

MisSynth: Improving MISSCI Logical Fallacies Classification with Synthetic Data

Mykhailo Poliakov and Nadiya Shvai

National University of Kyiv-Mohyla Academy, Kyiv, Ukraine.

Contributing authors: mykhailo.poliakov@ukma.edu.ua;
n.shvay@ukma.edu.ua;

Abstract

Health-related misinformation is very prevalent and potentially harmful. It is difficult to identify, especially when claims distort or misinterpret scientific findings. We investigate the impact of synthetic data generation and lightweight fine-tuning techniques on the ability of large language models (LLMs) to recognize fallacious arguments using the *MISSCI* dataset and framework. In this work, we propose *MisSynth*, a pipeline that applies retrieval-augmented generation (RAG) to produce synthetic fallacy samples, which are then used to fine-tune an LLM model. Our results show substantial accuracy gains with fine-tuned models compared to vanilla baselines. For instance, the LLaMA 3.1 8B fine-tuned model achieved an over 35% F1-score absolute improvement on the *MISSCI* test split over its vanilla baseline. We demonstrate that introducing synthetic fallacy data to augment limited annotated resources can significantly enhance zero-shot LLM classification performance on real-world scientific misinformation tasks, even with limited computational resources. The code and synthetic dataset are available on [GitHub](#).

Keywords: health misinformation, large language models, synthetic data generation, logical fallacy classification, parameter-efficient fine-tuning, retrieval-augmented generation

1 Introduction

Health-related misinformation has been identified as one of the major factors that deteriorate global health and lead to a decrease in public trust in science ([Brennen](#)

et al. 2020). This threat is growing because all forms of falsehood are spreading farther and faster than truth online (Vosoughi et al. 2018). The problem is especially dangerous when real scientific findings are distorted. For instance, misleading reports often use selective and deceptive quotations of scientific work to support false claims (Beers et al. 2023). On the other hand, discredited and retracted research continues to be mentioned as valid, supporting arguments with empty research (Frederick 2023). These arguments use the credibility of the source to hide subtle logical fallacies.

Detecting such fallacies is a major challenge. It requires a deep understanding of scientific context and logical reasoning. Often, the flawed thinking shortcuts that make readers susceptible to fallacies are more intuitive than the deliberate analysis required to debunk them (Lewandowsky et al. 2020). Even the largest large language models (LLMs) can perform poorly on this task. One recent benchmark highlights this performance gap by testing for implicit fallacious reasoning (Glockner et al. 2024). Other new datasets tools also show that LLMs lag far behind humans in identifying fine-grained fallacies (Hong et al. 2024). A comprehensive benchmark that unifies previous datasets further confirms these limitations (Helwe et al. 2024). This performance gap can be attributed to the scarcity of large, high-quality annotated datasets for training.

We find that current methods are often insufficient. Traditional fact-checking systems are designed to find explicit counter-evidence (Nakov et al. 2021). Such systems are not suited for complex cases where evidence is slightly distorted rather than outright fabricated (Guo et al. 2022). Synthetic data can help address data scarcity (Møller et al. 2024). However, synthetic data often produces templated and unnatural examples. This creates a critical distribution gap that risks training models to excel at detecting AI-generated misinformation while leaving them vulnerable to the diverse and unpredictable real-world misinformation (Li et al. 2024).

We introduce *MisSynth*, a new pipeline aiming to address this issue. Our novel technique employs retrieval-augmented generation (RAG) to produce realistic and context-sensitive synthetic data (Lewis et al. 2020). We use this data to fine-tune an LLM with a parameter-efficient technique called Low-Rank Adaptation (LoRA) by Hu et al. (2021). Our experiments show this approach yields significant gains. For example, a fine-tuned LLaMA 3.1 8B model improved its F1-score by over 35% (absolute gain) on the *MISSCI* test split. This demonstrates the effectiveness of our method, even with limited computational resources. Our primary contributions are as follows:

- We present *MisSynth*, a novel RAG-based pipeline for generating high-quality synthetic data of logical fallacies.
- We show that fine-tuning with our synthetic data significantly improves an LLM’s performance on the logical fallacy classification subtask of the *MISSCI* benchmark.
- We release the synthetic dataset generated by *MisSynth* (GPT-5 version) publicly.

The main novelty is the integration of RAG with parameter-efficient fine-tuning (LoRA) specifically for logical fallacy classification. Unlike earlier data augmentation techniques that often produce templated or context-less examples, the *MisSynth* pipeline enforces a same-source retrieval constraint. This crucial step ensures the generated synthetic arguments are grounded in the source scientific article and are

realistic. By utilizing this pipeline, we introduce an efficient and effective method for specializing large language models for complex scientific reasoning tasks, particularly in scenarios where high-quality annotated data is scarce.

The rest of this paper is structured as follows. We first review related work. Then, we detail our methodology. Next, we present our experiments and results. Finally, we discuss our findings and suggest future research directions.

2 Related Work

2.1 Fallacy Detection and Scientific Misinformation

Detecting flawed reasoning in arguments is a significant challenge, particularly within the context of scientific misinformation (Wachsmuth et al. 2017). Traditional methods often fail due to scientific misinformation because fallacies are implicit and heavily dependent on context (Boudry et al. 2015). While recent work utilizes LLMs, their ability to classify subtle reasoning errors remains limited, underscoring the need for improved training data and methods (Ruiz-Dolz and Lawrence 2023).

2.2 Synthetic Data Generation

Synthetic data generation offers a way to augment scarce training resources (Chung et al. 2023; Sennrich et al. 2016). However, some methods produce templated data, like the *LFUD* dataset, which lacks the complexity of real-world arguments (Li et al. 2024). Our RAG-based approach generates more diverse and contextually grounded examples by drawing from authentic scientific texts.

2.3 Fine-Tuning

Full fine-tuning of large models is often impractical. Parameter-efficient fine-tuning (PEFT) methods provide an efficient alternative. We use Low-Rank Adaptation (LoRA), which freezes the pre-trained model and injects small, trainable matrices into its layers (Hu et al. 2021). This approach greatly reduces the number of trainable parameters and memory usage. LoRA allows effective fine-tuning on consumer-grade hardware without sacrificing performance.

2.4 MISSCI

Our work uses a recent benchmark designed for fallacy detection. The *MISSCI* dataset provides a formal framework for our task (Glockner et al. 2024). It models misinformation as an argument where an inaccurate claim, \bar{c} , is supported by an accurate premise, P , and a fallacious premise, \bar{P} . The accurate premise alone does not support the claim ($P \not\Rightarrow \bar{c}$), but the combination does ($P \cup \bar{P} \Rightarrow \bar{c}$). Each fallacious step is a triplet $R_i = (s_i, \bar{p}_i, f_i)$, composed of scientific context s_i , a fallacious premise \bar{p}_i , and a fallacy class f_i . The complete argument is represented as:

$$\begin{array}{c} s_0 & s_1 & s_N \\ \overline{\overline{p}_0}, \overline{\overline{p}_1}, \dots, \overline{\overline{p}_N} & \Rightarrow \bar{c} \\ \downarrow & \downarrow & \downarrow \\ f_0 & f_1 & f_N \end{array} \quad (1)$$

The task involves identifying and classifying the fallacious premise \bar{p}_i and its type f_i . Our work focuses on the classification part of *MISSCI*. Its extension, *MISSCIPlus*, incorporates additional arguments identified in the original scientific texts into the original dataset (Glockner et al. 2025). Models must first find the relevant passage from the article before recognizing the fallacy.

2.5 Other Related Benchmarks

Other notable benchmarks also confirm that LLMs struggle with nuanced argumentation, motivating our approach to improve model training. The *LOGIC* dataset and its climate-focused subset, *LogicClimate*, provide a general reasoning challenge for language models (Jin et al. 2022). The bilingual *RuoZhiBench* uses subtle logical inconsistencies to highlight the performance gap between LLMs and humans (Zhai et al. 2025). The *Fallacies* dataset offers a hierarchical taxonomy of over 200 fallacy types to assess the self-verification capabilities of LLMs (Hong et al. 2024). *MAFALDA* unifies several previous datasets and introduces a "disjunctive annotation scheme" to account for the subjectivity of fallacy annotation by allowing multiple labels (Helwe et al. 2024).

3 Methodology

Detecting health misinformation that misuses scientific claims is a significant problem, partially due to the scarcity of real-world annotated data. This section describes *MisSynth* methodology, which tackles the data shortage. We generate synthetic data to fine-tune Large Language Models (LLMs) for this task. The complete process is shown in Figure 1. Our method first retrieves relevant text using RAG (3.1). An LLM then uses this text to create new fallacy examples (3.2). We use this new dataset to locally fine-tune a model using LoRA (3.3). In addition, we detail our evaluation strategy (3.4).

3.1 RAG for publication context

We base synthetic examples on the same publication contexts that produce fallacious reasoning in *MISSCI*. For each instance in the dev split, we download the cited source S and segment it with a recursive character splitter (chunk size 512, overlap 64). Each passage $d_j \in S$ is embedded with a PubMedBERT biomedical encoder by Gu et al. (2021) ϕ , yielding

$$\mathbf{e}_j = \phi(d_j) \in \mathbb{R}^m \quad (2)$$

We store the passages in the Langchain's (Chase 2022) in-memory vector index along with metadata $\text{source}(d_j) = u$. At retrieval time, we build the query from the inaccurate claim only,

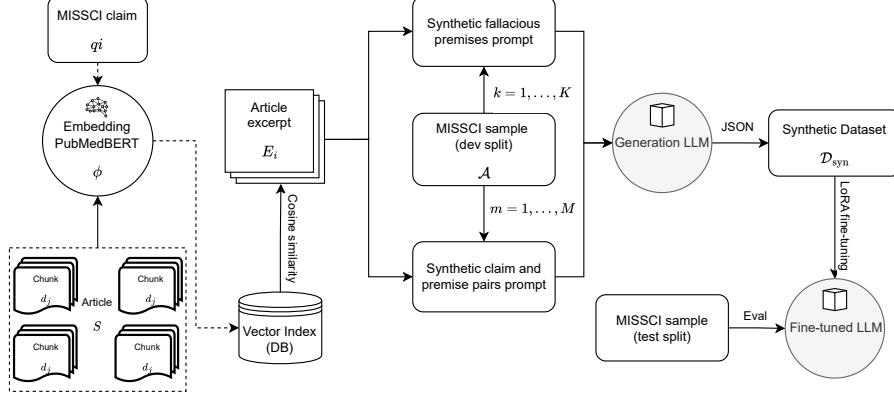


Fig. 1 Overview of *MisSynth* synthetic data generation and fine-tuning pipeline. A RAG retrieves an article excerpt (E_i) from a source article (S) based on a MISSCI claim (q_i). This excerpt, along with a dev split sample (\mathcal{A}), is used by a *Generation LLM* to create a synthetic dataset (\mathcal{D}_{syn}). This dataset is then used to fine-tune a model with LoRA, which is finally evaluated on the MISSCI test split.

$$q_i = \bar{c}, \quad \mathbf{e}_{q_i} = \phi(q_i) \quad (3)$$

and compute the cosine similarity:

$$\text{sim}(q_i, d_j) = \frac{\mathbf{e}_{q_i}^\top \mathbf{e}_j}{\|\mathbf{e}_{q_i}\| \|\mathbf{e}_j\|} \quad (4)$$

We then retrieve the top- k passages subject to a same-source constraint, implemented as a metadata filter:

$$\mathcal{R}_k(q_i, u) = \arg \max_{d_j \in \mathcal{D}, \text{source}(d_j)=u} \text{sim}(q_i, d_j) \quad (5)$$

with $k = 5$. We concatenate the retrieved passages into a single excerpt:

$$E_i = \text{concat}(\mathcal{R}_k(q_i, u)). \quad (6)$$

which is then used to generate the final answer. This setup follows retrieval with dual encoders (one for the query and one for the RAG passages) and top- k similarity search, as well as a multi-passage sequence for generation (Lewis et al. 2020). At the same time, the same-source filter enforces the *MISSCI* assumption that fallacious reasoning is based on the cited source.

3.2 Synthetic Data

Let an annotated *MISSCI* argument be $\mathcal{A} = (\bar{c}, p_0, R)$, where each reasoning step is $R_i = (s_i, \bar{p}_i, f_i)$ with publication context s_i , fallacious premise \bar{p}_i , and fallacy class f_i . For each dev instance, we extract the set of gold fallacies and their classes:

$$\mathcal{F}^{\text{real}} = \{(\bar{p}_\ell^{\text{real}}, f_\ell^{\text{real}}, s_\ell^{\text{real}})\}_\ell \quad (7)$$

format a prompt with $(\bar{c}, p_0, \mathcal{F}^{\text{real}}, E_i)$, and use a *Generation LLM* to produce structured JSON. We generate two kinds of synthetic data:

3.2.1 Synthetic fallacious premises.

We sample K synthetic variants per dev split instance as triples of synthetic context, fallacious premise, and class:

$$(\tilde{s}_{i,k}, \tilde{\bar{p}}_{i,k}, \tilde{f}_{i,k}) \sim p_\theta(s, \bar{p}, f \mid \bar{c}, p_0, \mathcal{F}^{\text{real}}, E_i), \quad k = 1, \dots, K \quad (8)$$

Each item must use a class from the fallacy inventory and be derived from the content of E_i .

3.2.2 Synthetic claim–premise pairs.

When enabled, we also sample M coherent claim / accurate-premise pairs supported by the same source and excerpt:

$$(\tilde{c}_{i,m}, \tilde{p}_{0,i,m}) \sim p_\theta(c, p_0 \mid \mathcal{F}^{\text{real}}, E_i), \quad m = 1, \dots, M \quad (9)$$

to increase diversity of inputs, since each K fallacious premises from the above contain the same real claim–premise pairs per instance.

3.2.3 Prompting and parsing.

Prompts include the fallacy inventory (extracted from a template file) and require a strict JSON array with fields "context", "fallacy", and "class" (Appendices A, B). We skip instances with empty retrieval results or invalid JSON. The temperature of retrieval LLM is kept at 1.0, where applicable.

3.2.4 Train/validation set for fine-tuning.

We convert synthetic items into instruction–completion pairs using *MISSCI*'s "classify with definition" template. For each synthetic fallacy:

$$x_{i,j} = T(\bar{c}, p_0, \tilde{s}_{i,j}, \tilde{\bar{p}}_{i,j}), \quad y_{i,j} = \text{"Fallacy: } \hat{f}_{i,j} \text{"} \quad (10)$$

We form the training set $\mathcal{D}_{\text{syn}} = \{(x_{i,j}, y_{i,j})\}$. The validation set uses only gold *MISSCI* dev examples (original interchangeable fallacies) formatted with the same template, ensuring that validation contains no synthetic completions.

We also include a *random-baseline ablation* that replaces synthetic contexts and premises with lorem ipsum while keeping answers intact, to test whether gains come from synthetic content rather than prompt template or answer structure.

3.3 Fine-tuning

We adapt a frozen base model with parameters Φ_0 using trainable LoRA (Hu et al. 2021) parameters Θ while keeping Φ_0 fixed. Let $\mathcal{Z} = (x, y)$ denote the training pairs produced from \mathcal{D}_{syn} . We maximize the conditional likelihood with respect to the adapter parameters:

$$\max \Theta \sum_{(x,y) \in \mathcal{Z}} \sum_{t=1}^{|y|} \log(p_{\Phi_0 + \Delta\Phi(\Theta)}(y_t | x, y_{<t})) \quad (11)$$

In LoRA, the task-specific increment $\Delta\Phi(\Theta)$ is encoded by low-rank updates to selected linear projections, typically the attention projections W_q and W_v . For any adapted weight $W_0 \in \mathbb{R}^{d \times k}$ in Φ_0 , we learn $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times k}$ with $r \ll \min d, k$ and set

$$W = W_0 + \frac{\alpha}{r} AB \quad (12)$$

where only A and B are trainable and α is a fixed scaling factor. All other parameters of Φ_0 remain frozen.

3.4 Evaluation

We evaluate the *MisSynth* pipeline by fine-tuning several LLMs using Low-Rank Adaptation (LoRA) and testing their performance on the classification sub-task of the *MISSCI* benchmark. Our primary evaluation metrics are accuracy (Acc) and macro-averaged F1-score (F1) on the *MISSCI* test split.

All fine-tuning experiments were performed using the LoRA technique with a rank of $r = 8$ on the attention projections (W_q and W_v). The training set \mathcal{D}_{syn} was generated from the *MISSCI* dev set using the *Generation LLM* at a temperature of 1.0, where applicable. We use the gold *MISSCI* dev examples as a consistent fine-tuning validation set across all runs (96 samples). All experiments were conducted locally on an M1 MacBook Pro with 32 GB of unified memory using the MLX framework by Hannun et al. (2023).

4 Results

This section details our experimental findings. We first optimize the synthetic data generation parameters in 4.1. We then select the best LLM to create the dataset in 4.2 and explore some of the created dataset statistics in 4.3. Finally, the most important results are presented in 4.4, where we benchmark several models fine-tuned on this data. These show the final benchmark performance. Our results confirm that fine-tuning with our synthetic data dramatically improves model performance.

4.1 Optimization of Data Generation Parameters

We first analyzed the impact of varying the number of synthetic fallacious premises (K) and synthetic claim/premise pairs (M) on the performance of a fine-tuned Phi-4 (8-bit) model by Abdin et al. (2024). The results are presented in Table 1.

Table 1 Fine-tuned performance of Phi-4 (8-bit) with varying synthetic data parameters K and M . LoRA layers are 16. All fine-tuning runs were executed for 500 iterations. Performance is measured on the *MISSCI* test split.

K	M	Generation LLM	Val Loss 1 iter	Val Loss 500 iters	Acc	F1 (macro)	Train Samples
0	0	Vanilla	-	-	0.667	0.550	-
10	0	Random baseline	1.938	0.166	0.606	0.512	299
10	0	o4-mini	1.938	0.147	0.685	0.622	299
15	5	o4-mini	1.943	0.076	0.711	0.654	929
30	15	o4-mini	1.940	0.067	0.762	0.690	2344
40	20	o4-mini	1.943	0.074	0.711	0.647	2984

The vanilla Phi-4 model achieved an F1-score of 0.550. Fine-tuning consistently improved performance, demonstrating the effectiveness of the synthetic data, which is further validated by the significant drop in the validation loss from approximately 1.94 down to as low as 0.067 for all successful configurations. Notably, the *Random baseline ablation* underperformed (F1 of 0.512), despite showing an initial Val Loss of 1.938 dropping to 0.166, confirming that the model learned from the synthetic data, rather than the prompt template or answer structure alone.

We observed a maximum F1-score of 0.690 and a maximum accuracy of 0.762 at $K = 30$ and $M = 15$. Increasing the data volume further to $K = 40$ and $M = 20$ led to a decrease in performance, with the F1-score dropping to 0.647. Considering this peak in performance and the associated generation/fine-tuning costs, we selected the configuration $K = 30$ and $M = 15$ for subsequent experiments. This configuration achieved a competitive F1-score of 0.690 (a 14% absolute gain over the vanilla model), representing the optimal setting with 2344 training samples and a favorable validation loss reduction from 1.940 to 0.067.

4.2 Selecting the Generation LLM

Next, we investigated whether the quality of the LLM used for synthetic data generation impacts the final fine-tuned model’s performance. Using the optimal parameters ($K = 30, M = 15$), we compared four different generator models (Table 2).

Table 2 Comparison of different LLMs used for generating synthetic data (\mathcal{D}_{syn}) for Phi-4 (8-bit) fine-tuning. $K = 30, M = 15$, LoRA layers are 16. All fine-tuning runs were executed for 500 iterations.

Generation LLM	Fine-tuned LLM	Val Loss 1 iter	Val Loss 500 iters	Acc	F1 (macro)
o4-mini	Phi-4 (8-bit)	1.940	0.067	0.762	0.690
GPT-4.1	Phi-4 (8-bit)	1.951	0.058	0.751	0.653
GPT-5 (medium)	Phi-4 (8-bit)	1.945	0.063	0.764	0.705
o3	Phi-4 (8-bit)	1.945	0.081	0.731	0.621

We observed that the data generated by GPT-5 (medium) resulted in the highest F1-score (0.705) and accuracy (0.764), demonstrating its superior ability to generate high-quality training examples. Therefore, prioritizing maximum performance for this task, we selected GPT-5 (medium) as the best generation model for *MisSynth*, despite the higher generation cost.

4.3 Optimal Synthetic Dataset

We publicly release our optimal synthetic dataset. We note that the dataset was generated once using the final iteration of the *MisSynth* code with $K = 30$ and $M = 15$ (GPT-5). This dataset is used for all subsequent experiments. The dataset was generated once. Table 3 details the distribution of fallacy categories. The synthetic dataset’s distribution differs from the *MISSCI* splits. For instance, *Fallacy of Exclusion* comprises a smaller portion (7.44%) compared to the test split (27.53%). Conversely, some minority classes in the *MISSCI* test split, such as *False Dilemma* (4.19%) and *Impossible Expectations* (1.32%), are represented more frequently in the synthetic data (11.21% and 8.44%, respectively).

Table 3 Distribution of Fallacy Categories (Count and Percentage) across datasets.

Fallacy Category	MISSCI, dev split	MISSCI, test split	GPT-5 \mathcal{D}_{syn} dataset ($K = 30, M = 15$)
Ambiguity	7 (7.29%)	44 (9.69%)	129 (14.32%)
Biased Sample Fallacy	10 (10.42%)	37 (8.15%)	84 (9.32%)
Causal Oversimplification	14 (14.58%)	73 (16.08%)	133 (14.76%)
Fallacy of Division/Composition	7 (7.29%)	33 (7.27%)	73 (8.10%)
Fallacy of Exclusion	25 (26.04%)	125 (27.53%)	67 (7.44%)
False Dilemma	8 (8.33%)	19 (4.19%)	101 (11.21%)
False Equivalence	14 (14.58%)	85 (18.72%)	115 (12.76%)
Hasty Generalization	6 (6.25%)	32 (7.05%)	123 (13.65%)
Impossible Expectations	5 (5.21%)	6 (1.32%)	76 (8.44%)
Overall	96 (100.00%)	454 (100.00%)	901 (100.00%)

Table 4 shows the ROUGE recall (Lin 2004), measuring textual overlap between entities and their source article excerpt E_i . The synthetic *Context* (0.766) and *Accurate Premise* (0.862) show higher recall than the *MISSCI dev split* (0.635 and 0.741). In contrast, the synthetic *Fallacy* (0.493) and *Claim* (0.612) have lower recall than the dev split (0.608 and 0.642). This suggests that both synthetic data and *MISSCI* is well-grounded in the source article. Textual comparison of dataset entities are available in Appendix C tables C1, C2, C3.

4.4 Evaluation of Fine-tuned Models

Finally, we benchmarked the performance gains achieved by fine-tuning various LLMs using our GPT-5 \mathcal{D}_{syn} dataset ($K = 30, M = 15$). Table 5 compares the vanilla and fine-tuned performance.

Table 4 ROUGE recall between article excerpt E_i and entity.

Dataset Entity	ROUGE MISSCI dev split	ROUGE GPT-5 \mathcal{D}_{syn} dataset ($K = 30, M = 15$)
Fallacy (K)	0.608	0.493
Context (K)	0.635	0.766
Claim (M)	0.642	0.612
Accurate Premise (M)	0.741	0.862

Table 5 Comparison of different base models before and after fine-tuning with GPT-5 \mathcal{D}_{syn} ($K = 30, M = 15$). All fine-tuning runs were executed for 500 iterations. Performance measured on the *MISSCI* test split.

Fine-tuned LLM	Val Loss 1 iter	Val Loss 500 iters	Vanilla Acc	Vanilla F1	Fine Acc	Fine F1	LLM Size	LoRA Layers
Gemma 3 (8-bit)	3.324	0.067	0.531	0.377	0.764	0.691	4B	32
LLaMA 3.1 (4-bit)	2.451	0.050	0.414	0.334	0.778	0.711	8B	32
LLaMA 2 (4-bit)	2.145	0.073	0.326	0.218	0.722	0.681	13B	32
Phi-4 (8-bit)	1.945	0.063	0.667	0.550	0.764	0.705	15B	16
Mistral Small 3.2 (4-bit)	2.124	0.072	0.698	0.553	0.762	0.718	24B	16
LLaMA 2 *	-	-	0.577	0.464	-	-	70B	-
GPT-4 *	-	-	0.738	0.649	-	-	-	-

* Results by [Glockner et al. \(2024\)](#)

The results confirm that the *MisSynth* significantly improves performance across different model architectures. All fine-tuned models showed substantial decreases in validation loss, with LLaMA 3.1 (8B) by [Grattafiori et al. \(2024\)](#) dropping from 2.451 to 0.050 and Gemma 3 (4B) by [Team et al. \(2025\)](#) dropping from 3.324 to 0.067, indicating successful adaptation to the fallacy classification task.

The LLaMA 2 13B model ([Touvron et al. 2023](#)) showed the largest absolute improvement, increasing its F1-score from a baseline of 0.218 to 0.681, alongside a validation loss reduction from 2.145 to 0.073. The fine-tuned Mistral Small 3.2 model achieved the highest F1-score overall at 0.718 (a 16.5% absolute gain). Other models also showed strong performance, with LLaMA 3.1 achieving 0.711 (37.7% absolute gain) and Phi-4 reaching 0.705 (15.5% absolute gain). Notably, several fine-tuned smaller models outperformed the proprietary model. Our fine-tuned Mistral Small 3.2 ([Mistral AI 2025](#)), LLaMA 3.1, Phi-4, and Gemma 3 (F1 of 0.691) all surpassed the vanilla GPT-4 model ([OpenAI et al. 2024](#)), which was reported to have an F1 of 0.649.

Due to VRAM limitations, we were unable to fine-tune or evaluate the LLaMA 2 70B ([Touvron et al. 2023](#)) model directly. Therefore, the reported vanilla performance for the LLaMA 2 70B (F1: 0.464, Acc: 0.577) and GPT-4 (F1: 0.649, Acc: 0.738) is taken from Table 3 of the original MISSCI paper ([Glockner et al. 2024](#)). Critically, the fine-tuned LLaMA 2 13B (F1: 0.681) substantially outperformed the

vanilla, much larger LLaMA 2 70B model (F1: 0.464). This highlights a key finding: targeted training using high-quality, RAG-supported synthetic data can close the performance gap between small, parameter-efficient models and large foundation models for domain-specific tasks like fallacy classification.

Table 6 Comparison of Vanilla vs. Fine-Tuned LLaMA 2 13B F1-Scores by Fallacy Category on the *MISSCI* test split.

Fallacy Category	Count	Vanilla F1 (macro)	Fine-Tuned F1 (macro)	Absolute Gain
Ambiguity	44	0.044	0.333	0.289
Biased Sample Fallacy	37	0.143	0.704	0.561
Causal Oversimplification	73	0.485	0.820	0.335
Fallacy of Division/Composition	33	0.050	0.485	0.435
Fallacy of Exclusion	125	0.110	0.954	0.844
False Dilemma	19	0.148	0.812	0.664
False Equivalence	85	0.614	0.479	-0.135
Hasty Generalization	32	0.586	0.912	0.326
Impossible Expectations	6	0.000	0.632	0.632
Macro Average F1	454	0.218	0.681	0.463
Accuracy	454	0.326	0.722	0.396

The category-specific analysis of LLaMA 2 13B model results reveals that the largest absolute improvement in macro F1-score from 0.218 to 0.681 across all fallacy categories is driven by dramatic performance improvements across nearly all categories, especially those where the vanilla model was weakest. The most significant improvements were seen in *Fallacy of Exclusion*, which rose from an F1-score of 0.110 to 0.954, and *False Dilemma*, which increased from 0.148 to 0.812. Furthermore, the model learned to identify the highly minority class *Impossible Expectations*, improving from an F1-score of zero to 0.632. Strong gains were also observed in other low-performing categories such as *Biased Sample Fallacy* (0.143 to 0.704). Notably, performance on *False Equivalence* decreased from 0.614 to 0.479 after fine-tuning. Overall, the results demonstrate that our synthetic data generation pipeline is highly effective at strengthening model performance, particularly on challenging fallacy classes, significantly improving the model’s overall robustness, F1 score and accuracy.

5 Discussion

We introduced *MisSynth*, a novel pipeline for generating high-quality synthetic data to detect scientific fallacies. Our method significantly improves the performance of LLMs on the *MISSCI* benchmark. Fine-tuning even small models, such as LLaMA 3.1 8B, with our data yielded substantial gains, surpassing the performance of much larger vanilla models, like GPT-4. This demonstrates that targeted, parameter-efficient fine-tuning with context-aware synthetic data is an effective strategy for specialized reasoning tasks.

5.1 Limitations

Our research's primary limitation is its exclusive focus on the *MISSCI* benchmark by [Glockner et al. \(2024\)](#). Consequently, our synthetic data and fine-tuned models are specialized for this dataset. Furthermore, our methodology addresses only the classification sub-task. We do not evaluate the generation of fallacious premises, which is another part of the *MISSCI* dataset.

5.2 Future work

Future work includes generalizing *MisSynth*. We aim to adapt the method for other fallacy benchmarks, such as those mentioned in our related work, like *MAFALDA* by [Helwe et al. \(2024\)](#). We also plan to scale our solution. This involves moving beyond local hardware to fine-tune larger models on cloud infrastructure.

5.3 Ethical considerations

Our synthetic dataset was generated automatically by an LLM. No medical experts or health professionals reviewed the synthetic data. There is a potential danger that malicious actors could exploit our synthetic data to spread health misinformation more effectively.

Acknowledgements. We thank Max Glockner for validating the initial idea and appreciate his helpful feedback during the development of this work.

Appendix A Single Class Synthetic Fallacy Prompt Template

You are provided with a claim, an accurate premise for the claim, a list of real-world fallacious premises (fallacies) from the scientific article with the fallacy class, and relevant text exempt from this article.

Claim: {claim} Accurate Premise: {premise}

{fallacies}

Article Excerpt: {article_excerpt}

Task:

Based on the example and relevant text from the article, create {n_entries} synthetic fallacies that differ from the provided real-world fallacies and their class in the JSON format:

```
[  
  {  
    "context": // Synthetic Context 1,  
    "fallacy": // Synthetic Fallacy 1,  
    "class": // Synthetic Class 1  
  },
```

```

{
    "context": // Synthetic Context 2,
    "fallacy": // Synthetic Fallacy 2,
    "class": // Synthetic Class 2
},
...
{
    "context": // Synthetic Context {n_entries},
    "fallacy": // Synthetic Fallacy {n_entries},
    "class": // Synthetic Class {n_entries}
}
]

```

Creating fallacies of the classes different from provided real-world examples is encouraged, but the class could be only from the fallacy inventory.

{fallacy_inventory}

Structure created fallacy text similarly to real-world examples.

Appendix B Synthetic Claim-Accurate Premise Prompt Template

You are provided with a claim, an accurate premise, a list of real-world fallacious premises (fallacies) from the scientific article with the fallacy class, and relevant text exempt from this article.

Claim: {claim} Accurate Premise: {premise}

{fallacies}

Article Excerpt: {article_excerpt}

Task:

Based on the example and relevant text from the article, create {n_entries} synthetic claim and accurate premise pairs that differ from the provided real-world premises in the JSON format. Make sure that the created claim-accurate premise pair is coherent.

```

[
{
    "premise": // Synthetic Accurate Premise 1,
    "claim": // Synthetic Claim 1,
},
{
    "premise": // Synthetic Accurate Premise 2,
    "claim": // Synthetic Claim 2,
},
...

```

```

    {
        "premise": // Synthetic Accurate Premise {n_entries},
        "claim": // Synthetic Claim {n_entries}
    }
]

```

Structure created claims and accurate premises text similarly to real-world examples.

Appendix C Optimal Synthetic Dataset Examples

Table C1 Comparison of *MISSCI* dev split vs. randomly chosen Synthetic Claim-Accurate Premise pairs (GPT-5 \mathcal{D}_{syn} dataset $K = 30$, $M = 15$, argument ID 171)

Source	Claim	Accurate Premise
<i>MISSCI</i> dev split	COVID-19 immunity likely lasts for years.	Different types of immune cells contributing to immune memory and long-term protection remained detectable in the blood of recovered COVID-19 patients
Synthetic	SARS-CoV-2 T cell memory may stabilize rather than rapidly decline over time.	Data suggest T cell memory may reach a stable plateau beyond the first eight months after infection.
Synthetic	Long-term protection from COVID-19 depends on durable immune memory.	Immune memory is the source of long-term protective immunity against reinfection.
Synthetic	Definitive conclusions about the duration of COVID-19 immunity are still premature.	The overall amount of data on protective immunity to SARS-CoV-2 remains limited.

References

- Abdin M, Aneja J, Behl H, et al (2024) Phi-4 technical report. URL <https://arxiv.org/abs/2412.08905>, arXiv:2412.08905
- Beers A, Nguyen S, Starbird K, et al (2023) Selective and deceptive citation in the construction of dueling consensuses. Science Advances 9. <https://doi.org/10.1126/sciadv.adh1933>
- Boudry M, Paglieri F, Pigliucci M (2015) The fake, the flimsy, and the fallacious: Demarcating arguments in real life. Argumentation 29(4):431–456. <https://doi.org/10.1007/s10503-015-9359-1>, URL <https://doi.org/10.1007/s10503-015-9359-1>
- Brennen J, Simon F, Howard P, et al (2020) Types, sources, and claims of covid-19 misinformation. Tech. rep., Reuters Institute for the Study of Journalism

- Chase H (2022) LangChain. URL <https://github.com/langchain-ai/langchain>
- Chung J, Kamar E, Amershi S (2023) Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In: Rogers A, Boyd-Graber J, Okazaki N (eds) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Toronto, Canada, pp 575–593, <https://doi.org/10.18653/v1/2023.acl-long.34>, URL <https://aclanthology.org/2023.acl-long.34/>
- Frederick DE (2023) Zombie papers, the data deluge column. Library Hi Tech News 40(9):1–6. <https://doi.org/10.1108/LHTN-10-2023-0194>, URL <https://doi.org/10.1108/LHTN-10-2023-0194>, <https://www.emerald.com/lhtn/article-pdf/40/9/1/1790212/lhtn-10-2023-0194.pdf>
- Glockner M, Hou Y, Nakov P, et al (2024) Missci: Reconstructing fallacies in misrepresented science. In: Ku LW, Martins A, Srikanth V (eds) Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Bangkok, Thailand, pp 4372–4405, <https://doi.org/10.18653/v1/2024.acl-long.240>, URL [https://aclanthology.org/2024.acl-long.240/](https://aclanthology.org/2024.acl-long.240)
- Glockner M, Hou Y, Nakov P, et al (2025) Grounding fallacies misrepresenting scientific publications in evidence. In: Chiruzzo L, Ritter A, Wang L (eds) Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Association for Computational Linguistics, Albuquerque, New Mexico, pp 9732–9767, <https://doi.org/10.18653/v1/2025.naacl-long.491>, URL [https://aclanthology.org/2025.naacl-long.491/](https://aclanthology.org/2025.naacl-long.491)
- Grattafiori A, Dubey A, Jauhri A, et al (2024) The llama 3 herd of models. URL <https://arxiv.org/abs/2407.21783>, arXiv:2407.21783
- Gu Y, Tinn R, Cheng H, et al (2021) Domain-specific language model pretraining for biomedical natural language processing. ACM Trans Comput Healthcare 3(1). <https://doi.org/10.1145/3458754>, URL <https://doi.org/10.1145/3458754>
- Guo Z, Schlichtkrull M, Vlachos A (2022) A survey on automated fact-checking. Transactions of the Association for Computational Linguistics 10:178–206. https://doi.org/10.1162/tacl_a_00454, URL https://doi.org/10.1162/tacl_a_00454, https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00454/1987018/tacl_a_00454.pdf
- Hannun A, Digani J, Katharopoulos A, et al (2023) MLX: Efficient and flexible machine learning on apple silicon. URL <https://github.com/ml-explore>
- Helwe C, Calamai T, Paris PH, et al (2024) MAFALDA: A benchmark and comprehensive study of fallacy detection and classification. In: Duh K, Gomez H, Bethard

S (eds) Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Association for Computational Linguistics, Mexico City, Mexico, pp 4810–4845, <https://doi.org/10.18653/v1/2024.naacl-long.270>, URL <https://aclanthology.org/2024.naacl-long.270/>

Hong R, Zhang H, Pang X, et al (2024) A closer look at the self-verification abilities of large language models in logical reasoning. In: Duh K, Gomez H, Bethard S (eds) Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Association for Computational Linguistics, Mexico City, Mexico, pp 900–925, <https://doi.org/10.18653/v1/2024.naacl-long.52>, URL <https://aclanthology.org/2024.naacl-long.52/>

Hu EJ, Shen Y, Wallis P, et al (2021) Lora: Low-rank adaptation of large language models. CoRR abs/2106.09685. URL <https://arxiv.org/abs/2106.09685>, 2106.09685

Jin Z, Lalwani A, Vaidhya T, et al (2022) Logical fallacy detection. In: Goldberg Y, Kozareva Z, Zhang Y (eds) Findings of the Association for Computational Linguistics: EMNLP 2022. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, pp 7180–7198, <https://doi.org/10.18653/v1/2022.findings-emnlp.532>, URL <https://aclanthology.org/2022.findings-emnlp.532/>

Lewandowsky S, Cook J, Lombardi D (2020) Debunking handbook 2020

Lewis P, Perez E, Piktus A, et al (2020) Retrieval-augmented generation for knowledge-intensive nlp tasks. In: Larochelle H, Ranzato M, Hadsell R, et al (eds) Advances in Neural Information Processing Systems, vol 33. Curran Associates, Inc., pp 9459–9474

Li Y, Wang D, Liang J, et al (2024) Reason from fallacy: Enhancing large language models' logical reasoning through logical fallacy understanding. In: Duh K, Gomez H, Bethard S (eds) Findings of the Association for Computational Linguistics: NAACL 2024. Association for Computational Linguistics, Mexico City, Mexico, pp 3053–3066, <https://doi.org/10.18653/v1/2024.findings-naacl.192>, URL <https://aclanthology.org/2024.findings-naacl.192/>

Lin CY (2004) Rouge: A package for automatic evaluation of summaries. URL <https://aclanthology.org/W04-1013/>

Mistral AI (2025) Mistral Small 3. URL <https://mistral.ai/news/mistral-small-3>

Møller AG, Pera A, Dalsgaard J, et al (2024) The parrot dilemma: Human-labeled vs. LLM-augmented data in classification tasks. In: Graham Y, Purver M (eds) Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, St. Julian's, Malta, pp 179–192, <https://doi.org/10.18653/v1/>

2024.eacl-short.17, URL <https://aclanthology.org/2024.eacl-short.17/>

- Nakov P, Corney D, Hasanain M, et al (2021) Automated fact-checking for assisting human fact-checkers. In: Zhou ZH (ed) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. International Joint Conferences on Artificial Intelligence Organization, pp 4551–4558, <https://doi.org/10.24963/ijcai.2021/619>, URL <https://doi.org/10.24963/ijcai.2021/619>, survey Track
- OpenAI, Achiam J, Adler S, et al (2024) Gpt-4 technical report. URL <https://arxiv.org/abs/2303.08774>, arXiv:2303.08774
- Ruiz-Dolz R, Lawrence J (2023) Detecting argumentative fallacies in the wild: Problems and limitations of large language models. In: Alshomary M, Chen CC, Muresan S, et al (eds) Proceedings of the 10th Workshop on Argument Mining. Association for Computational Linguistics, Singapore, pp 1–10, <https://doi.org/10.18653/v1/2023.argmining-1.1>, URL <https://aclanthology.org/2023.argmining-1.1>
- Sennrich R, Haddow B, Birch A (2016) Improving neural machine translation models with monolingual data. In: Erk K, Smith NA (eds) Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Berlin, Germany, pp 86–96, <https://doi.org/10.18653/v1/P16-1009>, URL <https://aclanthology.org/P16-1009>
- Team G, Kamath A, Ferret J, et al (2025) Gemma 3 technical report. URL <https://arxiv.org/abs/2503.19786>, arXiv:2503.19786
- Touvron H, Lavril T, Izacard G, et al (2023) Llama: Open and efficient foundation language models. arXiv:2302.13971
- Vosoughi S, Roy D, Aral S (2018) The spread of true and false news online. Science 359(6380):1146–1151. <https://doi.org/10.1126/science.aap9559>, URL <https://www.science.org/doi/abs/10.1126/science.aap9559>, <https://www.science.org/doi/pdf/10.1126/science.aap9559>
- Wachsmuth H, Naderi N, Habernal I, et al (2017) Argumentation quality assessment: Theory vs. practice. In: Barzilay R, Kan MY (eds) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, Vancouver, Canada, pp 250–255, <https://doi.org/10.18653/v1/P17-2039>, URL <https://aclanthology.org/P17-2039>
- Zhai Z, Li H, Han X, et al (2025) Ruozhibench: Evaluating llms with logical fallacies and misleading premises. URL <https://arxiv.org/abs/2502.13125>, arXiv:2502.13125

Table C2 Comparison of *MISSCI* dev split vs. randomly chosen Synthetic Claim-Accurate Premise pairs (GPT-5 \mathcal{D}_{syn} dataset $K = 30, M = 15$, argument ID 171). "N/A" means that the entity is not present in the argument. Part 1

Fallacy Category	<i>MISSCI</i> dev split Fallacy Example	<i>MISSCI</i> dev split Context Example	Synthetic Fallacy Example	Synthetic Context Example
Ambiguity	Stating that COVID-19 immunity likely lasts "for years" is precise enough to understand that antibodies, memory B cells, and memory T cells were found five to eight months after infection.	Different types of immune cells remained detectable in the blood of recovered COVID-19 patients for up to eight months.	Blunting disease severity is the same as having full immunity for years.	Sub-sterilizing neutralizing antibody titers can blunt the size of the initial infection and limit COVID-19 severity.
Biased Sample Fallacy	N/A	N/A	Citing extreme cases of long-lived memory implies typical COVID-19 immunity is similarly long-lived.	Reports of extremely long-lived B cell memory to other infections (e.g., smallpox, influenza).
Causal Oversimplification	Antibodies, memory B cells, and memory T cells provide the main protection against viral infections. Therefore, because they were found months after infection, COVID-19 immunity likely lasts for years.	Antibodies, memory B cells and memory T cells are part of the immune system. Therefore, the immune system lasts for years if antibodies last for years.	A few months of stability cause a guaranteed linear extension into multiple years.	RBD-specific memory B cells showed no apparent decline over 5–8 months.
Fallacy of Division/Composition	Antibodies, memory B cells and memory T cells are part of the immune system. Therefore, the immune system lasts for years if antibodies last for years.	N/A	Because memory cells are present in blood, every tissue and mucosal site will be protected for years.	Detection of immune memory cells in blood up to 8 months post-infection.

Table C3 Comparison of Real-World vs. randomly chosen Synthetic Fallacy Examples by Category (GPT-5 \mathcal{D}_{syn} dataset $K = 30, M = 15$, argument ID 171). "N/A" means that the entity is not present in the argument. Part 2

Fallacy Category	MISSCI dev split Fallacy Example	MISSCI dev split Context Example	Synthetic Fallacy Example	Synthetic Context Example
Fallacy of Exclusion	It is irrelevant to the claim that different types of immune cells remained detectable in the blood of recovered COVID-19 patients for up to eight months.	Different types of immune cells remained detectable in the blood of recovered COVID-19 patients for up to eight months.	By focusing only on cohorts with robust responses and short follow-up, we can conclude multi-year immunity.	Several cohorts detected robust RBD memory B cells within the first 8 months.
False Dilemma	Either something vanishes quickly or stays for years.	Different types of immune cells remained detectable in the blood of recovered COVID-19 patients for up to eight months.	Either immunity blocks all infection or it is worthless; since memory cells are present, they must block infection for years.	Minimizing COVID-19 disease severity by confining SARS-CoV-2 to the upper respiratory tract is a primary goal mediated by memory T and B cells.
False Equivalence	N/A	N/A	Since some pathogens induce decades-long B cell memory, SARS-CoV-2 immunity will last decades too.	B cell memory to smallpox vaccination can last 60+ years, and after influenza infection 90+ years.
Hasty Generalization	N/A	N/A	Therefore, SARS-CoV-2 immunity lasts at least 17 years in humans.	SARS-CoV T cells have been detected 17 years after the initial infection.
Impossible Expectations	N/A	N/A	Conclusions are constrained by the limited overall amount of data on protective immunity to SARS-CoV-2.	Because it is impossible to have 10-year follow-up data right now, we should accept that immunity lasts for years based on months of data.