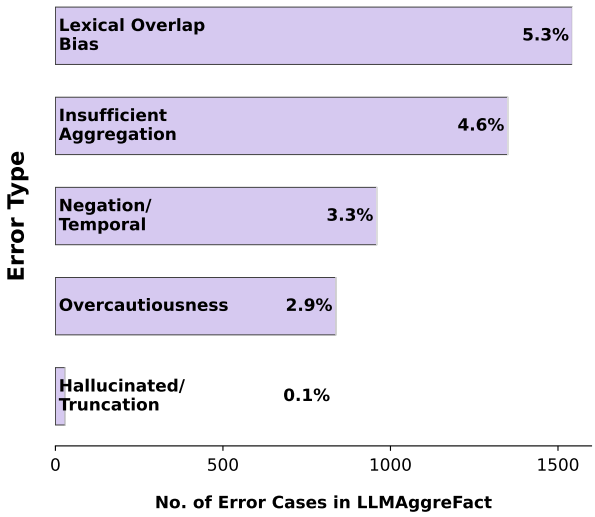


Figure 9: This figure shows the percentage of errors in the development set attributed to each major error type identified during our analysis of the ThinknCheck claim verification model. The most prevalent error is Lexical Overlap Bias (5.3%), where the model incorrectly predicts support for a claim based primarily on surface-level phrase overlap between the claim and document, rather than true entailment. Insufficient Aggregation (4.6%) represents failures to synthesize evidence across multiple sentences or paragraphs—often required for complex, multi-hop claims. Negation/Temporal errors (3.3%) arise when the model fails to correctly handle negation or temporal relationships, frequently confusing past, future, or negated statements. Overcautiousness (2.9%) is observed when the model predicts "NO" unless every aspect of a composite claim is explicitly supported, leading to false negatives even when most sub-claims are correct. The least frequent category, Hallucinated/Truncation (0.1%), captures instances where the model generates unsupported or speculative justifications, typically due to truncated input context. These findings highlight key areas for targeted mitigation and future improvement in LLM-based claim verification systems.

I.2 Representative Error Snippets

We present characteristic examples of each error type, illustrating the specific patterns and reasoning failures observed:

Lexical Overlap Bias:

> **Claim:** Roberto Martinez felt Seamus Coleman should have been awarded a free-kick before the defender conceded the penalty that allowed Swansea to pinch a 1-1 draw at the Liberty Stadium.
> **Ground Truth:** NO
> **Predicted:** YES
> **Analysis:** The model matches surface phrases without verifying true entailment.

Negation/Temporal Confusion:

> **Claim:** Lazio beat Napoli 1-0 on Wednesday to reach the Coppa Italia final...
> **Ground Truth:** YES
> **Predicted:** NO
> **Analysis:** The model confuses event chronology, failing to parse past vs. fu-

ture events.

Hallucinated Justification (Truncation):

> **Claim:** [Claim regarding match details, with evidence cut off by truncation]
> **Truncated:** True
> **Predicted:** YES
> **Analysis:** The model fills in missing information with confident, unsupported rationale.

Insufficient Aggregation (Multi-hop):

> **Claim:** Leicester City are just three points from safety... have won back-to-back games against Arsenal and West Brom...
> **Ground Truth:** YES
> **Predicted:** NO
> **Analysis:** Model fails to aggregate evidence across sentences.

Overcautiousness:

> **Claim:** Maxime Machenaud crossed for Racing Metro 92 in the first half. Charlie Hodgson kicked two penalties... Marcelo Bosch won the match with a last-minute penalty.
> **Ground Truth:** YES
> **Predicted:** NO
> **Analysis:** Model returns NO unless every part is perfectly supported, even if most are.

I.3 Mitigation Strategies

Based on our error analysis, we propose targeted mitigation approaches for each error category:

For **Lexical Overlap Bias**, we recommend incorporating adversarial examples with high lexical overlap but contradictory semantics, multi-task training with established NLI datasets, and explicit prompting for evidence-based reasoning that requires justification of entailment decisions.

To address **Insufficient Aggregation**, enriching training data with multi-hop claims is essential, alongside chain-of-thought prompting or structured claim decomposition techniques. Having the model explicitly highlight or enumerate relevant evidence sentences across the document can also enhance multi-hop reasoning capabilities.

For **Negation/Temporal Confusion**, augmenting training with carefully constructed examples highlighting negation and temporal relationships is critical. Explicit instructions to attend to these linguistic cues, potentially combined with integration of specialized temporal parsers, could significantly improve performance.

Overcautiousness might be mitigated by introducing finer-grained labeling schemes such as "PARTIAL" or "INSUFFICIENT" support, requiring the model to verify each claim component separately, and calibrating the model to avoid overuse of NO predictions for partially supported claims.

Finally, **Hallucinated Justification** errors can be addressed by training with truncated documents explicitly labeled as providing "Insufficient Information," prompting the model to recognize and flag missing evidence, and implementing confidence calibration techniques specific to incomplete inputs.

Our findings underscore that claim verification remains challenging for LLMs, particularly in contexts involving high lexical overlap, multi-hop reasoning requirements, or truncated evidence. We contend that structured data augmentation, adversarial example mining, and carefully designed prompting strategies are essential to addressing these challenging cases and advancing the state of LLM-based claim verification.

J Comprehensive Error Analysis and Mitigation Strategies: SciFact

J.1 Error Taxonomy and Distribution

Our analysis of model errors on the SciFact development set identified four primary error categories, with distinct distributions from the general claim verification dataset:

Overcautiousness dominates the scientific claim verification errors, manifesting as the model predicting NO unless every component of a claim is directly and explicitly supported, even when the majority of elements are correct. **Negation/Temporal** errors involve misinterpretation of negations, risk factors, associations, or causal and temporal relationships—particularly problematic in scientific contexts where precise interpretation of these elements is critical. **Lexical Overlap Bias** occurs when the model incorrectly predicts YES based on surface-level terminology matches without proper scientific entailment. **Insufficient Aggregation** errors, while less frequent in SciFact compared to general news verification, still occur when the model fails to synthesize distributed evidence for

complex scientific claims.

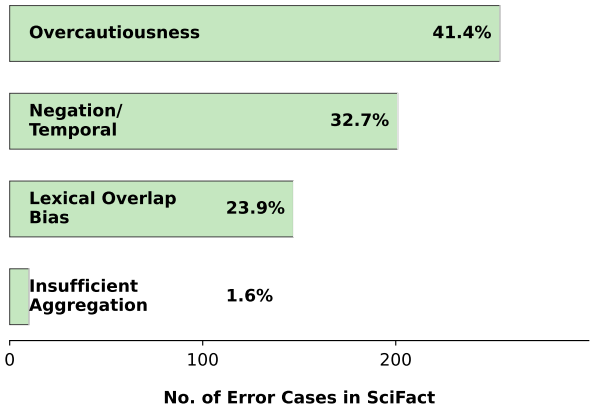


Figure 10: This bar chart shows the percentage of errors attributable to each major error type. The most frequent is Overcautiousness (41.4%), where the model predicts "NO" unless every component of a scientific claim is directly supported by the abstract. Negation/Temporal errors (32.8%) arise from failure to correctly process negations or temporal/causal relations, often leading to mistakes about risk, association, or event directionality. Lexical Overlap Bias (24.1%) refers to errors where the model incorrectly predicts "YES" based on surface word overlap, without true scientific entailment. Insufficient Aggregation (1.7%) captures failures in synthesizing multi-hop or compositional evidence, a rare but notable error on SciFact. This distribution highlights the unique challenges of scientific claim verification for LLMs.

J.2 Representative Error Examples

The following examples illustrate characteristic instances of each error type on scientific claims:

Overcautiousness:

> **Claim:** 1,000 genomes project enables mapping of genetic sequence variation consisting of rare variants with larger penetrance effects than common variants.
> **Ground Truth:** YES
> **Predicted:** NO
> **Analysis:** The document discusses the identification of common variants and the implications of synthetic associations arising from rare variants... but does not provide any information about the specific number of genomes being mapped...

Negation/Temporal:

> **Claim:** APOE4 expression in iPSC-derived neurons increases AlphaBeta production and tau phosphorylation causing GABA neuron degeneration.
> **Ground Truth:** YES
> **Predicted:** NO
> **Analysis:** ...ApoE4 resulted in higher levels of tau phosphorylation and increased A β production, but these effects were not observed in mouse neurons...

Lexical Overlap Bias:

> **Claim:** ALDH1 expression is associated with better breast cancer outcomes.
> **Ground Truth:** NO
> **Predicted:** YES
> **Analysis:** The document states that ALDH1 expression is correlated with poor prognosis in breast cancers... which is directly supported by the information provided in the document.

Insufficient Aggregation:

> **Claim:** The YAP1 and TEAD complex translocates into the nucleus where it interacts with transcription factors and DNA-binding proteins that modulate target gene transcription.
> **Ground Truth:** YES
> **Predicted:** NO
> **Analysis:** ...discusses the regulation of the Hippo pathway and mentions YAP/-TAZ are co-activators that interact with TEAD, but does not explicitly state that YAP1 and TEAD complex translocates and interacts as claimed...

J.3 Mitigation Strategies

To address the specific challenges of scientific claim verification, we propose several targeted mitigation strategies:

For **Overcautiousness**, implementing more nuanced verification labels such as "PARTIAL" or "INSUFFICIENT" support could capture the varying degrees of evidence often found in scientific literature. Additionally, prompting the model to explicitly assess each component of a complex scientific claim independently may mitigate the tendency toward global rejection of partially supported claims.

To address **Negation/Temporal** errors, which are particularly problematic in scientific contexts,

targeted data augmentation with examples emphasizing negation, risk factors, and causal relationships is essential. Specialized instruction to attend to these linguistic features, possibly combined with domain-specific pre-training on scientific literature with these relations, could improve performance.

For **Lexical Overlap Bias**, incorporating challenging negative examples with high domain-specific terminology overlap but contradictory semantics is crucial. Training the model to articulate precise scientific evidence for its predictions, rather than relying on terminology matches, may reduce this error type.

Finally, while less frequent, **Insufficient Aggregation** errors could be addressed by including more complex multi-hop scientific claims in training and employing claim decomposition or structured chain-of-thought prompting to facilitate evidence synthesis across abstracts.

These domain-specific approaches, tailored to the unique challenges of scientific text, are essential for improving the robustness of LLM-based scientific claim verification systems.

K GSMClaims Error Analysis and Mitigation

K.1 Error Taxonomy and Distribution

Our analysis of mathematical claim verification errors revealed distinct patterns requiring tailored mitigation strategies. We extended our general error taxonomy with an additional category specific to mathematical reasoning:

Arithmetic Reasoning errors occur when the model fails to execute the correct computation or misapplies a mathematical operation, despite correctly understanding the problem structure. **Overcautiousness** manifests as the model predicting NO unless every calculation step is made fully explicit, even when the reasoning is largely correct. **Insufficient Aggregation** errors arise when the model struggles to combine multiple steps or facts required for a multi-hop calculation. **Lexical Overlap Bias** involves incorrectly predicting YES due to matching numbers or mathematical terms, rather than validating computational correctness. **Negation/Temporal** errors, while rare in this domain, occur when the model mishandles negation or temporal aspects of mathematical problems.

Error Type	Percentage (%)
Arithmetic Reasoning	43.2
Overcautiousness	25.0
Insufficient Aggregation	22.7
Lexical Overlap Bias	8.2
Negation/Temporal	0.9

Table 6: Distribution of error types in the GSMClaims test set.

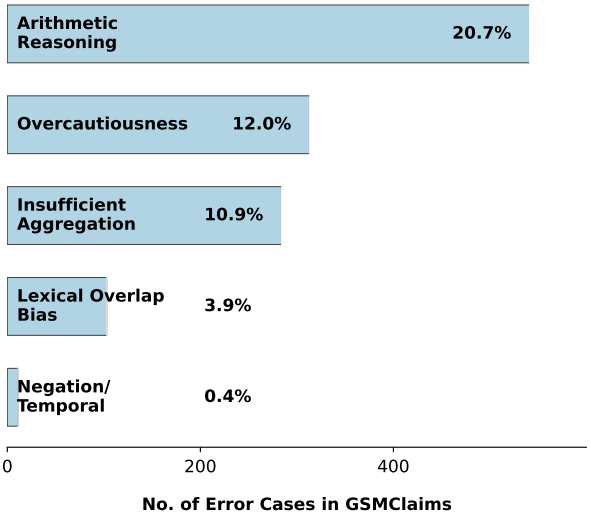


Figure 11: This histogram shows the percentage of errors attributed to each major error type. The most common are Arithmetic Reasoning errors (43.2%), where the model fails to perform the correct computation or makes mistakes in multi-step math. Overcautiousness (25.0%) and Insufficient Aggregation (22.7%) indicate, respectively, that the model tends to say "NO" unless every calculation is explicit, or struggles to combine multiple facts or operations across the scenario. Lexical Overlap Bias (8.2%) arises when the model is fooled by matching numbers or phrases even if the math is wrong. Negation/Temporal errors are rare (0.9%), reflecting occasional mishandling of negatives or time-dependent claims. This distribution highlights the special challenges of claim verification in math word problems, especially around robust arithmetic reasoning.

K.2 Representative Error Examples

The following examples illustrate characteristic instances of each error type in mathematical claim verification:

Arithmetic Reasoning:

> **Claim:** Janet makes $18 every day at the farmers' market.
> **Ground Truth:** YES
> **Predicted:** NO

Analysis: The document states that Janet sells the remainder of her eggs at the farmers' market for $2 per egg... To determine the total earnings, we need to calculate the total revenue... [model fails in calculation].

Overcautiousness:

Claim: Josh made a profit of $70,000.
Ground Truth: YES
Predicted: NO
Analysis: The document states that Josh buys a house for $80,000 and invests $50,000 in repairs, resulting in an increase in value by 150%... Therefore, the claim... is not directly supported...

Insufficient Aggregation:

Claim: The robe takes 2 bolts of fabric in total.
Ground Truth: NO
Predicted: YES
Analysis: The document states a robe takes 2 bolts of blue fiber and half that much white fiber... total is 2 bolts + 1 bolt, not 2...

Lexical Overlap Bias:

Claim: Cynthia will spend $24.00 on ice cream after 60 days.
Ground Truth: NO
Predicted: YES
Analysis: ... calculates surface numbers but matches claim even if the computation is wrong.

Negation/Temporal:

Claim: 120 pounds of carrots will not be used.
Ground Truth: YES
Predicted: NO
Analysis:‘ ... Model misreads the negation, misinterpreting the distribution and what's left unused.

K.3 Mitigation Strategies

The unique challenges of mathematical claim verification require domain-specific mitigation strategies:

For **Arithmetic Reasoning** errors, which constitute the largest error category, we propose augmenting training with intermediate calculation supervision to improve computational accuracy. Prompting for step-by-step computation with explicit requirements to output and verify intermediate values could substantially improve performance. Integration with symbolic calculators or specialized numerical modules may also enhance computation reliability.

To address **Overcautiousness**, implementing partial credit mechanisms for nearly correct answers and encouraging model self-correction through additional verification steps could be beneficial. Training on intentionally ambiguous or underspecified mathematical claims may also reduce the tendency toward rejecting claims that require implicit calculation steps.

For **Insufficient Aggregation** errors, incorporating more multi-hop mathematical problems in training data is essential. Enforcing structured multi-step solution explanations with clearly delineated sub-problems could enhance the model's ability to integrate distributed mathematical information.

Lexical Overlap Bias could be mitigated through adversarial training with examples containing similar numerical values but different computational pathways and outcomes. Requiring explicit calculation steps rather than allowing the model to match surface-level numbers would reduce these errors.

Finally, for the rare **Negation/Temporal** errors, targeted data augmentation focusing on problems involving mathematical negation and time-dependent calculations could improve performance in these edge cases.

These findings highlight the distinct challenges of mathematical claim verification in language models, particularly around arithmetic reasoning and computational accuracy. While sharing some error categories with general claim verification, mathematical verification requires specialized approaches focusing on computational precision and structured reasoning.

L Ethical Considerations and Broader Impact: Details

The development of automated claim verification systems like ThinknCheck has significant potential for positive societal impact, but also demands careful consideration of ethical challenges and potential

risks.

L.1 Broader Impact and Potential Benefits

- **Combating Misinformation:** By providing tools to assess the factual grounding of claims against provided evidence, ThinknCheck aims to contribute to efforts to identify and mitigate the spread of misinformation. The explicit reasoning component is designed to offer transparency, which can be crucial in understanding why a claim is deemed supported or unsupported.

- **Enhancing Transparency and Trust in AI:** The generation of structured reasoning alongside verification decisions is a step towards more interpretable AI. This can foster greater understanding and appropriate trust from users, moving beyond "black box" systems.

- **Democratizing Fact-Checking Tools:** Our work demonstrates the feasibility of building state-of-the-art reasoning and verification models that are considerably smaller (1B parameters) than many leading systems. This improved efficiency can make such tools more accessible for deployment on edge devices or by organizations with limited computational resources, broadening their availability.

- **Supporting Specialized Domains:** Variants like ThinknCheck-Science, with improved capabilities in scientific and arithmetic claim verification, can be valuable in academic, research, or educational settings for verifying complex information.

- **Fostering Research:** The introduction of reasoning-augmented datasets like LLMAggreFact-Think and the GSMClaims benchmark for arithmetic reasoning are intended to spur further research into more robust, explainable, and nuanced claim verification systems. Our commitment to releasing datasets and models openly supports this goal.

L.2 Potential Risks and Ethical Challenges

- **Perpetuation of Biases:** ThinknCheck models are trained on existing benchmarks (LLMAggreFact, SciFact) and reasoning traces generated by large language models (GPT-4o-mini for LLMAggreFact-Think). These underlying data sources may contain societal biases (e.g., related to viewpoints, topics, or demographic representation) which could be learned and propagated by our models. While ThinknCheck aims for factual grounding based on provided text, the interpretation and reasoning patterns learned could inadvertently reflect these biases.

- **Over-reliance and Misplaced Trust:** While explanations are intended to improve transparency, there is a risk that users might overly rely on the system's output, especially if the generated reasoning appears plausible but is subtly flawed. A "DOES NOT SUPPORT" label, for instance, might be misinterpreted as active refutation rather than an absence of evidence in the provided document. The quality and true faithfulness of explanations remain an ongoing research challenge.

- **Dual Use and Adversarial Attacks:** Sophisticated claim verification tools, including those that generate reasoning, could potentially be exploited by malicious actors. For instance, they might be used to understand how to craft more convincing misinformation that can evade detection, or to generate plausible-sounding but false justifications.

- **Errors and Their Consequences:** Despite outperforming baselines, ThinknCheck is not infallible. False negatives (correct claims marked as unsupported) could lead to the dismissal of valid information, while false positives (incorrect claims marked as supported, particularly if backed by flawed reasoning) could contribute to the spread of inaccuracies. The impact of such errors can vary depending on the application domain.

- **Scope of Verification:** It is crucial to recognize that ThinknCheck verifies claims only against the provided document(s). It does not perform open-world fact-checking against comprehensive world knowledge unless that knowledge is present in the input text. This limitation must be clearly communicated to users to prevent misinterpretation of its capabilities.

We are committed to responsible AI development. The open release of our models and datasets is intended to facilitate scrutiny, further research

into robustness and fairness, and the development of better evaluation methodologies. Future work should include dedicated audits for bias in both the datasets and model outputs. We also advocate for the use of systems like ThinknCheck as tools to assist human experts and critical thinking, rather than as infallible arbiters of truth. Further research into improving the faithfulness and comprehensibility of the generated reasoning, and educating users on the capabilities and limitations of such systems, will be essential for their responsible deployment.

M Reproducibility Statement: Details

M.1 Datasets

- Core Benchmarks: Our experiments utilize several publicly available benchmarks:
  - ⋆ LLMAggreFact (Tang et al., 2024), which aggregates 9 existing datasets detailed in their work.
  - ⋆ SciFact (Wadden et al., 2020) for scientific claim verification.
  - ⋆ GSM8K (Cobbe et al., 2021) was used as the source for our new GSMClaims dataset.
- Newly Created Datasets: We introduce two new datasets:
  - ⋆ LLMAggreFact-Think: This dataset was created by augmenting the 30.4K examples in the LLMAggreFact development set with reasoning chains generated by GPT-4o-mini. After filtering for label consistency, this resulted in 24.1K examples. The prompt used for generating these reasoning traces is provided in Appendix A.
  - ⋆ GSMClaims: This dataset, comprising 2,634 balanced claim verification instances requiring arithmetic reasoning, was generated from the GSM8K test set using GPT-4o to reformat problems and create positive/negative claims. The generation prompt is detailed in Appendix D.
- Availability: As stated in our contributions, LLMAggreFact-Think and GSMClaims will be released openly under an Apache 2.0 license.

M.2 Models

- ThinknCheck Models:
  - ⋆ Our primary model, ThinknCheck-1B, is a Gemma3 1B parameter model fine-tuned on LLMAggreFact-Think.
  - ⋆ The ablation model, ThinknCheck-nothink-1B, uses the same Gemma3 1B architecture and data but is trained without the reasoning generation step.
  - ⋆ ThinknCheck-Science-1B is the ThinknCheck-1B model further fine-tuned on a combination of LLMAggreFact-Think, SciFact, and GSMClaims data.
- Baselines: We compare against several models:
  - – AlignScore (Zha et al., 2023).
  - – MiniCheck-7B (Tang et al., 2024), which is available at https://huggingface.co/bespokelabs/Bespoke-MiniCheck-7B.
- Availability: Our developed models (ThinknCheck-1B, ThinknCheck-nothink-1B, and ThinknCheck-Science-1B) and training code will be released openly under an Apache 2.0 license. We also note a relevant Hugging Face resource provided in the context of our problem formulation: `URL withheld for blind review`.

N AI Writing/Coding Assistance Disclosure

In accordance with the ACL Policy on AI Writing Assistance[7], the authors attest that we used generative AI tools for assistance purely with the language of the paper, including spell checking, grammar fixes, and proof reading. Additionally, we used GPT-4o to fix LaTeX issues, and to generate LaTeX tables from spreadsheets. In all such uses, the outputs were verified by the first author for correctness.

[7] https://www.aclweb.org/adminwiki/index.php/ACL_Policy_on_Publication_Ethics#Guidelines_for_Generative_Assistance_in_Authorship