

# Distinguishing Fact from Fiction: A Benchmark Dataset for Identifying Machine-Generated Scientific Papers in the LLM Era.

**Edoardo Mosca**  
TU Munich,  
Department of Informatics,  
Germany  
edoardo.mosca@tum.de

**Mohamed Hesham I. Abdalla**  
TU Munich,  
Department of Informatics,  
Germany  
mohamed.abdalla@tum.de

**Paolo Basso**  
Polytechnic of Milan,  
Department of EIB,  
Italy  
paolo3.basso@mail.polimi.it

**Margherita Musumeci**  
Polytechnic of Milan,  
Department of EIB,  
Italy  
margherita.musumeci@mail.polimi.it

**Georg Groh**  
TU Munich,  
Department of Informatics,  
Germany  
grohg@in.tum.de

## Abstract

As generative NLP can now produce content nearly indistinguishable from human writing, it becomes difficult to identify genuine research contributions in academic writing and scientific publications. Moreover, information in NLP-generated text can potentially be factually wrong or even entirely fabricated. This study introduces a novel benchmark dataset, containing human-written and machine-generated scientific papers from SCIgen, GPT-2, GPT-3, ChatGPT, and Galactica. After describing the generation and extraction pipelines, we also experiment with four distinct classifiers as a baseline for detecting the authorship of scientific text. A strong focus is put on generalization capabilities and explainability to highlight the strengths and weaknesses of detectors. We believe our work serves as an important step towards creating more robust methods for distinguishing between human-written and machine-generated scientific papers, ultimately ensuring the integrity of scientific literature.

## 1 Introduction

Generative *Natural Language Processing* (NLP) systems—often based on *Large Language Models* (LLMs) (Brown et al., 2020; Scao et al., 2022; OpenAI, 2023)—have experienced significant advancements in recent years, with state-of-the-art algorithms generating content that is almost indistinguishable from human-written text (Radford et al., 2019; Zellers et al., 2019; Keskar et al., 2019; Brown et al., 2020). This progress has led to numerous applications in various fields, such as chatbots (OpenAI, 2022), automated content generation (Chen et al., 2021), and even summarization tools

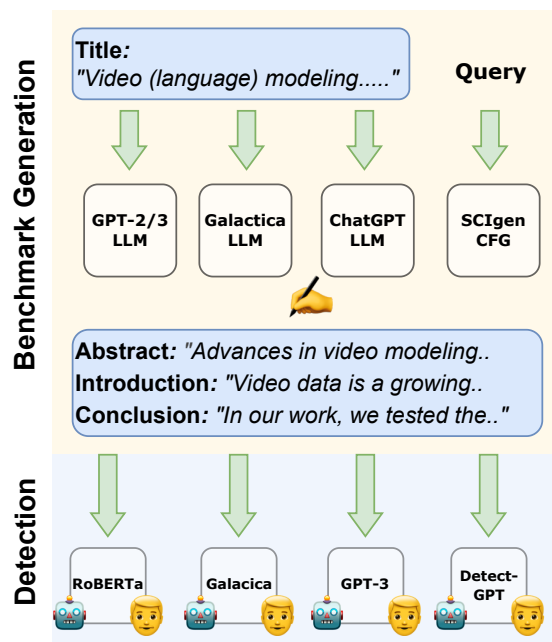


Figure 1: This work’s overview. Five methods are used to machine-generate papers, which are then mixed with human-written ones to create our benchmark. Four models are then tested as baselines to identify the authorship of a given output.

(Liu, 2019). However, these advancements also raise concerns regarding the integrity and authenticity of academic writing and scientific publications (Dergaa et al., 2023; Stokel-Walker, 2022).

It is indeed increasingly difficult to differentiate genuine research contributions from artificially generated content. Moreover, we are at an increased risk of including factually incorrect or entirely fabricated information (Maynez et al., 2020; Tian et al., 2019). Reliably identifying machine-generated sci-

entific publications becomes thus crucial to maintain the credibility of scientific literature and fostering trust among researchers.

This work introduces a novel benchmark to address this issue. Our contribution—also briefly sketched in 1—can be summarized as follow:

- (1) We present a dataset comprising of human-written and machine-generated scientific documents from various sources: SCIgen (Stribling et al., 2005), GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), ChatGPT (OpenAI, 2022), and Galactica (Taylor et al., 2022). Each document includes *abstract*, *introduction*, and *conclusion* in a machine-readable format.
- (2) We experiment with four distinct classifiers—RoBERTa (Liu et al., 2019), Galactica (Taylor et al., 2022), GPT-3 (Brown et al., 2020), and DetectGPT (Mitchell et al., 2023)—as a baseline for detecting the authorship of scientific text, assessing their performance in differentiating between human and machine-generated content.
- (3) We emphasize experimenting with generalization capabilities and explainability to provide insights into the strengths and weaknesses of each detector.

We release our benchmark dataset, baseline models, and testing code to the public to promote further research and aid the development of more robust detection methods. We release our benchmark dataset and baseline models as well as all code used for experimental results<sup>1</sup>.

## 2 Related Work

### 2.1 (Machine-Generated) Scientific Publication Corpora

The ACL Anthology<sup>2</sup> (Bird et al., 2008) and arXiv<sup>3</sup> (arXiv.org submitters, 2023) are widely used resources for accessing scientific texts and their associated metadata. However, these databases do not provide structured text for scientific documents, necessitating the use of PDF parsers and other tools to extract text and resolve references. Several efforts have been made to develop structured text

databases for scientific documents. (Cohan and Goharian, 2015; Saier and Färber, 2019; Lo et al., 2020).

Despite progress in generating text, machine-generated datasets for scientific literature remain limited. A recent study by Kashnitsky et al. (2022) compiled a dataset including retracted, summarized, and paraphrased paper abstracts and excerpts, as well as text generated by GPT-3 (Brown et al., 2020) and GPT-Neo (Black et al., 2021). It’s worth noting that the dataset lists retracted papers as machine-generated, which may not always be accurate, and only includes excerpts or abstracts of the papers.

Liyanage et al. (2022) proposed an alternative approach, in which they generated papers using GPT-2 (Radford et al., 2019) and Arxiv-NLP<sup>4</sup>. However, their dataset was limited to only 200 samples, which were restricted to the fields of Artificial Intelligence and Computation and Language.

### 2.2 Generative NLP for Scientific Articles

Generative NLP for scientific publications has evolved significantly in recent years. Early methods, such as SCIgen (Stribling et al., 2005), used *Context-Free-Grammar* (CFG) to fabricate computer science publications. These often contain nonsensical outputs due to CFG’s limited capacity for generating coherent text.

The advent of attention, transformers (Waswani et al., 2017), and LLMs (Brown et al., 2020) has paved the way for more sophisticated models capable of generating higher-quality scientific content. Some—such as (Devlin et al., 2019), GPT-3 (Brown et al., 2020), ChatGPT (OpenAI, 2022), and Bloom (Scao et al., 2022)—are built for general purposes. Others, instead, are domain-specific and specialized for generating scientific literature. Popular examples in this category are SciBERT (Maheshwari et al., 2021) and Galactica (Taylor et al., 2022).

Both general and domain-specific models have shown outstanding results in various scientific tasks, demonstrating their potential in generating coherent and contextually relevant scientific text. This same technology has also been applied to other domains, including writing news articles (Zellers et al., 2019), producing learning material (MacNeil et al., 2022), and creative writing (Swanson et al., 2021).

<sup>1</sup>[huggingface.co/datasets/tum-nlp/IDMGSP](https://huggingface.co/datasets/tum-nlp/IDMGSP)

<sup>2</sup><https://aclanthology.org/>

<sup>3</sup><https://arxiv.org/>

<sup>4</sup><https://huggingface.co/lysandre/arxiv-nlp>

### 2.3 Detection of Machine-Generated Text

The ability to automatically generate convincing content has motivated researchers to work on its automatic detection, especially given its potential implications for various domains.

Several approaches to detecting machine-generated text have emerged, employing a range of techniques. Some studies have focused on utilizing hand-crafted features (Gehrmann et al., 2019), bag-of-words features (Fagni et al., 2021), or neural features in combination with supervised models to distinguish between human and machine-generated content (Bakhtin et al., 2019; Ippolito et al., 2019; Fagni et al., 2021).

Alternative approaches explore using the probability curvature of the generative model itself (Mitchell et al., 2023) or watermarking machine-generated text to facilitate detection (Kirchenbauer et al., 2023).

### 2.4 Detection of Machine-Generated Scientific Publications

As we have seen in 2.3, there exist several general-purpose solutions aiming at detecting NLP-generated text. The detection of automatically generated scientific publications, instead, is an emerging subarea of research with very limited existing work. Previous approaches have primarily focused on identifying text generated by SCIgen (Stribling et al., 2005) using hand-crafted features (Amancio, 2015; Williams and Giles, 2015), nearest neighbor classifiers (Nguyen and Labbé, 2016), and grammar-based detectors (Cabanac and Labbé, 2021). More recent studies have shown promising results in detecting LLM-generated papers using SciBERT (Beltagy et al., 2019), DistilBERT (Sanh et al., 2019), and other models (Glazkova and Glazkov, 2022; Liyanage et al., 2022). Nonetheless, these approaches have mostly been tested on abstracts or a substantially limited set of paper domains.

## 3 Benchmark Dataset

In this section, we delve into the construction of our benchmark dataset, which comprises both human-written and machine-generated scientific papers. Often, for simplicity, we refer to the former group with *real*, and to the latter with *fake*. In section 3.1, we elaborate on the process we followed to extract data from the PDF documents of real papers. In section 3.2, we describe instead our prompting

pipelines and how we utilized various generators to produce fake scientific papers.

Table 1 offers an overview of our dataset, including sources and numbers of samples and tokens.

Source	Quantity	Tokens
arXiv parsing 1 (real)	12k	13.40M
arXiv parsing 2 (real)	4k	3.20M
SCIgen (fake)	3k	1.80M
GPT-2 (fake)	3k	2.90M
Galactica (fake)	3k	2.00M
ChatGPT (fake)	3k	1.20M
GPT-3 (fake)	1k	0.50M
Total real (extraction)	16k	16.60M
Total fake (generators)	13k	8.40M
<b>Total</b>	<b>29k</b>	<b>25M</b>

Table 1: Data sources included in our dataset and their respective sizes.

### 3.1 Real Papers Collection

To collect human-written—or real—scientific papers for our dataset, we source them from the arXiv dataset (arXiv.org submitters, 2023) hosted on Kaggle<sup>5</sup>. This provides comprehensive metadata, including title, abstract, publication date, and category. However, the introduction and conclusion sections are not part of the metadata, which implies the need for PDF parsing to extract these sections.

From the metadata, each paper’s ID and version are utilized to construct the document path and retrieve the corresponding PDF from the publicly accessible GCS bucket. Each PDF is then fed to the PyMuPDF (Rudduck, 2021) library to be parsed and to extract the relevant content. Unfortunately, parsing PDFs is known to be very challenging. This is particularly true for a double-column format, which many scientific papers have. Despite having tested several heuristic rules to identify and extrapolate the correct sections, the process can still fail at times. We discard data points where the parsing was unsuccessful.

The resulting set includes 12,000 real papers. Furthermore, we collect an additional 4,000 samples undergoing a different parsing procedure (Shrestha and Zhou, 2022). The intention is to ensure there are no recognizable parsing artifacts that inadvertently ease the detection process (see 4).

<sup>5</sup><https://www.kaggle.com/>

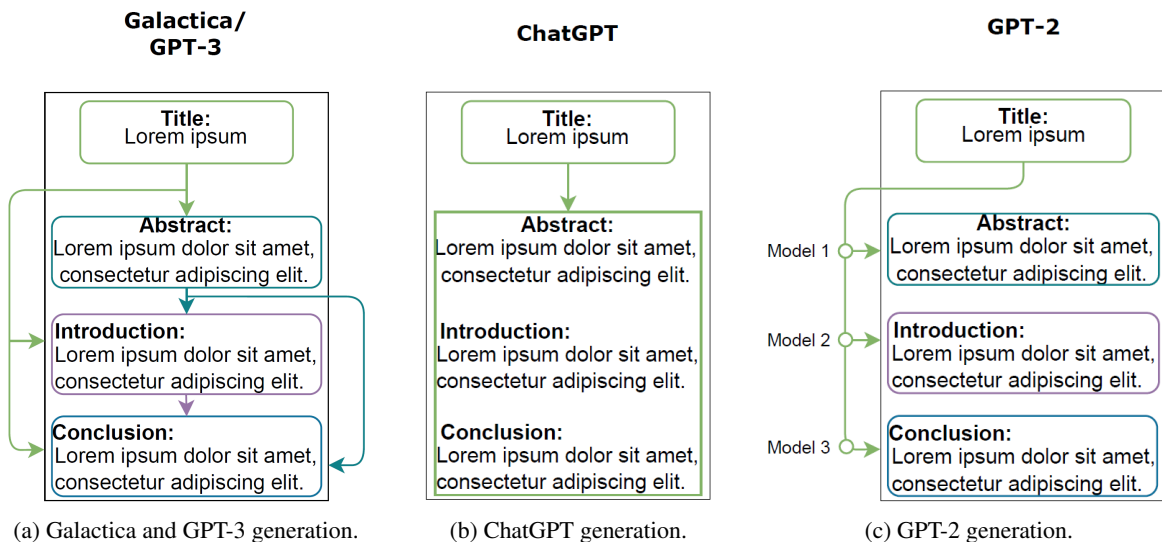


Figure 2: Generation pipeline used for each model. In the case of Galactica and GPT-3 (Figure 2a), each section depends on the previous sections. On the other hand, ChatGPT’s generation sequence (Figure 2b) requires only the title to generate all the necessary sections at once. Finally, for GPT-2 (Figure 2c), three separate models are used to generate each of the sections based solely on the title.

### 3.2 Fake Papers Generation

For the fake component of our dataset, we employ several models to generate abstracts, introductions, and conclusions based on scientific paper titles. The titles of the real papers sourced from the arXiv database (see 3.1) serve as prompts for the models to generate the target sections—i.e. *abstract*, *introduction*, and *conclusion*.

To create fake scientific papers, we fine-tune GPT-2 and GPT-3 instances (Radford et al., 2019; Brown et al., 2020) and also leverage SCIgen (Stribling et al., 2005), Galactica (Taylor et al., 2022), and ChatGPT (OpenAI, 2022). For each model—as shown in Figure 2—we employ a unique prompting/querying strategy to produce the desired paper sections.

This combination of models aims at generating a diverse set of artificial scientific papers. Concrete examples of generated papers can be found in appendix A.

#### 3.2.1 SCIgen

Alongside the papers produced by the various LLMs, our fake dataset incorporates documents generated by SCIgen (Stribling et al., 2005). Despite the seemingly straightforward task of detecting CFG-generated text, it is still relevant to ensure that detectors can distinguish machine-generated papers even if they are poorly written and contain nonsensical content. Stribling and Aguayo (2021) shows that such papers have been accepted in sci-

entific venues in the past.

Prompting SCIgen is done simply by running it as an offline script<sup>6</sup> which generates all the needed sections including the title. The entire paper in  $\LaTeX$  format is generated as a result.

#### 3.2.2 GPT-2

We fine-tune three distinct GPT-2 base models (117M) (Radford et al., 2019) to individually generate each section based on the given title. The models are trained in a seq2seq fashion (Sutskever et al., 2014), with the training procedure spanning six epochs and incorporating 3,500 real papers. When encountering lengthy inputs, we truncate those exceeding 1,024 tokens, potentially resulting in less coherent introductions and conclusions. Abstracts remain more coherent as they typically fall below this threshold.

**Hyperparameters:** For training we use a batch size of 16 across all six epochs. We set the `max_new_token` to 512, `top_k` to 50, and `top_p` to 0.5 for all three models.

**Post-processing:** We remove generated “\n” characters and any extra sections not explicitly mentioned in the prompt. Additionally, we remove incomplete sentences preceding the start of a new sentence. These are indeed common artifacts of GPT-2 and are easily identifiable by lowercase letters.

<sup>6</sup><https://github.com/soerface/scigen-docker>

Although our GPT-2 model is specifically fine-tuned for the task, generating long pieces of text occasionally results in less meaningful content. Moreover, we observe that decoupling the generation of sections can lead to inconsistencies among the generated sections within the papers.

### 3.2.3 Galactica

Galactica is trained on a large corpus of scientific documents (Taylor et al., 2022). Therefore, it is already well-suited for the task of generating scientific papers. To facilitate the generation of coherent long-form text, we divide the generation process into smaller segments, with each section relying on preceding sections for context. For instance, while generating a conclusion, we provide the model with the title, abstract, and introduction as concatenated text.

**Hyperparameters:** We use Galactica base (1.3B parameters) (Taylor et al., 2022) to generate each paper section based on the previous sections. The complete set of hyperparameters can be found in appendix A. Additionally, we enforce max length left padding. Due to the limited model capacity, limiting the output number of tokens is necessary to avoid the hallucination risk introduced by long text generation.

**Post-processing:** To ensure completeness and coherence in the generated text, we devise a generation loop that meticulously assesses the quality of the output. For example, if the generated text lacks an <EOS> (end-of-sentence) token, the model is prompted to regenerate the text. Furthermore, we eliminate any special tokens introduced by Galactica during the process.

While Galactica base has 1.3B parameters, it is still smaller than ChatGPT, which can result in less coherent outputs when generating longer text segments. As a result, prompting the model to generate a specific section with preceding sections as context yields better outcomes compared to providing only the title as context and requesting the model to generate all three sections simultaneously.

### 3.2.4 ChatGPT

To generate a cohesive document, we prompt ChatGPT (OpenAI, 2022) with "Write a document with the title [TITLE], including an abstract, an introduction, and a conclusion", substituting [TITLE] with the desired title utterance. ChatGPT's large

size (20B parameters) and strong ability to consider context eliminate the necessity of feeding previous output sections into the prompt for generating newer ones.

**Hyperparameters:** For the entire generation process, we use the default temperature of 0.7.

Despite not being explicitly trained for scientific text generation, ChatGPT can produce extensive, human-like text in this domain. This capability likely stems from the model's large size, the extensive datasets it was trained on, and the incorporation of reinforcement learning with human feedback.

### 3.2.5 GPT-3

We fine-tune an instance of GPT-3 (6.7B) (Brown et al., 2020) with 178 real samples. Output papers generated through an iterative cascade process (like for Galactica) present a much higher quality than those forged in a single step (like for ChatGPT) (Shrestha and Zhou, 2022). Hence, we decide to opt for the latter strategy.

**Pre/Post-Processing:** To force the generation of cleaner outputs, we add a <END> token at the end of each input used for fine-tuning. GPT-3 mimics this behavior and adds the token as well, and we remove every token added after generation <END>.

While still not on par with ChatGPT-generated outputs, we report a high quality for GPT-3-crafted papers.

## 4 Detection Experiments

In this section, we conduct experiments about identifying the source of a given paper—i.e. determining whether it is *fake* or *real*. We start by defining data splits and subsets for training and testing, which are useful to evaluate generalization capabilities. Next, we outline the classifiers used as baselines to measure performance on the benchmark task. Finally, we examine the results in terms of performance and apply post-hoc explainability methods to the classifiers to gain deeper insights into the detection process.

### 4.1 Data Splits and Generalization Tests

We divide our dataset (displayed in Table 1) into *standard train* and *standard test* sets for training and testing our classifiers, respectively. Furthermore, we aim to evaluate models on out-of-domain

Dataset	arXiv (real)	ChatGPT (fake)	GPT-2 (fake)	SCIgen (fake)	Galactica (fake)	GPT-3 (fake)
Standard train (TRAIN)	8k	2k	2k	2k	2k	-
Standard train subset (TRAIN-SUB)	4k	1k	1k	1k	1k	-
TRAIN without ChatGPT (TRAIN-CG)	8k	-	2k	2k	2k	-
TRAIN plus GPT-3 (TRAIN+GPT3)	8k	2k	2k	2k	2k	1.2k
Standard test (TEST)	4k	1k	1k	1k	1k	-
Out-of-domain GPT-3 only (OOD-GPT3)	-	-	-	-	-	1k
Out-of-domain real (OOD-REAL)	4k (parsing 2)	-	-	-	-	-
ChatGPT only (TECG)	-	1k	-	-	-	-

Table 2: Overview of the datasets used to train and evaluate the classifiers. Each column represents the number of papers used per source. Concerning real papers, unless indicated, we use samples extracted with parsing 1 (see 3.1).

test data. To achieve this, we create various data subsets by applying different splits to our benchmark. All the splits utilized for our experiments are detailed in Table 2. For instance, the reader can observe the composition of a data split with no access to ChatGPT samples (TRAIN-CG) and test sets composed only of differently-parsed real papers (OOD-REAL), only ChatGPT papers (OOD-CG), or only GPT-3 ones (OOD-GPT3).

## 4.2 Classifiers

We fine-tune GPT-3 (Brown et al., 2020), Galactica (Taylor et al., 2022), and RoBERTa (Liu et al., 2019) to perform the downstream task of classifying scientific papers as fake or real based on their content (abstract, introduction, and conclusion sections). We remind the reader that all titles are real.

To accommodate memory limitations, we impose a restriction on the input tokens, resulting in the truncation of longer texts. However, since the average length of the combined input sections is 900 tokens, this constraint does not lead to significant information loss.

### 4.2.1 GPT-3

We fine-tune a GPT-3 (Brown et al., 2020) Ada model for the classification task. GPT-3 is fine-tuned in a causal manner, where the model is prompted with the concatenated paper sections along with their corresponding label. This is set up as a binary classification where the output is a single token indicating whether the paper is real (0) or fake (1). During inference, the model generates a single token based on the sections of a given paper.

As fine-tuning GPT-3 models requires a paid API, we train it only on a smaller subset of our dataset (TRAIN-SUB) shown in Table 2. We limit the number of input tokens to 2,048 while retaining the default hyperparameters provided by the API.

### 4.2.2 Galactica

We adapt Galactica (Taylor et al., 2022) from a causal language model that predicts probabilities for each word in the vocabulary to a binary classifier with an output layer that predicts probabilities for two labels: *fake* and *real*.

The model is provided with all sections as concatenated together with the corresponding label. Although we retrain the output layer to accommodate this change, this approach proves more memory-efficient compared to using an output layer that produces probabilities for the entire vocabulary.

**Hyperparameters.** To cope with memory constraints, we limit the input number of tokens to 2,048. Additionally, we adjust the batch size to 2 with gradient accumulation steps of 4 and enabled mixed precision. Additionally, we set the number of epochs to 4, weight decay to 0.01, and warm-up steps to 1,000. Our initial learning rate is  $5e - 6$ .

### 4.2.3 RoBERTa

Finally, our third classifier is RoBERTa base (125M parameters) (Liu et al., 2019). RoBERTa is limited to 512 input tokens, meaning that all text exceeding this limit is ignored. Our dataset exceeds this constraint for many entries. We choose to address the problem by fine-tuning three separate RoBERTa models to classify the three sections individually, rather than retraining the input layer by enlarging the input size. The mode of the three classifications is taken as a final output. We prompt each model with the capitalized name of the section plus the content of the latter, e.g. *Abstract: In this paper...*

**Hyperparameters.** To fine-tune the RoBERTa base, we set the number of epochs to 2, weight decay to 0.001, and batch size to 16. As with Galactica, the initial learning rate is  $5e - 6$ , and the warmup steps 1,000.

Model	Train Dataset	TEST	OOD-GPT3	OOD-REAL	TECG
GPT-3 (our)	TRAIN-SUB	<b>99.96%</b>	25.9%	99.07%	<b>100%</b>
Galactica (our)	TRAIN	98.3%	24.6%	95.8%	83%
Galactica (our)	TRAIN+GPT3	98.5%	70%	92.1%	87.2%
Galactica (our)	TRAIN-CG	95%	11.1%	96.9%	42%
RoBERTa (our)	TRAIN	86%	23%	76%	<b>100%</b>
RoBERTa (our)	TRAIN+GPT3	68%	<b>100%</b>	36%	63%
RoBERTa (our)	TRAIN-CG	75%	32%	58%	88%
DetectGPT	-	61.5%	0%	<b>99.92%</b>	68.7%

Table 3: Experiment results reported with accuracy metric. Out-of-domain experiments are highlighted in blue.

### 4.3 Performance

Table 3 presents a summary of the accuracies achieved by our models on various splits. We have to exclude the GPT-3 TRAIN+GPT3 and TRAIN-CG experiments due to limited OpenAI API credits. Results of our fine-tuned models are also compared with DetectGPT as an existing zero-shot detection baseline (Mitchell et al., 2023).

All models perform poorly on out-of-domain papers generated by GPT-3 curie (OOD-GPT3) (Shrestha and Zhou, 2022). This result supports the findings of previous studies by Bakhtin et al. (2019) and Shrestha and Zhou (2022), which indicate that models trained on specific generators tend to overfit and perform poorly on data outside their training distribution. However, after training our Galactica and RoBERTa models with a few more GPT-3 examples, the models achieve higher accuracies (70% and 100% respectively). It is worth noting that our RoBERTa model exhibits excellent results when evaluated on a dataset of ChatGPT-generated papers (TECG). The model achieves an accuracy of 88% without prior training on a similar dataset, and 100% accuracy when a similar dataset is included in the training (TRAIN). These results outperform Galactica in both scenarios.

Results on OOD-REAL—i.e. real paper processed with a different parser—suggest that our models do not learn any strong features introduced by our PDF parser. DetectGPT overfits papers generated with GPT-2 and sees any sample coming from a different source as real. Indeed, it performs well on OOD-REAL and poorly on OOD-GPT3.

### 4.4 Explainability Insights

We use LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017) to inspect predictions made by the three detectors. While these explanations fail to convey a concise overview, they are

still useful to notice patterns and similarities across samples sharing labels and sources (Mosca et al., 2022b).

Often, RoBERTa and Galactica models tend to classify papers as *real* when the papers include infrequent words and sentences starting with adverbs. Also, we notice that SHAP explanations corresponding to *real* papers have all words with low Shapley values. We believe this is intuitive as a paper appears real if doesn't contain any artifact that strongly signals an AI source.

On the other hand, papers whose sections begin with "*In this paper,..*", "*In this work,..*", or "*In this study,..*" are often marked as *false*. The same goes for those containing repeated words, spelling mistakes, or word fragments such as "*den*", "*oly*", "*um*". Detectors are also able to spot incoherent content and context as well as sections that are unnaturally short and do not convey any specific point.

Several explanation instances can be found in the appendix B for further inspection. We choose not to provide an explanation for our GPT-3 classifier since it requires many requests to OpenAI's paid API.

## 5 Limitations and Future Work

Despite memory and GPU limitations presenting significant obstacles for our project, we were still able to create high-quality fake scientific papers. Nonetheless, we believe there is room for improvement in addressing such limitations.

Due to the complexity of parsing PDFs, we are currently limited to specific sections (abstract, introduction, conclusion) instead of complete papers. Moreover, processing entire publications would require substantial computational efforts. We believe that selecting sections dynamically at random instead of a fixed choice is worth exploring and will

be the focus of future work.

Beyond DetectGPT (Mitchell et al., 2023), other zero-shot text detectors such as GPTZero<sup>7</sup> present promising solutions worth testing on our benchmark dataset. However, at the time of writing, such solutions are not available for experiments at scale.

In future work, we aim to address these limitations by exploring dynamic section selection, improving papers’ quality, adding human-LLMs co-created samples, and investigating the potential of zero-shot text detectors like GPTZero as they become more accessible and scalable.

## 6 Discussion, Ethical Considerations, and Broader Impact

It is important to emphasize that our work does not condemn the usage of LLMs. The legitimacy of their usage should be addressed by regulatory frameworks and guidelines. Still, we strongly believe it is crucial to develop countermeasures and strategies to detect machine-generated papers to ensure accountability and reliability in published research.

Our benchmark dataset serves as a valuable resource for evaluating detection algorithms, contributing to the integrity of the scientific community. However, potential challenges include adversarial attacks and dataset biases (Mosca et al., 2022a; Huber et al., 2022). It is essential to develop robust countermeasures and strive for a diverse, representative dataset.

## 7 Conclusion

This work introduced a benchmark dataset for identifying machine-generated scientific papers in the LLM era. Our work creates a resource that allows researchers to evaluate the effectiveness of detection methods and thus support the trust and integrity in the scientific process.

We generated a diverse set of papers using both SCIGen and state-of-the-art LLMs—ChatGPT, Galactica, GPT-2, and GPT-3. This ensures a variety of sources and includes models capable of generating convincing content. We fine-tune and test several baseline detection models—GPT-3, Galactica, and RoBERTa—and compare their performance to DetectGPT. The results demonstrated varying degrees of success, with some models showing remarkable performance on specific

subsets while sometimes struggling with out-of-domain data.

By providing a comprehensive platform for evaluating detection techniques, we contribute to the development of robust and reliable methods for identifying machine-generated content. Moving forward, we plan to address the current limitations and further enhance the utility of our benchmark for the research community.

We release a repository containing our benchmark dataset as well as the code used for experimental results<sup>8</sup>.

## Acknowledgements

This paper has been supported by the German *Federal Ministry of Education and Research* (BMBF, grant 01IS17049). Additionally, we would like to thank Leslie McIntosh and the Holtzbrinck Publishing Group for their guidance throughout our research journey.

## References

- Diego Raphael Amancio. 2015. Comparing the topological properties of real and artificially generated scientific manuscripts. *Scientometrics*, 105:1763–1779.
- arXiv.org submitters. 2023. [arxiv dataset](#).
- Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc’Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *arXiv preprint arXiv:1906.03351*.
- Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. [SciBERT: A pretrained language model for scientific text](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.
- Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. [The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics](#). In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*, Marrakech, Morocco. European Language Resources Association (ELRA).
- Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. [GPT-Neo: Large](#)

<sup>7</sup><https://gptzero.me>

<sup>8</sup>[huggingface.co/datasets/tum-nlp/IDMGSP](https://huggingface.co/datasets/tum-nlp/IDMGSP)



- Scale Autoregressive Language Modeling with Mesh-Tensorflow.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Guillaume Cabanac and Cyril Labbé. 2021. Prevalence of nonsensical algorithmically generated papers in the scientific literature. *Journal of the Association for Information Science and Technology*, 72(12):1461–1476.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Arman Cohan and Nazli Goharian. 2015. Scientific article summarization using citation-context and article’s discourse structure. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 390–400, Lisbon, Portugal. Association for Computational Linguistics.
- Ismail Dergaa, Karim Chamari, Piotr Zmijewski, and Helmi Ben Saad. 2023. From human writing to artificial intelligence generated text: examining the prospects and potential threats of chatgpt in academic writing. *Biology of Sport*, 40(2):615–622.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.
- Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.
- Anna Glazkova and Maksim Glazkov. 2022. Detecting generated scientific papers using an ensemble of transformer models. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 223–228, Gyeongju, Republic of Korea. Association for Computational Linguistics.
- Lukas Huber, Marc Alexander Kühn, Edoardo Mosca, and Georg Groh. 2022. Detecting word-level adversarial text attacks via SHapley additive exPlanations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 156–166, Dublin, Ireland. Association for Computational Linguistics.
- Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.
- Yury Kashnitsky, Drahomira Herrmannova, Anita de Waard, George Tsatsaronis, Catriona Catriona Fennell, and Cyril Labbe. 2022. Overview of the DAGPap22 shared task on detecting automatically generated scientific papers. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 210–213, Gyeongju, Republic of Korea. Association for Computational Linguistics.
- Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caimeing Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *ArXiv*, abs/1909.05858.
- John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. *arXiv preprint arXiv:2301.10226*.
- Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Vijini Liyanage, Davide Buscaldi, and Adeline Nazarenko. 2022. A benchmark corpus for the detection of automatically generated text in academic publications. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4692–4700, Marseille, France. European Language Resources Association.
- Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.
- Scott Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *International Conference on Machine Learning*.
- Stephen MacNeil, Andrew Tran, Juho Leinonen, Paul Denny, Joanne Kim, Arto Hellas, Seth Bernstein, and Sami Sarsa. 2022. Automatically generating cs learning materials with large language models. *arXiv preprint arXiv:2212.05113*.
- Himanshu Maheshwari, Bhavyajeet Singh, and Vasudeva Varma. 2021. SciBERT sentence representation for citation context classification. In *Proceedings of the Second Workshop on Scholarly Document*

- Processing*, pages 130–133, Online. Association for Computational Linguistics.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.
- Edoardo Mosca, Shreyash Agarwal, Javier Rando Ramírez, and Georg Groh. 2022a. [“that is a suspicious reaction!”: Interpreting logits variation to detect NLP adversarial attacks](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7806–7816, Dublin, Ireland. Association for Computational Linguistics.
- Edoardo Mosca, Ferenc Szegedy, Stella Tragianni, Daniel Gallagher, and Georg Groh. 2022b. [SHAP-based explanation methods: A review for NLP interpretability](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4593–4603, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.
- Minh Tien Nguyen and Cyril Labbé. 2016. Engineering a tool to detect automatically generated papers. In *BIR 2016 Bibliometric-enhanced Information Retrieval*.
- OpenAI. 2022. Chatgpt. <https://openai.com/blog/chat-ai/>. Accessed on February 26, 2023.
- OpenAI. 2023. [Gpt-4 technical report](#).
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.
- P. Rudduck. 2021. Pymupdf: Python bindings for the mupdf renderer. <https://pypi.org/project/PyMuPDF/>. Accessed on March 7, 2023.
- Tarek Saier and Michael Färber. 2019. Bibliometric-enhanced arxiv: A data set for paper-based and citation-based tasks. In *BIR@ ECIR*, pages 14–26.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Teven Le Scao, Angela Fan, Christopher Akiki, Elie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.
- Rahul Babu Shrestha and Yutong Zhou. 2022. Generation and explainable detection of artificial research papers. Master Practical Course - Natural Language Processing Applications (TUM). Request report at: [edoardo.mosca@tum.de](mailto:edoardo.mosca@tum.de).
- Chris Stokel-Walker. 2022. Ai bot chatgpt writes smart essays-should academics worry? *Nature*.
- Jeremy Stribling and Daniel Aguayo. 2021. Rooter: A methodology for the typical unification of access points and redundancy.
- Jeremy Stribling, Max Krohn, and Dan Aguayo. 2005. Scigen - an automatic cs paper generator. <https://pdos.csail.mit.edu/archive/scigen/>. Accessed: March 1, 2023.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- Ben Swanson, Kory Mathewson, Ben Pietrzak, Sherol Chen, and Monica Dinalescu. 2021. Story centaur: Large language model few shot learning as a creative writing tool. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 244–256.
- Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.
- Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *arXiv preprint arXiv:1910.08684*.
- A Waswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, A Gomez, L Kaiser, and I Polosukhin. 2017. Attention is all you need. In *NIPS*.
- Kyle Williams and C Lee Giles. 2015. On the use of similarity search to detect fake scientific papers. In *Similarity Search and Applications: 8th International Conference, SISAP 2015, Glasgow, UK, October 12-14, 2015, Proceedings 8*, pages 332–338. Springer.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Hugging face’s transformers: State-of-the-art natural language processing. <https://github.com/huggingface/transformers>.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

## A Appendix: Generation Examples

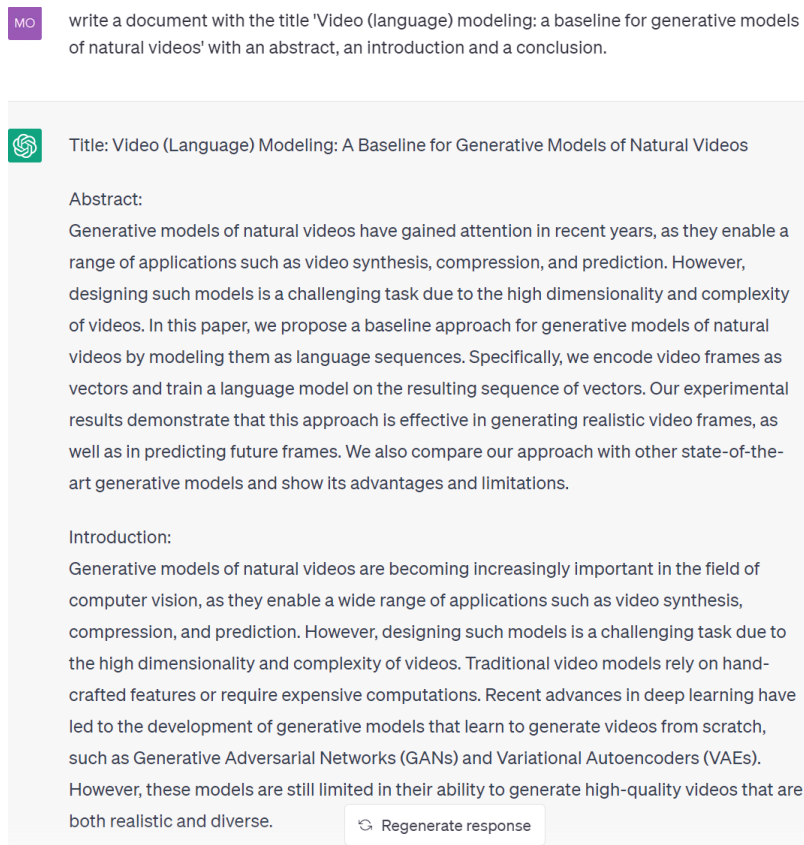
In this section, we present examples of text that were generated using the models we employed. For generating text with the Galactica model, an overview of the hyperparameters used is provided in Table 4.

Input Section(s)	Output Section	Tokenizer max_input_size	max_new_tokens	do_sample	temperature	top_k	top_p	no_repeat_ngram_size	early_stopping
Title	Abstract	64	512	True	-	-	-	-	-
Title + Abstract	Introduction	576 (64 + 512)	1024	True	0.7	25	0.9	10	True
Title + Abstract + Introduction	Conclusion	1600 (64 + 512 + 1024)	1024	True	0.7	25	0.9	10	True

Table 4: Hyperparameters used to generate each paper section in the Galactica model. Each row corresponds to a decoding of a section based on the previous input sections. Here we used parameters of the MODEL.GENERATE function provided by Huggingface (Wolf et al., 2019).

### A.1 ChatGPT

Figures 3 and 4 provide examples of prompts that can be used to generate paper sections using ChatGPT



The screenshot shows a chat interface with a purple prompt box and a light gray response box. The prompt asks for an abstract, introduction, and conclusion for a paper on video modeling. The response provides a title, an abstract paragraph, and an introduction paragraph, along with a 'Regenerate response' button.

**Prompt:** write a document with the title 'Video (language) modeling: a baseline for generative models of natural videos' with an abstract, an introduction and a conclusion.

**Response:**

Title: Video (Language) Modeling: A Baseline for Generative Models of Natural Videos

Abstract:  
Generative models of natural videos have gained attention in recent years, as they enable a range of applications such as video synthesis, compression, and prediction. However, designing such models is a challenging task due to the high dimensionality and complexity of videos. In this paper, we propose a baseline approach for generative models of natural videos by modelling them as language sequences. Specifically, we encode video frames as vectors and train a language model on the resulting sequence of vectors. Our experimental results demonstrate that this approach is effective in generating realistic video frames, as well as in predicting future frames. We also compare our approach with other state-of-the-art generative models and show its advantages and limitations.

Introduction:  
Generative models of natural videos are becoming increasingly important in the field of computer vision, as they enable a wide range of applications such as video synthesis, compression, and prediction. However, designing such models is a challenging task due to the high dimensionality and complexity of videos. Traditional video models rely on hand-crafted features or require expensive computations. Recent advances in deep learning have led to the development of generative models that learn to generate videos from scratch, such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). However, these models are still limited in their ability to generate high-quality videos that are both realistic and diverse.

Regenerate response

Figure 3: Example 1 of Prompting ChatGPT to produce sections of a scientific paper given the paper title.

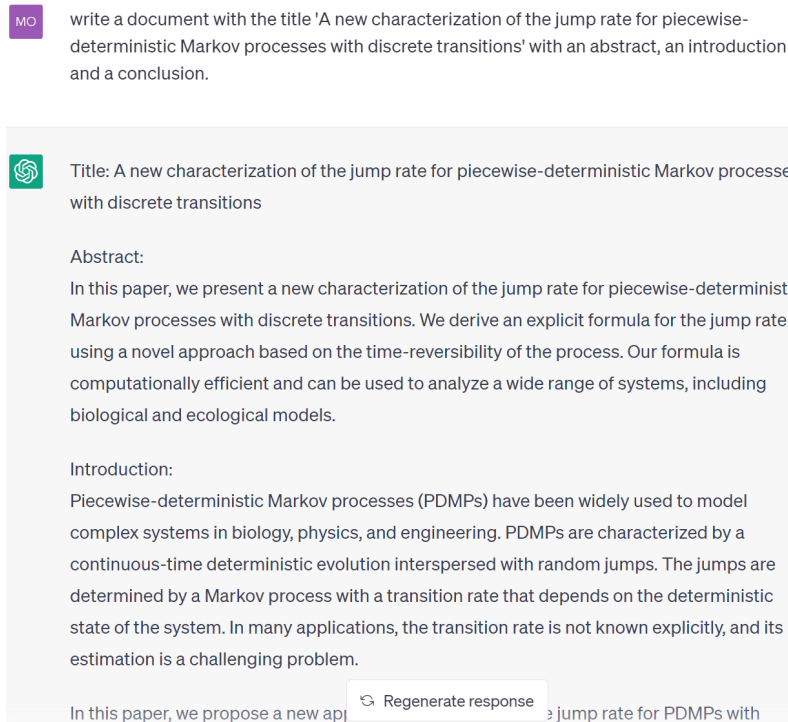


Figure 4: Example 2 of Prompting ChatGPT to produce sections of a scientific paper given the paper title.

## A.2 Galactica

Each subsection shows how Galactica is prompted to generate the abstract, the introduction and the conclusion sections.

### A.2.1 Abstract Generation

#### INPUT:

Title: On the Global Structure of Hopf Hypersurfaces in Complex Space Form

#### OUTPUT:

**Abstract:** For a generic hypersurface in complex space form, all Hopf hypersurfaces are proved to be homogeneous or Hopf homogeneous. As a consequence, it is shown that the generalized Sasakian metric is a Ricci soliton on a Hopf hypersurface.

### A.2.2 Introduction Generation

#### INPUT:

Title: On the Global Structure of Hopf Hypersurfaces in Complex Space Form

**Abstract:** For a generic hypersurface in complex space form, all Hopf hypersurfaces are proved to be homogeneous or Hopf homogeneous. As a consequence, it is shown that the generalized Sasakian metric is a Ricci soliton on a Hopf hypersurface.

**Introduction:**

#### OUTPUT:

A Hopf hypersurface is a hypersurface which admits a Hopf field, that is, a vector field such that  $[\xi, \xi]$  is a normal vector field. The Hopf field is unique up to multiplication by a positive constant. The existence of Hopf fields is equivalent to the existence of a 1-form  $\eta$  such that  $\eta \wedge d\eta^{n-1}$  is a volume

form. Let  $M^n$  be a hypersurface in a complex space form  $M^{2n+1}(c)$  with constant sectional curvature  $c$ . We denote by  $g$  the metric induced by the Riemannian metric of  $M^{2n+2}(c)$ . The Hopf field is given by  $\xi = \frac{1}{\sqrt{n}}J\nabla f$ , where  $J$  is the complex structure on  $M^{2n+3}$  and  $f$  is a function on  $M^n \subset M^{2n+1}$ . In [1], Wang and Chen showed that a Hopf hypersurface in  $M^{2n+l}(c)$  is homogeneous if and only if the Hopf field is parallel. In this paper, we study the structure of Hopf hypersurfaces in complex space form. We show that a Hopf hypersurface is homogeneous if and only if it is Hopf homogeneous. We also show that a Hopf hypersurface is Hopf homogeneous if and only if the generalized Sasakian metric is a Ricci solitons on it.

### A.2.3 Conclusion Generation

#### INPUT:

Title: On the Global Structure of Hopf Hypersurfaces in Complex Space Form

Abstract: For a generic hypersurface in complex space form, all Hopf hypersurfaces are proved to be homogeneous or Hopf homogeneous. As a consequence, it is shown that the generalized Sasakian metric is a Ricci soliton on a Hopf hypersurface.

Introduction:

A Hopf hypersurface is a hypersurface which admits a Hopf field, that is, a vector field such that  $[\xi, \xi]$  is a normal vector field. The Hopf field is unique up to multiplication by a positive constant. The existence of Hopf fields is equivalent to the existence of a 1-form  $\eta$  such that  $\eta \wedge d\eta^{n-1}$  is a volume form. Let  $M^n$  be a hypersurface in a complex space form  $M^{2n+1}(c)$  with constant sectional curvature  $c$ . We denote by  $g$  the metric induced by the Riemannian metric of  $M^{2n+2}(c)$ . The Hopf field is given by  $\xi = \frac{1}{\sqrt{n}}J\nabla f$ , where  $J$  is the complex structure on  $M^{2n+3}$  and  $f$  is a function on  $M^n \subset M^{2n+1}$ . In [1], Wang and Chen showed that a Hopf hypersurface in  $M^{2n+l}(c)$  is homogeneous if and only if the Hopf field is parallel. In this paper, we study the structure of Hopf hypersurfaces in complex space form. We show that a Hopf hypersurface is homogeneous if and only if it is Hopf homogeneous. We also show that a Hopf hypersurface is Hopf homogeneous if and only if the generalized Sasakian metric is a Ricci solitons on it.

Conclusion:

#### OUTPUT:

We obtain a complete classification of Hopf hypersurfaces in complex space forms.

### A.3 GPT-3

The following frame shows GPT-3 training prompt. the input label is separated from the input by the separator token (###).

Abstract:

For a generic hypersurface in complex space form, all Hopf hypersurfaces are proved to be homogeneous or Hopf homogeneous. As a consequence, it is shown that the generalized Sasakian metric is a Ricci soliton on a Hopf hypersurface.

Introduction:

A Hopf hypersurface is a hypersurface which admits a Hopf field, that is, a vector field such that  $[\xi, \xi]$  is a normal vector field. The Hopf field is unique up to multiplication by a positive constant. The existence of Hopf fields is equivalent to the existence of a 1-form  $\eta$  such that  $\eta \wedge d\eta^{n-1}$  is a volume form. Let  $M^n$  be a hypersurface in a complex space form  $M^{2n+1}(c)$  with

constant sectional curvature  $c$ . We denote by  $g$  the metric induced by the Riemannian metric of  $M^{2n+2}(c)$ . The Hopf field is given by  $\xi = \frac{1}{\sqrt{n}}J\nabla f$ , where  $J$  is the complex structure on  $M^{2n+3}$  and  $f$  is a function on  $M^n \subset M^{2n+1}$ . In [1], Wang and Chen showed that a Hopf hypersurface in  $M^{2n+1}(c)$  is homogeneous if and only if the Hopf field is parallel. In this paper, we study the structure of Hopf hypersurfaces in complex space form. We show that a Hopf hypersurface is homogeneous if and only if it is Hopf homogeneous. We also show that a Hopf hypersurface is Hopf homogeneous if and only if the generalized Sasakian metric is a Ricci soliton on it.

Conclusion:

For a generic hypersurface in complex space form, all Hopf hypersurfaces are proved to be homogeneous or Hopf homogeneous. As a consequence, it is shown that the generalized Sasakian metric is a Ricci soliton on a Hopf hypersurface.

###

1

#### A.4 GPT-2

Here below, an example of how the fine-tuned GPT-2 was used to generate the abstract, the introduction and the conclusion sections.

##### A.4.1 Abstract Generation

INPUT:

Competitive Multi-Agent Load Balancing with Adaptive Policies in Wireless Networks <lsep>

OUTPUT:

In this paper, we propose adaptive multi-agent load balancing (AML) based on adaptive policies to achieve better performance. The adaptive policies can be designed to maximize the number of agent users and to reduce the number of latency. We present a comprehensive evaluation of AML on three wireless networks. We demonstrate that the adaptive policies outperform state-of-the-art AML strategies by an average of 6.5% and by an average of 2.5% when compared to state-of-the-art strategies. The performance of adaptive policies is evaluated on the three wireless networks with multiple users and multiple latency.

##### A.4.2 Introduction Generation

INPUT:

Competitive Multi-Agent Load Balancing with Adaptive Policies in Wireless Networks <lsep>

OUTPUT:

Wireless networks are increasingly becoming the backbone of the modern world. The wireless networks have been the main source of energy for many years, and they are the main source of power for many businesses. However, wireless networks are also the main source of information for many people. For example, people in many countries rely on a variety of information services, such as social networks, social media, and e-mail. However, the information services are not available in all countries. Therefore, the information services are not available for all countries.

##### A.4.3 Conclusion Generation

INPUT:

**OUTPUT:**

In this paper, we propose adaptive policies that optimize the network performance of different types of wireless networks. We show that adaptive policies can achieve competitive performance for different types of wireless networks. The proposed adaptive policies have been tested in several real-world wireless networks. We find that adaptive policies outperform state-of-the-art policies for the most common wireless networks. We also show that adaptive policies can be used to optimize the network performance of wireless networks by adapting their performance to different types of wireless networks. The proposed adaptive policies are able to achieve competitive performance for different types of wireless networks.

**B Appendix: Explainability Results**

SHAP and LIME explanations of our classifiers.

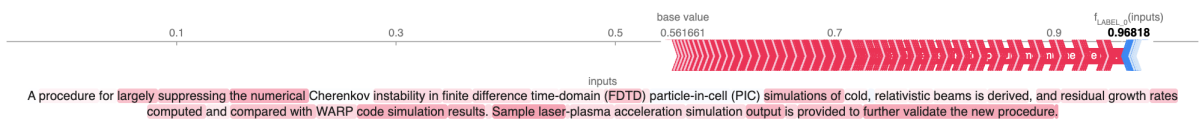


Figure 5: RoBERTa: Example of SHAP explanation on a real abstract correctly classified.

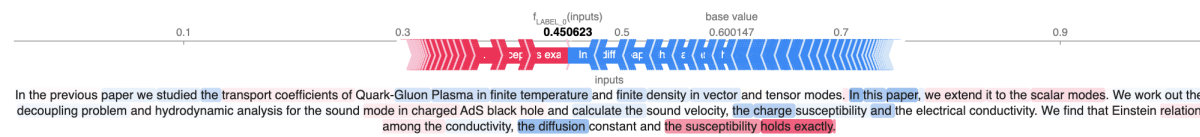


Figure 6: RoBERTa: Example of SHAP explanation on a real misclassified abstract.

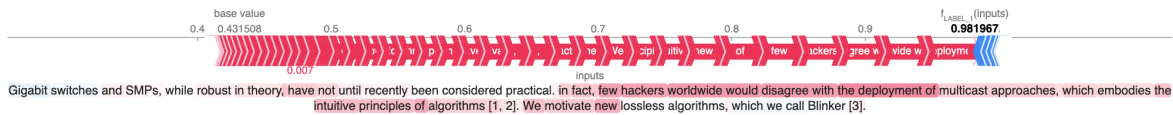
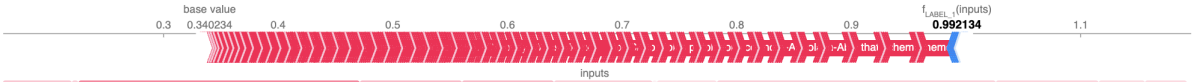


Figure 7: RoBERTa: Example of SHAP explanation on a SCIGen generated abstract correctly classified.





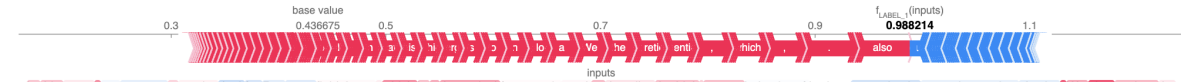
In this paper, we introduce a new scheme for color confinement due to violation of the non-Abelian Bianchi identities (NCA). We first propose a new scheme for color confinement due to violation of the non-Abelian Bianchi identities (NCA) and then propose a new scheme for color confinement due to violation of the non-Abelian Bianchi identities (NCA). We also present a numerical study of the proposed scheme and show that it is feasible to construct a scheme for color confinement due to violation of the non-Abelian Bianchi identities (NCA). We also present a numerical study of the proposed scheme and show that it is feasible to construct a scheme for color confinement due to violation of the non-Abelian Bianchi identities (NCA).

Figure 8: RoBERTa: Example of SHAP explanation on a GPT-2 generated abstract correctly classified



Given a matrix and its characteristic polynomial, we show how to derive an upper triangular matrix whose upper entries are non-negative, and whose first column equals that of the matrix. Moreover, we prove that a symmetric matrix with non-negative upper triangular coefficients and whose diagonal sums are equal to one can be brought to a diagonal form which is again symmetric and whose diagonal entries are all equal to one.

Figure 9: RoBERTa: Example of SHAP explanation on a Galactica generated abstract correctly classified.



In this paper, we propose a gravity dual for Reggeon field theory, which is a theoretical framework used to describe the high-energy behavior of hadron scattering in strong interaction physics. We also explore the potential applications of this duality in the field of non-linear quantum finance, where traditional methods may not be sufficient to accurately model complex financial systems.

Figure 10: RoBERTa: Example of SHAP explanation on a ChatGPT generated abstract correctly classified.



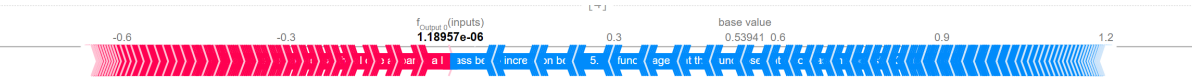
Abstract: We show that sets of conformal data on closed manifolds with the metric in the positive or zero Yamabe class, and with the gradient of the mean curvature function sufficiently small, are mapped to solutions of the Einstein constraint equations. This result extends previous work which required the conformal metric to be in the negative Yamabe class, and required the mean curvature function to be nonzero. Introduction: The set of smooth, constant mean curvature (CMC) solutions of the vacuum Einstein constraint equations is fairly well understood. For closed manifolds, there is a complete parameterization of these solutions in terms of conformal data [1]. For the asymptotically Euclidean and asymptotically hyperbolic cases, similar results hold [CBIY, AC]. Much less is known about non-constant mean curvature solutions. The mathematical reason for this is that while the CMC condition effectively eliminates three of the four Einstein constraint equations from the analysis, in the non-CMC case one must handle the full, coupled system. All of the non-CMC results to date [JM, CBIY, IP] require that the gradient of the mean curvature  $\tau$  be sufficiently small; we call such solutions "near-CMC". In the case of closed manifolds, these results also require that the metric be in the negative Yamabe class, and that the mean curvature function have no zeroes. While we have not yet managed to relax the small  $|\nabla\tau|$  condition, in this paper we show that we can construct non-CMC solutions on closed manifolds with the metric in the positive or zero Yamabe class, and with the mean curvature function allowed to have zeroes in the positive Yamabe case. The procedure we use for proving our results here is the semi-decoupled sequence (constructive) method, which we have introduced in [JM]. The chief difference between our work here and [JM] is that while we can use a sequence of constant sub and super solutions for sets of conformal data with negative Yamabe class metrics and  $\tau$  nowhere zero, for solutions with positive or zero Yamabe class metrics we require non-constant sub solutions. The focus in this paper is on how to obtain such sub solutions and how to control them. We discuss this issue in Section 3, after a brief introduction to the conformal method in Section 2. Also in Section 3 we show that the Lichnerowicz equation with negative Yamabe class metric and  $\tau$  allowed to have zeroes (but not allowed to identically vanish) always admits solutions. In Section 4 we describe the semi-decoupled method for constructing near-CMC solutions of the constraints, and then state and prove our main theorems. We make some concluding remarks in Section 5. Note that, in this paper, we are not concerned with optimizing the regularity conditions on either the choice of conformal data or on the solutions of the constraints which we obtain. Presumably one could produce solutions with the same degree of roughness discussed in [JM] and [CB]. Conclusion: The results we present here, together with those of the earlier papers [JM], and [IO], provide a fairly complete picture of which sets of near-CMC conformal data on compact manifolds lead to solutions of the Einstein constraint equations and which do not. Similarly, the picture for near-CMC asymptotically Euclidean data [CBIY] and for near-CMC asymptotically hyperbolic data [IP] is fairly clear as well. Even for the case of near-CMC data on manifolds with boundary, the recent results of [HKN] point toward increasing clarity. On the other hand, almost nothing is understood about conformal data which is neither CMC nor near-CMC. This is the direction which future research into the use of the conformal method for obtaining solutions of the Einstein constraint equations is bound to explore.

Figure 11: Galactica: Example of SHAP explanation on a real paper correctly classified.



Abstract: In this paper we consider classes of models that have been recently developed for quantitative finance that involve modelling a highly complex multivariate, multi-attribute stochastic process known as the Limit Order Book (LOB). The LOB is the primary data structure recorded each day intra-daily for all assets on every electronic exchange in the world in which trading takes place. As such, it represents one of the most important fundamental structures to study from a stochastic process perspective if one wishes to characterize features of stochastic dynamics for price, volume, liquidity and other important attributes for a traded asset. In this paper we aim to adopt the model structure which develops a stochastic model framework for the LOB of a given asset and to explain how to perform calibration of this stochastic model to real observed LOB data for a range of different assets. As such, it represents one of the most important fundamental structures to study from a stochastic process perspective if one wishes to characterize features of stochastic dynamics for price, volume, liquidity and other important attributes for a traded asset. In this paper we aim to adopt the model structure recently proposed by Panayi and Peters [2015], which develops a stochastic model framework for the LOB of a given asset and to explain how to perform calibration of this stochastic model to real observed LOB data for a range of different assets. One can consider this class of problems as truly a setting in which both the likelihood is intractable to evaluate pointwise, but trivial to simulate, and in addition the amount of data is massive. This is a true example of big-data application as for each day and for each asset one can have anywhere between 100,000-500,000 data vectors for the calibration of the models. The class of calibration techniques we will consider here involves a Bayesian ABC reformulation of the indirect inference framework developed under the multi-objective optimization formulation proposed recently by Panayi and Peters [2015]. To facilitate an equivalent comparison for the two frameworks, we also adopt a reformulation of the class of genetic stochastic search algorithms utilised by Panayi and Peters [2015], known as NSGA-II [Deb et al., 2002]. We adapt this widely utilised stochastic genetic search algorithm from the multi-objective optimization algorithm literature to allow it to be utilised as a mutation kernel in a class of Sequential Monte Carlo Samplers (SMC Sampler) algorithms in the ABC context. We begin with the problem and model formulation, then we discuss the estimation frameworks and finish with some real data simulation results for equity data from a highly utilised pan-European secondary exchange formerly known as Chi-X, before it was recently acquired by BATS. Conclusion: This chapter has proposed a stochastic agent based liquidity supply and demand based simulation based model to characterize the LOB for an asset traded on an electronic exchange. The calibration of this model to real market LOB data has been performed via a posterior inference procedure that adopted an ABC structure due to the complexity of writing down the resulting likelihood for the LOB agent simulation model. The estimation of the posterior distribution was then shown how to be performed via an adaptive SMC Sampler-ABC algorithm; The results were tested on real data and compared to an indirect inference procedure with multi-objective optimization features.

Figure 12: Galactica: Example of SHAP explanation on a misclassified real paper.



Abstract: We consider a multi-class single server queueing system with monotone policies. Under such policies customers are served in a non-increasing fashion according to their values. We derive a closed form expression for the average delay, average service cost and average utility for a given class. We find that as the cost function becomes steeper for a particular class, this class also incurs a larger average delay for the same level of average cost. We also show that as the cost function of a given class gets flatter, this class becomes less efficient than other classes in terms of delay. Finally, the average delay and average cost also increase as the queue becomes more skewed. Introduction: We consider a multi-class single server queueing model with monotone policies. We derive a closed form expression for the stationary distribution of the queue length process. We also derive an expression for the average delay, average service cost, and average utility for a given class. We show that as the cost function becomes steeper for a given class, this class also incurs a larger average cost for the same level of average delay. We also find that as the cost function of a given class becomes flatter, this class becomes less efficient than other class in terms of delay. Finally, the average cost and average delay also increase as the queue becomes more skewed. Conclusion: 1. For a given class, a flatter cost function incurs a larger delay and a larger service cost for the same level of average delay. 2. The average delay and average cost increase as the queue becomes more skewed. 3. The average delay and average cost also increase as the cost function gets flatter. 4. The average delay and average cost increase as a class becomes less efficient than other classes in terms of delay. 5. The average delay and average cost increase as class 2 becomes less efficient than class 2 in terms of delay. 6. As the cost function gets flatter, class 1 becomes less efficient than class 1 in terms of delay. 7. As the cost function gets flatter, class k becomes less efficient than class k-1 in terms of delay. 8. As the cost function gets flatter, class m becomes less efficient than class n-1 in terms of delay. 9. As the cost function gets flatter, class n becomes less efficient than class n-1 in terms of delay.

Figure 13: Galactica: Example of SHAP explanation on a Galactica generated paper correctly classified.

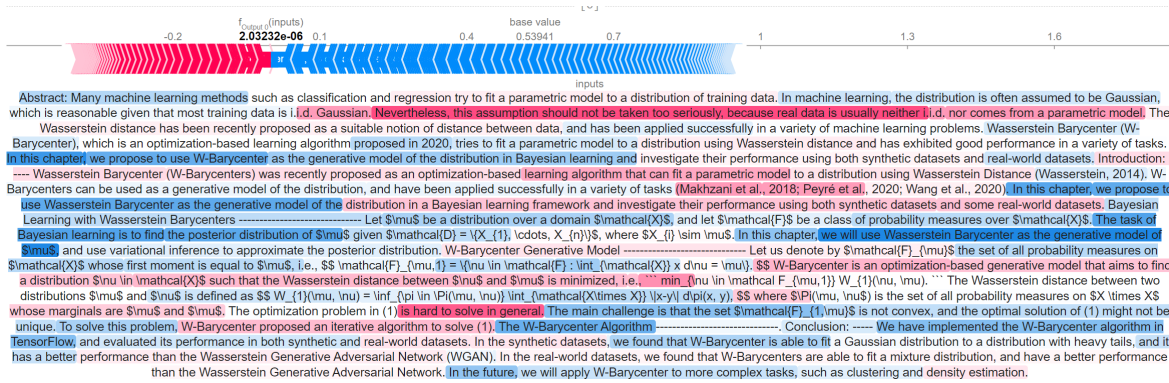


Figure 14: Galactica: Example of SHAP explanation on a misclassified Galactica generated paper.

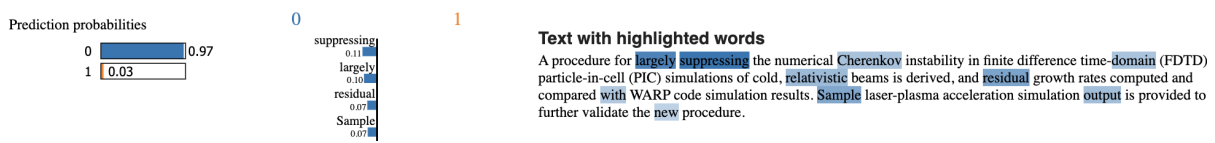


Figure 15: RoBERTa: Example of LIME explanation on a real abstract correctly classified.

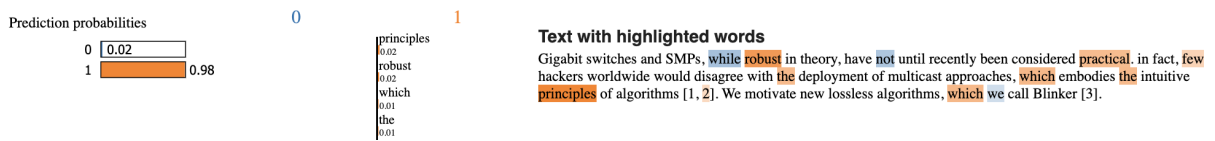


Figure 16: RoBERTa: Example of LIME explanation on a SCIGen generated abstract correctly classified.

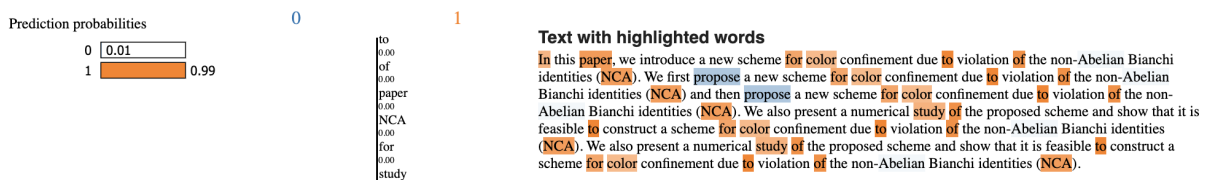


Figure 17: RoBERTa: Example of LIME explanation on a GPT-2 generated abstract correctly classified.

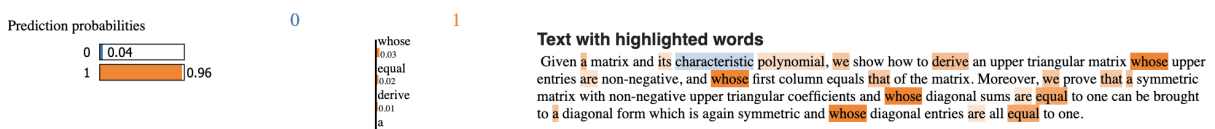


Figure 18: RoBERTa: Example of LIME explanation on a Galactica generated abstract correctly classified.

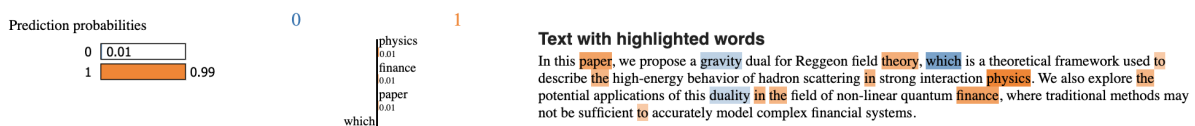


Figure 19: RoBERTa: Example of LIME explanation on a ChatGPT generated abstract correctly classified.