

DEMASQ: Unmasking the ChatGPT Wordsmith

Kavita Kumari

Technical University of Darmstadt
kavita.kumari@trust.tu-darmstadt.de

Alessandro Pegoraro

Technical University of Darmstadt
alessandro.pegoraro@trust.tu-darmstadt.de

Hossein Fereidooni

Technical University of Darmstadt¹
hossein.fereidooni@trust.tu-darmstadt.de

Ahmad-Reza Sadeghi

Technical University of Darmstadt
ahmad.sadeghi@trust.tu-darmstadt.de

Abstract—The potential misuse of ChatGPT and other Large Language Models (LLMs) has raised concerns regarding the dissemination of false information, plagiarism, academic dishonesty, and fraudulent activities. Consequently, distinguishing between AI-generated and human-generated content has emerged as an intriguing research topic. However, current text detection methods lack precision and are often restricted to specific tasks or domains, making them inadequate for identifying content generated by ChatGPT.

In this paper, we propose an effective ChatGPT detector named *DEMASQ*, which accurately identifies ChatGPT-generated content. Our method addresses two critical factors: (i) the distinct biases in text composition observed in human- and machine-generated content and (ii) the alterations made by humans to evade previous detection methods. *DEMASQ* is an energy-based detection model that incorporates novel aspects, such as (i) optimization inspired by the Doppler effect to capture the interdependence between input text embeddings and output labels, and (ii) the use of explainable AI techniques to generate diverse perturbations. To evaluate our detector, we create a benchmark dataset comprising a mixture of prompts from both ChatGPT and humans, encompassing domains such as medical, open Q&A, finance, wiki, and reddit. Our evaluation demonstrates that *DEMASQ* achieves high accuracy in identifying content generated by ChatGPT.

I. INTRODUCTION

The release of ChatGPT, an AI chatbot developed by OpenAI, has generated significant interest and extensive discussions within the Natural Language Processing (NLP) community and various other domains. ChatGPT, which was launched in November 2022, harnesses the capabilities of OpenAI’s language models from the GPT-3.5 and GPT-4 series. By employing supervised and reinforcement learning techniques based on Human Feedback [10], [32], ChatGPT generates sophisticated responses to diverse queries across various NLP fields, as highlighted in recent studies [7], [16], [38], [52], [58].

The media’s promotion of ChatGPT has generated various responses. News and media companies are employing it to enhance content creation, educators and academics are utilizing

it to support course objectives, and individuals are taking advantage of its language translation capabilities. Unfortunately, advanced technologies like ChatGPT often face instances of misuse. Students are using it to generate assignments and coding projects [8], [12], scholars rely on it for producing academic papers [20], and malicious actors exploit it to spread fake news on social media platforms [24], [13]. Furthermore, ChatGPT has the potential to create seemingly realistic stories that can deceive unsuspecting readers [6], [48]. Therefore, the development of an effective detection algorithm capable of distinguishing AI-generated text, particularly from ChatGPT, from human-generated text has become a significant focus for researchers.

When detecting AI-generated text using machine learning, two common approaches are black-box and white-box detection. Black-box detection involves accessing language models through APIs, posing challenges in identifying synthetic texts effectively [53]. It includes tasks like data collection, feature extraction, and constructing specialized classifiers, often using simpler ones like binary logistic regression [50]. In contrast, white-box detection has unrestricted access to language models, providing greater control over behavior and traceable results [53]. Zero-shot detection is an example of white-box detection, using pre-trained generative models like GPT-2 or Grover, along with fine-tuned language models for the detection task [36], [50], [59].

The development of text detectors has attracted substantial attention in recent research, with considerable efforts devoted to developing detectors capable of distinguishing text generated by AI bots [12], [20], [21], [22], [28], [30], [31], [36], [43], [50], [59], [60]. Moreover, certain assertions have been made concerning the discriminative ability of AI-text detectors in distinguishing ChatGPT-generated text from human-generated text [41], [4], [8], [14], [17], [18], [19], [23], [37], [39], [55], [56].

Our paper’s primary focus revolves around the development of detectors to effectively distinguish between text generated by ChatGPT and humans. In our pursuit of this goal, we conducted a comprehensive evaluation of existing schemes and tools dedicated to the detection of ChatGPT-generated

¹The author worked on this project while being affiliated with TU Darmstadt but is now at KOBIL GmbH.

content. Our evaluation of the existing ChatGPT detectors has revealed that, in the best-case scenario, these methods achieved an accuracy rate of 47%, as elaborated in Table I.

Our goals and contributions. We address the limitations of existing ChatGPT detectors and present our novel approach, inspired by the intricate nature of human communication. Human interactions exhibit diverse patterns and inherent biases, influenced by personal backgrounds, environmental factors, and the unique individuals involved. In contrast, machines interact based on predefined rules, maintaining consistent energy and frequency regardless of the human query. To account for this inherent bias, we leverage the Doppler effect to model the communication dynamics between humans and machines. This novel approach allows us to quantify the responsiveness of both humans and machines to questions posed by humans, thus providing a unique perspective on bias detection.

We extend the Doppler effect’s application to text, hypothesizing that thoughts translated into text follow a similar physical process as phonetic energy waves in human communication. And by adopting drumhead vibrations, we model waves and source frequency. Since the energy level is directly linked to its higher or lower frequency, we leverage this relationship and choose to develop and train an Energy-Based Model (EBM). In EBMs, energy serves as a key metric for understanding the connection between the model’s input and output, with lower energy configurations representing the most optimal relationship between these variables. This model facilitates the discrimination between responses generated by humans and ChatGPT through analyzing their respective energy levels, effectively capturing the biases introduced by both entities.

We stress that, no existing ChatGPT detector has explored these aspects, giving our novel design, named *DEMASQ*, a significant performance advantage as it achieves an high accuracy of 97% on a representative benchmark dataset that contains diverse prompts from both ChatGPT and humans, spanning different domains which we detail in Section V-B.

Moreover, we address ChatGPT’s rephrasing techniques by incorporating the explainable AI concept of Integrated Gradients (IG) [51] to perturb the input string’s embedding. This perturbation allows us to extract crucial features, compute their energies, and identify a natural threshold between ChatGPT and human-like responses during the string’s evolution. This integration of energy-based modeling, IG perturbation, and optimization enhances our detector’s ability to differentiate between human and ChatGPT-generated responses while accounting for biases and rephrasing techniques.

In summary, our contributions in this paper include:

- The design and implementation of *DEMASQ*, a novel ChatGPT detector for accurate discrimination between ChatGPT and human-generated text.
- A unique approach that incorporates the effect of bias introduced during human interactions and rephrasing techniques that modify ChatGPT-generated responses to evade detection. We utilize frequency and energy modeling, and explainable AI (cf. Section IV).

- The creation of a comprehensive benchmark dataset comprising over 100,000 samples and spanning various domains, such as wiki, reddit, openqa, finance, medical, arxiv, and political, and the rigorous evaluation of our approach’s robustness across diverse contexts (cf. Section VI). We evaluated the existing ChatGPT detectors on this benchmark, revealing that these methods achieve an accuracy rate of 47% in the best-case scenario (Table I). In contrast, *DEMASQ* achieves an accuracy of 96.5% on the above mentioned combined benchmark dataset, significantly outperforming the existing detectors.
- A new research direction that sheds light on the current state of detection techniques and aims to stimulate further exploration in this critical field.

While *DEMASQ* demonstrates a significant superiority over the existing ChatGPT detectors, we acknowledge that there are still several challenges to be tackled. Firstly, we expect that *DEMASQ*, given its generic design, will exhibit comparable performance when subjected to prompts generated by other generative language models. Hence, we are currently investigating this aspect as part of our ongoing research efforts. Secondly, although we have demonstrated the superior performance of *DEMASQ* when compared to recent work on hybrid texts (as discussed in Section VI-D), we are actively refining our methodology to bolster its effectiveness in distinguishing content generated by ChatGPT, especially when confronted with rephrased or adversarial prompts.

II. BACKGROUND

In this section, we present a concise overview of the essential background knowledge encompassing all the related concepts necessary to comprehend the design of our proposed detection method. By providing this contextual information, we aim to equip readers with the foundational understanding required to grasp the intricacies and rationale behind our innovative detection approach.

A. Large Language Models (LLMs)

Large Language Models (LLMs) are advanced AI systems designed to understand and generate human-like text. They undergo extensive training on diverse datasets from various sources like books, articles, and websites. This training enables LLMs to produce coherent and contextually relevant responses. Examples of notable LLMs include GPT-3, GPT-2, Transformer-XL, BERT, and ChatGPT. These models represent just a fraction of the diverse range of LLMs developed by different organizations and researchers, each with its own strengths and areas of specialization.

B. Doppler Effect

The Doppler effect occurs when the relative motion between a wave source and an observer exists. It manifests as a change in the source frequency as perceived by the observer. When a wave source moves toward an observer, there is an apparent increase in frequency, while the frequency decreases when the source moves away from the observer. This relationship

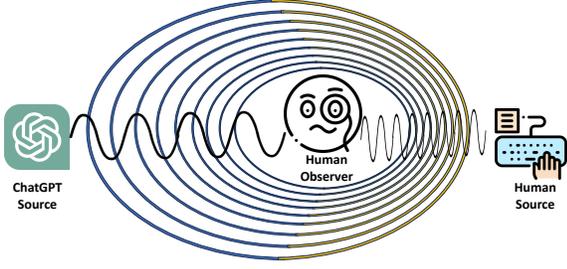


Fig. 1: Doppler effect.

between the observed frequency, denoted as E_f , and the emitted frequency, denoted as E_{f_0} , can be described by the following equation:

$$E_f = \left(\frac{c_v \pm v_r}{c_v \pm v_s} \right) E_{f_0} \quad (1)$$

In the above equation, c_v represents the propagation speed of waves in the medium. The term v_r denotes the receiver's speed relative to the medium. If the receiver moves towards the source, v_r is added to c_v ; however, if the receiver moves away from the source, v_r is subtracted. Similarly, v_s represents the speed of the source relative to the medium. If the source moves away from the receiver, v_s is added to c_v , and v_s is subtracted if the source is moving towards the receiver.

In this work, we conceptualize humans (humans generating the response) or ChatGPT (ChatGPT generating the response) as sources and consider another human as the observer, as shown in Figure 1. The medium represents the space which vibrates or vibrates with the strength of the human factor in the input text embedding. Alternatively, the medium represents a space in which ChatGPT tries to imitate content similar to human-generated content. More detailed explanation is given in Section IV-B.

C. Energy-based model

Statistical modeling and machine learning primarily aim to represent relationships between variables. By capturing these relationships, models can provide answers regarding the values of unknown variables based on the known variables. Energy-Based Models (EBMs) [33] assign "scalar values" to represent the compatibility between different variable configurations. These "scalar values" are called "energies" computed using an energy function defined to represent the compatibility between different variable configurations. Instead of focusing on classifying \vec{x} values to corresponding y values (as done in ML), the goal of EBMs is to determine whether a specific pair of (\vec{x}, y) is a good match or not. In other words, the objective is to find a compatible y that fits well with a given \vec{x} . It can be framed as the task of discovering a y for which a certain function $F(\vec{x}, y)$ has a low value, as the lower

energy configurations describe the best relationship between these variables.

We establish an energy function $F : X \times Y \rightarrow \mathbf{R}$ that quantifies the degree of dependency between pairs (\vec{x}, y) . When constructing and training an energy-based model using a training set S , the process entails developing and refining four essential components:

- **The architecture:** The architecture of the EBM refers to the internal arrangement and design of the parameterized energy function $F(w, \vec{x}, y)$ (with parameters w). It encompasses the specific structure and organization of the components within the energy function that are determined by the chosen model's parameters.
- **The inference algorithm:** The method for finding a value of y that minimizes $F(w, \vec{x}, y)$ for any given \vec{x} is given by the following equation:

$$y' = \arg \min_y F(w, x, y)$$

- **The loss function:** $L(w, F)$ quantifies the quality or performance of an energy function by evaluating its compatibility with the training set.
- **The learning algorithm:** The approach to discover a w that minimizes the loss functional across the family of energy functions F , considering the given training set.

In this work, EBM is defined to model the dependencies between the input text embedding ($V(\vec{x})$) and the output label of the text. Label 1 for humans generating the text is assigned high energy, and label 0 for ChatGPT-generated responses is given low energy. The instantiation of all these components is detailed in Section IV-B.

D. Explainable AI

Explainable AI (XAI), also known as interpretable AI or transparent AI, refers to the development and application of artificial intelligence (AI) systems that can effectively explain their reasoning and decision-making processes in a human-understandable manner. This has resulted in several novel model explanation techniques and toolkits [45], [35], [49], [3], [26], [1], [11], [5]. One class of XAI is the attributions method in which each feature contribution to the model function is computed. Mathematically, it is defined as: given an input data point $\vec{x} \in \mathbb{R}^n$ and a classification model F_c , an explanation method denoted as \mathcal{H} is employed to provide an influence or attribution vector. This vector elucidates the model's decision-making process by revealing the contributions of each feature. The i^{th} element of this vector, denoted as $\mathcal{H}_i(\vec{x})$, represents the impact of the i^{th} feature on the predicted label y for the given data point \vec{x} .

We utilize a specific attribution method, called Integrated Gradients (IG) [51], which is derived by accumulating gradients computed at all points along a linear path from a chosen baseline \vec{x}' (often set as $\vec{x}' = \vec{0}$) to the actual input \vec{x} [51]. Essentially, integrated gradients involve integrating the gradients along a straight-line path from the baseline \vec{x}' to the input \vec{x} . The integrated gradient for a specific feature i

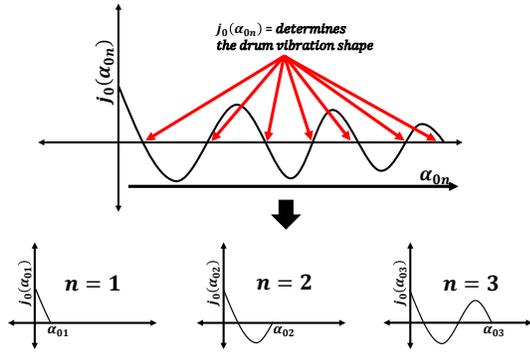


Fig. 2: The impact of different modes of the zeroth Bessel function ($J_0(\alpha_{0n})$) on the frequency with which the drumhead vibrates. For example, we have illustrated how the wave travels (shape of the wave) in a drum with a fixed $m = 0$, and $n = 1$, $n = 2$, $n = 3$.

with an input \vec{x} and baseline \vec{x}' is denoted as $\mathcal{H}_{IGRAD}(\vec{x})_i$, computed as shown in Equation 2.

$$\mathcal{H}_{IGRAD}(\vec{x})_i = (x_i - x'_i) * \int_{\alpha=0}^1 \frac{\partial F_c(\vec{x}' + \alpha(\vec{x} - \vec{x}'))}{\partial x_i} d\alpha \quad (2)$$

In this work, we utilize IG to account for ChatGPT’s rephrasing techniques that a human can request to evade detection from different ChatGPT detectors present in the literature. Specifically, we employ IG to compute different perturbations of the input text embedding ($V(\vec{x})$) to extract crucial features. Then, we compute the energies of the perturbations by systematically removing these features, generating additional samples with corresponding labels, and re-optimizing the loss function. Integrating this in our proposed scheme efficiently improves our detection of ChatGPT-generated responses.

E. Drumhead vibrations

The behavior of an idealized drumhead can be understood by examining the vibrations of a two-dimensional elastic membrane under tension [47] [27]. This membrane, which can be modeled as a circular surface with uniform thickness connected to a rigid frame, exhibits fascinating properties. The membrane can store vibrational energy at specific resonant frequencies through the resonance phenomenon. The surface of the membrane moves in distinct patterns characterized by standing waves, referred to as normal modes. The lowest frequency normal mode is called the fundamental mode, and the membrane possesses an infinite number of these normal modes. Specifically, there are two modes (that can have diverse values) of vibration in the membrane: angular mode (m) and circular mode (n). The membrane can vibrate in countless ways, with each mode determined by the initial shape of the membrane and the transverse velocity at each point on the surface. The membrane’s vibrations can be described mathematically by solving the two-dimensional wave equation, subject to Dirichlet boundary conditions that represent the constraints imposed by the frame [54]. This equation captures the intricate dynamics of the membrane and its interactions

with the surrounding medium. The solution of this equation for asymmetric case ($m = 0, n$), is given by:

$$D_H(r, t) = \sum_{n=1}^{\infty} J_0(\lambda_n r) * (A \cos(c_d \lambda_n t) + B \sin(c_d \lambda_n t)) \quad (3)$$

$D_H(r, t)$ represents the height of the drum head at a specific radial coordinate r , measured relative to the “still” drum head shape at time t . Here, we have an open disk with a radius of a , where the value of D_H is zero on the boundary. c_d^2 determines the wave speed in the membrane. In Equation 3 we use the root: $\sqrt{c_d^2} = c_d$. The coefficients A and B are determined by the initial conditions.

$J_0(\lambda_n r)$ is the zeroth Bessel function with angular mode $m = 0$, and circular mode n , and it represents the strength of the frequency computed at different modes for each $(0, n)$ value. The specific values of $\lambda_n r = \alpha_n$ are referred to as the zeros of the Bessel function J_0 [9] and are used to determine the shape of the drum that determines the frequency with which it vibrates, as shown in Figure 2.

In this work, we utilize drumhead vibration to compute the source frequency (human or ChatGPT) (E_{f_0}) for the computation of observed frequency by an observer (human) in the Doppler effect. We compute the zeros of the Bessel function $J_0(\alpha_{0n})$ (using the Python SciPy library [15]), which in turn is used to compute the source frequency. Frequency at mode $m = 0, n$ can be computed as $\alpha_{0n} \times c_d/a$, denoted as F_{J_0} . Since, we are unaware of the values of c_d and a , we compute $E_{f_0} = \frac{F_{J_0(0,n)}}{F_{J_0(0,1)}}$ to normalize the computation of E_{f_0} with fundamental frequency at $J_0(0,1)$. Thus, we only need to measure n to compute the source frequency (detailed in Section IV). Also, we specifically concentrate on the circular mode of vibration. This choice is based on our assumption that the “Doppler waves” related to both human-generated and ChatGPT-generated content propagate in a linear fashion, or to put it differently, we assume that the source’s waveform moves along the line of the observer’s sight.

III. THREAT MODEL

We investigate a situation where the detector or an observer (a human) functions as a black box and relies exclusively on the observed samples. However, the detector has no privileged knowledge about the underlying ChatGPT Large Language Model (LLM) responsible for generating these samples, including details such as its weights, structures, and gradients. This assumption accurately reflects the current situation, as OpenAI has not made the LLM utilized in the GPT-3.5 family publicly accessible, thereby limiting insider information. By considering this realistic scenario, we aim to explore the challenges and potential strategies of detectors against attacks in the absence of in-depth knowledge about the LLM.

We make the following assumptions regarding the attacker: Firstly, we assume that the attacker is a human who actively aims to avoid detection. Secondly, the attacker has knowledge of the deployed detector. However, the attacker has no information about our deployed detector’s specific

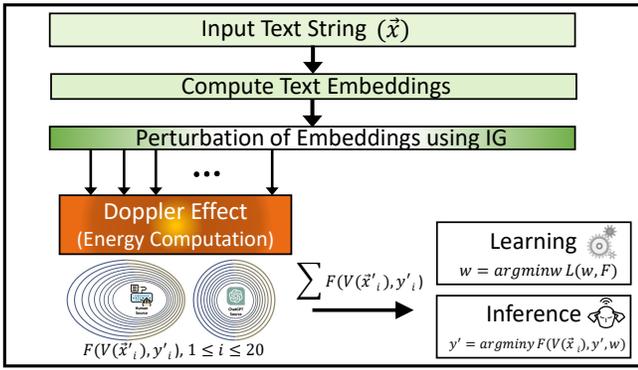


Fig. 3: High level idea of our proposed approach.

structural design, such as using EBM, doppler effect, and XAI. Additionally, we consider the attacker possibly rephrasing the given input string by utilizing ChatGPT multiple times to evade detection. The adversary may also employ other rephrasing techniques to alter the content and make it harder for the detector to identify. These assumptions acknowledge an attacker’s potential strategies to bypass our detection system, prompting us to address the challenges associated with such deceptive practices.

IV. SYSTEM DESIGN

In this section, we provide a comprehensive account of the primary constituents comprising our suggested ChatGPT detector. Initially, we present a broad overview of the concept, followed by a detailed description of its fundamental elements.

A. High-level Idea

Figure 3 represents the high-level architecture of our proposed ChatGPT detector *DEMASQ*.

As mentioned in the introduction, the main idea of our approach is to leverage the bias introduced by the human or ChatGPT in their responses to queries. Alternatively, one can perceive this as the fact that every individual interacts with others in a unique manner [25], [57] and brings biases, such as emotions, personal history with that specific person, and environmental factors, before engaging in communication. Our approach consists of three components: i) an adapted Doppler effect, ii) an energy-based model to capture the dynamics of human and ChatGPT behavior, and iii) an Integrated Gradient (IG) method to capture hybrid texts that contain both human- and ChatGPT-generated texts. We determine the different dynamics in human and machine behavior by first building an Energy-Based Model (EBM) in which we quantify the energy using the Doppler effect. We assume that in the Doppler-effect context, waves represent texts. During human communication, the recipient receives the phonetic energy contained in waveforms. Furthermore, the Doppler effect involves source waves radiating outward in concentric circles towards the observer. As a way to model source frequencies, we incorporate drumhead vibrations. This decision is motivated by the fact that diametric waves propagate in concentric circles towards a fixed outer boundary.

The optimization of the EBM training process is enhanced by incorporating the calculated wave frequencies into its cost function. This energy metric is employed to define the connection between the input text embedding and its corresponding output label (either 0 or 1). Hence, our approach completely differs from existing works that only propose a simple binary classifier ([34, 47]) trained on the input text.

To leverage the Doppler effect, we have made certain assumptions in our approach. Firstly, we consider the source of the text to be either a human or ChatGPT. If the source is a human, we assume that it is in motion with a velocity equal to the variance observed in responses generated by humans. Conversely, if the source is ChatGPT, we assume it is stationary. Additionally, we assume that the observer is another human and is moving with a constant velocity. Thus, we aim to determine how a human observer perceives a response generated by either a human or ChatGPT. To calculate the frequency of the source, we utilize the drumhead vibrations computed using Bessel functions $J_0(\alpha_n)$, as described in Section II. The reason for using drumhead vibrations is as follows: the Doppler effect involves source waves progressing toward the observer in concentric circles, and also, in the modeling of the vibrations (of waves) in drumhead, circular waves propagate in concentric circles toward the fixed outer boundary [54]. Hence, the utilization of drumhead vibrations to calculate the source frequency is grounded in the understanding that waves, such as sound waves, propagate in concentric paths [2]. There are two distinct modes of vibration associated with drumhead vibrations: circular (n) and angular (m). In this study, we specifically concentrate on the circular mode of vibration. This choice is based on our assumption that the "Doppler waves" related to both human-generated and ChatGPT-generated content propagate in a linear fashion, or to put it differently, we assume that the source’s waveform moves along the line of the observer’s sight.

Firstly, we compute the unique values in an embedding of the text ($V(\vec{x})$), and then employ the Bessel function ($J_0(\alpha_n)$) to determine a specific source frequency (E_{f_0}), which is then further utilized to compute E_f . This is based on the idea that each human response vibrates at a distinct frequency. Furthermore, we employ explainable artificial intelligence (XAI) techniques, specifically Integrated Gradients (IG), to generate various perturbations of the input embedding. This is done to observe the behavior and outputs of different perturbed inputs from the perspective of an observer. We are motivated to understand how different perturbations of a particular input influence the perceived nature of the response. For instance, when a human rephrases a sentence using ChatGPT, the output should indicate that the response is generated by ChatGPT. However, if a human rephrases the same text multiple times, the output should indicate that the response is generated by a human. Thus, we aim to incorporate and integrate these effects into our methodology. To achieve this, we compute the energy associated with each perturbed text embedding and average it. Then, we integrate this averaged energy with Binary Cross Entropy (BCE) loss of the model function to have an effective

total loss. Through optimization of this loss function, we are able to compute distinct energies for human-generated text and ChatGPT-generated text.

Furthermore, there are discernible distinctions between the energy of text embeddings and their associated labels across different classes (0 and 1) in the training process. Thus, it establishes a threshold between these classes that helps us determine whether the given input string is ChatGPT-generated or human-produced. Since, in DEMASQ, we have considered the count of perturbed embeddings (utilizing IG) that align with the original label, this count also indicates the shift from human-like responses to more ChatGPT-like responses the text evolves during rephrasing.

B. DEMASQ Components Design and Implementation

Below, we present in detail the main components of our proposed ChatGPT detector *DEMASQ*.

Energy-based model. We develop an energy-based model that quantifies a function known as "energy" to establish the relationship between $(V(\vec{x}), y)$, where $V(\vec{x})$ refers to the input text embedding, and y refers to the output label (0 or 1). We assign a label of 0 to the ChatGPT-generated content and a label of 1 to the human-generated response. In our design, we employ the Sentence Transformers library along with the "msmarco-distilbert-base-tas-b" model to extract the embeddings of the input text. Subsequently, we construct a binary classification model with an energy function comprising the Binary Cross-Entropy loss (BCE) and a regularization energy term, the computation of which involves employing the Doppler effect, as explained below.

To implement the model, we employ a neural network architecture consisting of six fully connected layers with Rectified Linear Unit (ReLU) activations. The sizes of the layers are set to 512, 256, 128, 64, 32, and 1, respectively. Additionally, we utilize the Adam optimizer with a learning rate of 0.0001 to train the model over a period of 12 epochs. Subsequently, we define the energy function to determine the relationship between $(V(\vec{x}), y)$ using the principles of the Doppler effect. We define the loss of EBM consisting the BCE loss and the energy loss E_f (explained in the next section), mentioned below:

$$loss = BCE + E_f(y) - \min(E_f(0), E_f(1)) \quad (4)$$

where E_f is the energy computed for true label, $E_f(0)$ is the energy computed for label 0 and $E_f(1)$ is the energy computed for label 1. The motivation behind incorporating the energy term is to incentivize the model to avoid making predictions that significantly diverge from the desired characteristics. Specifically, this means encouraging the model to assign labels 1 to instances with high-energy values and label 0 to instances with low-energy values.

Adapted Doppler Effect. As defined in Section II, Doppler effect is observed whenever the source of waves is moving with respect to an observer and is given by Equation 1.

Based on established principles, it is widely recognized that an object's energy can be linked to its vibrating frequency. Thus, we utilize the frequency of the observer, which is computed using the aforementioned formula, as the energy value for our energy-based model. To obtain the respective variables in Equation 1, we perform the following computations:

$$\begin{aligned} v_r &= 0.8 \\ v_s &= y \times \text{abs}(\text{var}(V(\vec{x}))) \\ c_v &= \text{var}(V(\vec{x})) \times F_{J_0}(0, 1) \end{aligned}$$

The reasoning for determining the above values for the velocities of the source (v_s) and receiver (v_r) is as follows: considering that higher (lower) frequency corresponds to greater (lesser) energy and velocity, we establish that the frequency, energy, and velocity are directly proportional to each other. Frequency also indicates the rate of change of a specific value within a given timeframe. Therefore, to compute the source's velocity, we consider the variation of the input string embedding multiplied by its label. Thus, if the source corresponds to a human-generated response, represented by a label of 1, the source velocity (v_s) is computed as $1 \times \text{abs}(\text{var}(V(\vec{x})))$. However, for the ChatGPT-generated response, the computed source velocity (v_s) is zero ($0 \times \text{abs}(\text{var}(V(\vec{x})))$). Since we have perceived the receiver (observer) as human, we assume that it is moving with a constant velocity and assign 0.8 (close to label 1 assigned to human-generated content). Generally, in the Doppler effect, the speed of the medium, denoted as c_v , is constant (either the speed of sound or the speed of light). Hence, in this paper, we integrate a constant of fundamental frequency $F_{J_0}(0, 1)$ by multiplying it with the variance of $V(\vec{x})$. As detailed in Section II, the zeroes of $J_0(0, n)$ determine the shape of the drum that vibrates with a specific frequency; thus, $F_{J_0}(0, 1)$ adds a constant of fundamental frequency (fixed size, as shown in Figure 2) with which the medium vibrates and only varies with the strength of the human factor in $V(\vec{x})$ (computed using $\text{var}(V(\vec{x}))$). Alternatively, the medium represents a space in which ChatGPT tries to imitate content similar to human-generated content.

The following steps are executed to determine the source frequency E_{f_0} : as the source moves closer to or farther away from the observer, the frequencies are represented by concentric circles, as depicted in Figure 1. Our motivation is to determine the frequency associated with each embedding, $V(\vec{x})$. We achieve it by calculating the vibrating circular mode (n) used in the Bessel function. We assume each value in $V(\vec{x})$ as the radius of a concentric circle, as shown in Figure 4, but given that $V(\vec{x})$ can contain duplicate values, our initial step involves computing the count of only distinct values in $V(\vec{x})$. The reason being we do not want to consider multiple circles with the same radius.

To simplify the computations and align the unique value's of $V(\vec{x})$ with Euclidean plane's origin $((0, 0))$, the following steps are undertaken: if the highest or lowest value in $V(\vec{x})$ is less than 0, that value is added to each value in $V(\vec{x})$. This

Algorithm 1	Source	Frequency	Computation
(SourceFrequency), E_{f_0}			
1:	Input:	$V(\vec{x})$ having length s	
2:	Output:	Source Frequency	
3:	Compute	$\min_{V(\vec{x})} = \min(V(\vec{x}))$ and $\max_{V(\vec{x})} = \max(V(\vec{x}))$	
4:	if	$\min_{V(\vec{x})} < 0$ then	
5:	for	$i = 1$ to n do	
6:		$V(\vec{x})[i] = V(\vec{x})[i] + \min_{V(\vec{x})}$	
7:	else		
8:	for	$i = 1$ to n do	
9:		$V(\vec{x})[i] = V(\vec{x})[i] - \max_{V(\vec{x})}$	
10:	$u_{V(\vec{x})} =$	unique values in $V(\vec{x})$	
11:	source frequency =	$\text{computezeroes}(0, \text{len}(u_{V(\vec{x})}))$	
12:	return	source frequency, E_{f_0} .	

adjustment ensures that the center of the circles is located at $(0,0)$. Alternatively, if the lowest positive value is greater than or equal to 0, it is appended to each value in $V(\vec{x})$ to achieve the same result. Consequently, the centers of the circles (representing frequencies) are shifted to the origin. However, this shift does not affect our methodology because we only require the unique values of the adjusted embedding to compute the source frequency using the Bessel function, as explained below in the description of Algorithm 1.

Algorithm 1 and Algorithm 2 encompass all these steps, enabling the computation and retrieval of the source frequency, whether it pertains to a human or ChatGPT. Algorithm 1 computes the source frequency of $V(\vec{x})$, which is assumed to have length s . On line 3, it computes the minimum and maximum values of the sequence, denoted as $\min_{V(\vec{x})}$ and $\max_{V(\vec{x})}$, respectively. Next, on lines 4, 5, and 6, it checks if the minimum value, $\min_{V(\vec{x})}$, is less than 0. If it is, it implies that the sequence contains negative values. In this case, the algorithm shifts the entire sequence by adding $\min_{V(\vec{x})}$ to each element, ensuring that the minimum value becomes 0. Else, on line 8 and line 9, if the minimum value is greater than or equal to 0, the algorithm assumes there are no negative values in the sequence. In this case, it shifts the sequence by subtracting $\max_{V(\vec{x})}$ from each element, making the maximum value become 0. Then, on line 10, it calculates the count of the unique values in the updated sequence, denoted as $u_{V(\vec{x})}$. These unique values represent the distinct concentric circles present in the sequence, as shown in Figure 4. On line 11, it calls a function, computezeroes , passing 0 and the length of $u_{V(\vec{x})}$. The reason we pass 0, which is the angular node (m) in Bessel functions, is because we assume that the source (human-generated or ChatGPT-generated content) is moving linearly towards the observer, as explained in Section II-E. Finally, the computed source frequency (E_{f_0}) is returned as the output of the algorithm.

Drumhead vibrations. In Algorithm 2, we compute the

Algorithm 2	Compute	Drumhead	Frequency
(computezeroes)			
1:	Input:	Number of diametric nodes (0),	Number of circular nodes (c_n)
2:	Output:	Drumhead Frequency	
3:	Set	$d_{\max} = m = 0$	\triangleright Number of diametric nodes to calculate
4:	Set	$c_{\max} = n = c_n$	\triangleright Number of circular nodes to calculate
5:	Compute	drumhead fundamental frequency: $F_{J_0}(0,1)$	
6:	Drumhead frequency \leftarrow	$\frac{F_{J_0}(m,n)}{F_{J_0}(0,1)}$	
7:	return	Drumhead frequency	

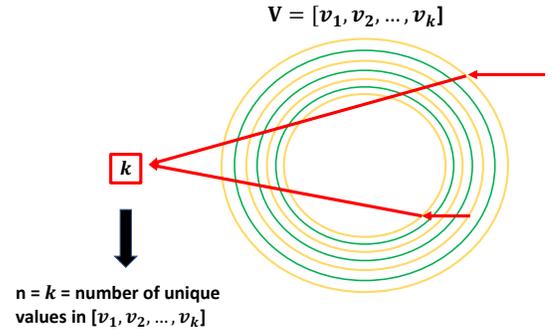


Fig. 4: The computation of the circular mode (n) to determine the frequency with which the drumhead vibrates. We compute the count of unique values (k) (unique concentric circles), as we have assumed each unique value in $V(\vec{x})$ as the radius of a concentric circle.

drumhead frequency using the Drumhead vibrations for a specific mode that are computed utilizing the Bessel functions, as given in Section II. Specifically, we use $(F_{J_0}(0, n)/F_{J_0}(0, 1))$ to compute the relative frequency for different zeroes of $J_0(0, n)$ with respect to fundamental mode $J_0(0, 1)$ frequency. First, we set the number of diametric nodes, i.e., m equal to zero, and the number of circular nodes, i.e., n equal to the number of unique values in $V(\vec{x})$ and pass it to the Bessel function to compute the frequency at that particular mode of the Bessel function. Algorithm 2 proceeds as follows: the inputs to the algorithm are the number of diametric nodes and the number of circular nodes. On lines 3 and 4, we set the input parameters. The number of diametric nodes denoted as d_{\max} , is set to zero, and the number of circular nodes, denoted as $c_{\max}(n)$, is set to the value of the input parameter c_n , which is determined as explained before. In line 5, we compute the drumhead's fundamental frequency. This frequency is obtained by evaluating the Bessel function $J_0(0, 1)$. Finally, on line 6, the drumhead frequency is calculated by dividing the value of $F_{J_0}(m, n)$ by the value of $F_{J_0}(0, 1)$. The algorithm returns the computed drumhead frequency as the output.

In summary, the algorithm takes the number of diametric and circular nodes as input, computes the drumhead fundamental frequency, and then calculates the drumhead frequency based on the obtained fundamental frequency. The

resulting drumhead frequency is returned as the output of the algorithm.

Computing Perturbations of the embedding text. Our second objective, as stated in Section I and Section II, is to accurately determine whether the given text is produced by ChatGPT, even when it undergoes multiple human manipulations to avoid detection. To achieve this, we employ the XAI IG method to perturb the input text embedding, as shown in Section II. We set the baseline $V(\vec{x}') = 0$ because [51] shows inaccurate gradient attribution maps for other baselines. Initially, we identify the top 20 most relevant features, which are the features that contribute the most to the output label of the input text embedding. In other words, we identify the features that make the given content ChatGPT or human. Subsequently, we systematically set each relevant feature to zero, one at a time, to obtain the perturbed input. This approach allows us to generate a perturbed input text embedding that leverages the most significant features from the original text embedding. Finally, we calculate the energies (or frequencies) for each perturbed sample using Algorithm 1 and 2, average them, and utilize the aggregated energy to optimize the loss function.

C. Detecting ChatGPT-generated content

Algorithm 3 aims to train and test an Energy-Based Model (EBM) to distinguish between ChatGPT-generated content and human-generated content. Below, we explain in detail its inner working. It takes as input a sequence, denoted as $V(\vec{x})$, with length s , and its corresponding true label, y , which indicates whether the content is ChatGPT-generated or human-generated. It also receives the number of epochs, T , for training the model. The output of the algorithm is the model’s prediction, which can be either 0 or 1. A prediction of 0 represents ChatGPT-generated content, while a prediction of 1 indicates human-generated content. On line 3, it enters a loop that iterates from $epoch = 1$ to T for training the model. Inside this training loop, we compute the observer’s frequency, E_f (line 8), by applying the Doppler effect. This computation involves considering the speed of the medium, c (line 4), which is calculated based on the variance of the input sequence, $V(\vec{x})$. Additionally, the speed of the source, v_s (line 5), is determined by multiplying the true label, y , with the absolute value of the variance of $V(\vec{x})$. The speed of the receiver, v_r (line 6), is set to a fixed value of 0.8. The source frequency, E_{f_0} (line 7), is obtained by using the function `SourceFrequency($V(\vec{x})$)` on the input embedding $V(\vec{x})$. The observer’s frequency, E_f (line 8), is computed as $\left(\frac{c+v_r}{c-v_s}\right)E_{f_0}$. This calculation considers the effects of the speeds of the medium, source, and receiver, as well as the source frequency. For training the EBM, on line 9, we set the loss function. The loss is defined as a combination of two terms: binary cross-entropy (BCE) loss and $E_f(y) - \min(E_f(0), E_f(1))$ (Section IV-B **Energy-based model**). This choice of loss function aims to optimize the model’s performance in differentiating between

Algorithm 3 DEMASQ main working.

- 1: **Input:** \vec{x} with length s , true label y , training epochs T
 - 2: **Output:** Model output: 0 or 1. "0" for ChatGPT-generated content and "1" for human-generated content
 - 3: **for** $epoch = 1$ to T **do**
 - 4: Speed of the medium, $c = \text{var}(\vec{x}) \times F_{J_0}(0, 1)$
 - 5: Speed of the source, $v_s = y \times \text{abs}(\text{var}(\vec{x}))$
 - 6: Speed of the receiver, $v_r = 0.8$
 - 7: Source frequency, $E_{f_0} = \text{SourceFrequency}(\vec{x})$
 - 8: Observer frequency, $E_f = \left(\frac{c+v_r}{c-v_s}\right)E_{f_0}$ ▷ Computing Observer’s frequency (E_f) using Doppler effect.
 - 9: Set EBM loss function: $\text{loss} = \text{BCE} + E_f(y) - \min(E_{f_0}, E_{f_1})$ ▷ Training EBM Model.
 - 10: Set Optimizer = Adam(lr: 0.0001)
 - 11: Negative energies for ChatGPT-generated content and positive for human-generated content ▷ Testing EBM Model.
 - 12: **Return the energy of the test query.**
-

ChatGPT-generated and human-generated content. The Adam optimizer [29] is selected for training the model. It is a popular optimization algorithm widely used in machine learning. The loop continues for the specified number of epochs, during which the model is trained by adjusting its parameters to minimize the defined loss function.

The algorithm assigns negative energies for ChatGPT-generated content and positive energies for human-generated content. This step involves calculating and assigning energies based on the generated content. Finally on line 12, the algorithm returns the energy of the test query, which represents the model’s prediction of whether the given content is generated by ChatGPT or a human.

In summary, the algorithm iteratively trains an EBM model using the Doppler effect-based observer’s frequency and a combination of BCE loss and the observer’s frequency as the training objective. The trained model is then utilized to predict whether a given content is ChatGPT-generated or human-generated based on the assigned energies.

V. EXPERIMENTAL SETUP

A. Setup

Our experiments are conducted using the PyTorch [44] framework on a server equipped with 4 NVIDIA RTX 8000 (each with 48GB memory), an AMD EPYC 7742, and 1024 GB of main memory. To evaluate the efficacy of our detector, we utilize a benchmark dataset (Section V-B) containing prompts from both ChatGPT and humans, covering a wide range of domains such as Medical, Wikipedia, Open Q&A, Reddit, Finance, arXiv, and Political (cf. Section I). In the subsequent section, we detail the configurations of the different datasets and the accuracy and precision metrics utilized to evaluate the performance of our approach, DEMASQ.

B. Benchmark Dataset

Our benchmark dataset is created using a variety of datasets spanning multiple domains, as mentioned above. To generate the samples from ChatGPT, we made use of the queries

gathered by Guo et al. [23] that also presented the corresponding human answer, then through the use of OpenAI’s API² we queried the latest version of GPT-3.5 at this time (gpt-3.5-turbo) requesting 3 different answers for each input question. Due to the limitation of the available data and the reported unstable behavior of ChatGPT [40], we assumed that the API would properly perform only with English prompts, therefore only querying for answer in this language. The dataset comprises a total of 134,178 unique samples, with 59,214 responses generated by humans and 74,964 responses generated by the ChatGPT model. These samples cover 25,290 distinct questions from various domains, including medicine, open-domain, finance, politics, and rephrased ArXiv’s abstract to cover the hybrid text case. Moreover, we expanded the dataset by incorporating responses from popular social networking platforms (such as Reddit and Wikipedia Q&A), offering a diverse range of user-generated perspectives. To ensure that the answers generated were diverse and not simply reworded versions of the same response, we employed the *mshmarco-distilbert-base-tas-b*³ sentence transformer to assess that those responses had a low grade of sentence similarity. Lastly, we observed that the human samples provided by Guo et al. [23] contained artifacts and symbols, probably due to flaws in the scraping algorithm used to obtain the text. Similarly, in the response of ChatGPT there were also artifacts and symbols, probably intended to implement formatting or symbols to its response (such as table tags or bullet points). Therefore, we employed the sentence transformer *all-MiniLM-L6-v2*⁴ to check and remove the artifacts from those samples.

C. Evaluation Metrics

In order to assess and compare the effectiveness of our approach, we employed the following metrics:

- **True Positive Rate (TPR):** This metric represents the tool’s sensitivity in detecting text that ChatGPT generates. True Positive (TP) is the total number of correctly identified samples, while we consider False Negative (FN) the number of samples not classified as generated text or incorrectly identified as human text. Therefore, $TPR = \frac{TP}{TP+FN}$.
- **True Negative Rate (TNR):** This metric indicates the tool’s specificity in detecting human-generated texts. True Negatives (TN) is the total number of correctly identified samples, while False Positives (FP) is the number of samples incorrectly classified as being produced by ChatGPT. Therefore, $TNR = \frac{TN}{TN+FP}$.

VI. EXPERIMENTAL EVALUATIONS

Next, we empirically illustrate the effectiveness of *DEMASQ* against the curated dataset and compare its efficacy against various baseline detection mechanisms. Further, we

outline how energy computation varies for the human and the ChatGPT-generated responses. Finally, we demonstrate the robustness of *DEMASQ* against the rephrasing techniques and transferability of *DEMASQ* to other domains, i.e., fields different from the ones that our model is trained on.

A. Baseline Approaches

In our evaluation, we assessed various tools and algorithms that have been proposed so far to determine their effectiveness in detecting ChatGPT-generated content, as summarized in Table I. This table provides an overview of these tools’ detection capabilities, specifically in terms of true positive rate (TPR) and true negative rate (TNR), when applied to ChatGPT-generated text. We considered several attributes of the tools mentioned in the literature for detecting ChatGPT-generated content, including the year of release, availability (public or private), and price (free or paid). Our analysis revealed that none of the evaluated approaches consistently achieved high detection rates for ChatGPT-generated text. The most effective online tool we examined demonstrated a success rate of less than 50%, as depicted in Table I. Notably, the maximum accuracy achieved by the detector was around 47%. In addition, recently the authors of [34] propose a ChatGPT-detector, CheckGPT, tailored specifically for academic abstracts of research papers, achieving in this domain an accuracy of 98% to 99%. Nevertheless, their detector’s performance significantly declined, as we evaluated it against our benchmark dataset. Thus, it demonstrates the limited effectiveness of their approach, with accuracy ranging from 2.1% to 13.3%, as further detailed in Section VI-D.

We believe that our dataset serves as a reliable benchmark to evaluate their approach since it contains 134,178 diverse and balanced samples, it effectively measures the similarity between human and ChatGPT responses, and leverages responses from various domains and social networks. As a result, we regard all the examined detectors as baselines for our study. In contrast, our proposed approach, *DEMASQ*, substantially surpasses existing detection methods, achieving an accuracy ranging from 74.5% to 96%.

B. Discipline-Specific detection

This section illustrates our detection approach against different sub-discipline datasets that constitute our benchmark dataset. Sub-disciplines dataset includes samples from different domains such as Medical, Wikipedia, Open Q&A, Reddit, Finance, arXiv, and Political. For each of these sub-datasets, we individually train our EBM (described in Section IV) and analyze the energy spread for the human-generated responses and ChatGPT-generated responses. Table II details our training and inference results. Since we train various EBM models for individual datasets, we generate unique detectors for diverse fields using the same foundational EBM. Thus, our work provides a generic detector that can be deployed in any field to detect whether the text is ChatGPT or human-generated.

Table II provides an overview of the performance of different sub-datasets in the conducted experiments. For each

²<https://openai.com/blog/introducing-chatgpt-and-whisper-apis>

³<https://huggingface.co/sentence-transformers/mshmarco-distilbert-base-tas-b>

⁴<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

TABLE I: Summary of analyzed papers. TPR represents the detection capability of ChatGPT-generated text, while TNR reflects the detection capability of Human-generated text.

Approach	Published in	Publicly Available	Free/Paid	TPR (%)	TNR (%)
Bleumink et al. [8]	2023	✓	Paid	13.4	95.4
ZeroGPT [60]	2023	✓	Paid	45.7	92.2
OpenAI Classifier [39]	2023	✓	Free	31.9	91.8
GPTZero [43]	2023	✓	Paid	27.3	93.5
Hugging Face [18]	2023	✓	Free	10.7	62.9
Guo et al. [23]	2023	✓	Free	47.3	98.0
Perplexity (PPL) [22]	2023	✓	Free	44.4	98.3
Writefull GPT [55]	2023	✓	Paid	21.6	99.3
Copyleaks [14]	2023	✓	Paid	22.9	92.1
Cotton et al. [12]	2023	×	-	-	-
Khalil et al. [28]	2023	×	-	-	-
Mitrovic et al. [37]	2023	×	-	-	-
Content at Scale [4]	2022	✓	Paid	38.4	79.8
Originality.ai [41]	2022	×	Paid	7.6	95.0
Writer AI Detector [56]	2022	✓	Paid	6.9	94.5
Draft and Goal [17]	2022	✓	Free	23.7	91.1
Gao et al. [20]	2022	×	-	-	-
Liu et al. [34] (Detector 1 - Task1)	2023	✓	Free	2.1	98.8
Liu et al. [34] (Detector 2 - Task2)	2023	✓	Free	13.3	99.4
Liu et al. [34] (Detector 3 - Task3)	2023	✓	Free	6.6	98.7

specific dataset, we mention the number of samples in it, the True Positive Rate (TPR), and the True Negative Rate (TNR). Next, we examine the findings of these specific datasets to illustrate the insights obtained. The Medical dataset comprises 4,992 records, achieving a TPR of 96.8% and a TNR of 96.4%. It indicates that the model trained on the Medical dataset exhibits an efficacy of 96.8% in correctly identifying ChatGPT-generated responses and 96.4% in correctly identifying human-generated responses. Similarly, the Financial dataset, consisting of 15,732 records, demonstrates a TPR of 93.6% and a TNR of 93.7%. With 3,368 records, the Wiki dataset achieves a TPR of 84.2% and a TNR of 83.2%. The Open Q&A dataset, encompassing 4,748 records, shows a TPR of 74.5% and a TNR of 73.0%. Among the examined datasets, the Reddit dataset stands out with 99,054 records and perfect scores of 100% for both TPR and TNR. This analysis implies that the model trained on the Reddit dataset accurately identifies ChatGPT-generated and human-generated content. Furthermore, the Political dataset, consisting of 3,620 records, demonstrates a TPR of 92% and a TNR of 91.5%. The arXiv dataset, comprising 2,664 records, shows a TPR of 90% and a TNR of 89.5%. Finally, the Combined dataset, consisting of 134,178 records, achieves a TPR of 97% and a TNR of 96.5%. Overall, each dataset exhibits varying levels of accuracy, as reflected in their respective TPR and TNR values. These results shed light on the models' performance on different datasets, allowing for a comparative analysis of their ability to accurately identify ChatGPT-generated and human-generated responses.

Further, as detailed in Table III, we trained and tested our model using the academic abstract datasets provided by Liu et al. [34] named Task1, Task2, and Task3. In their research [34], the authors developed CheckGPT, a detector for LLM-generated content, specifically designed to determine if academic abstracts in computer science (CS), physics (PHX), and humanitarian and social sciences (HSS) were authored by ChatGPT or not. Liu et al. [34] broke down the analysis of CheckGPT into three tasks: 1) Full abstracts written by GPT model, 2) Abstracts partially completed by GPT model, and 3) Abstracts polished by GPT model. For Task1, the researchers presented ChatGPT with a title and asked it to generate a corresponding abstract. For Task2, they supplied ChatGPT with the first half of an abstract and requested ChatGPT to complete the it. For Task3, the researchers gave ChatGPT a complete abstract to rephrase. We use the CS-TASK datasets that the authors made available and evaluated *DEMASQ* against all these tasks. Our *DEMASQ* EBM was retrained on Task1, Task2, and Task3 datasets, demonstrating accuracies of 96.4% for Task1, 88.7% for Task2, and 82.5% for Task3.

C. Energy distribution of human-written response and ChatGPT-generated content

We explore the differences in energy computation between human-generated responses and responses generated by ChatGPT. We visually represent the energy distribution for each dataset and highlight the significant variations in energy values between human and ChatGPT-generated answers. By examin-

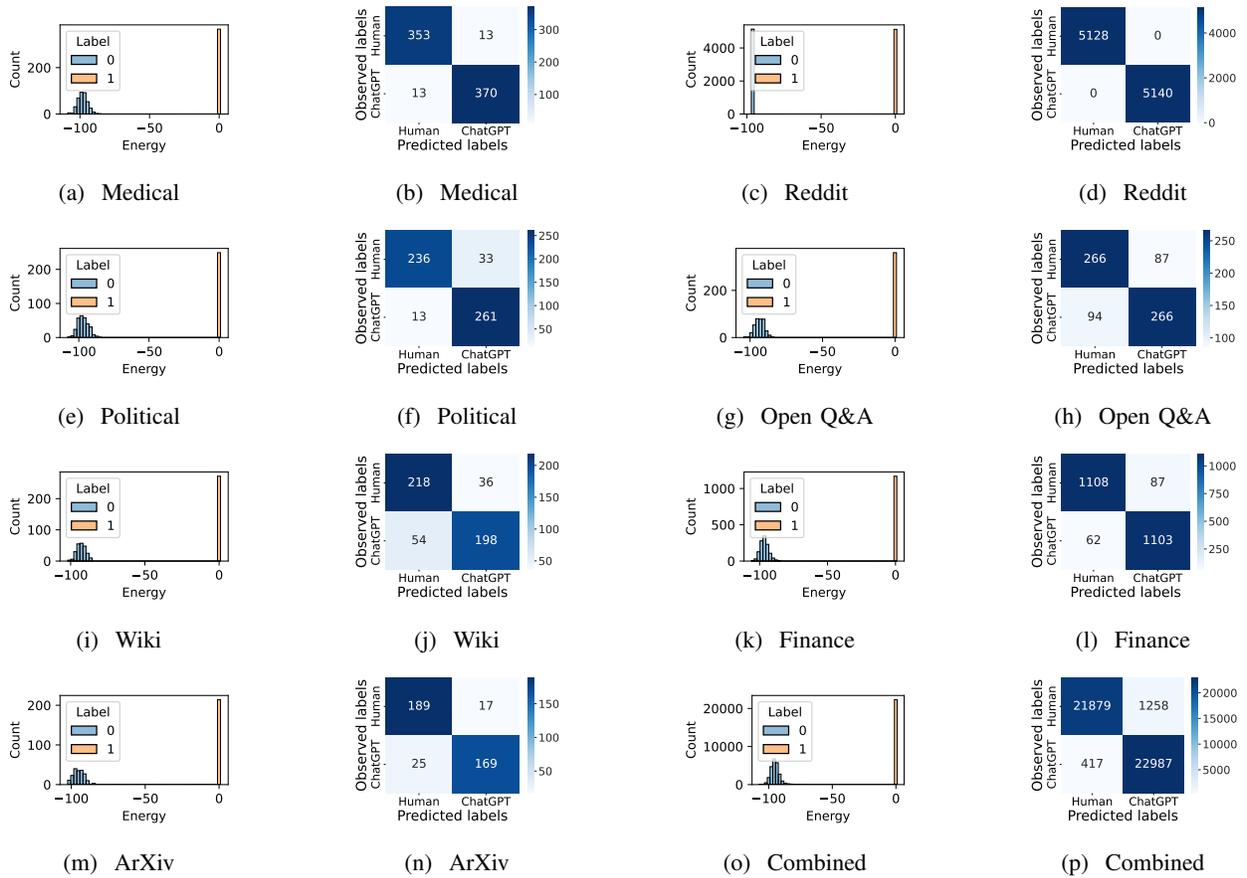


Fig. 5: Different energy levels for different datasets. (a), (c), (e), (g), (i), (k), (m), and (o) represent the energy histogram plots for the Medical, Reddit, Political, Open Q&A, Wiki, Finance, ArXiv, and the combined dataset. (b), (d), (f), (h), (j), (l), (n), and (p) represent the confusion matrix for the same.

TABLE II: Statistical results of sub-datasets in the experiments.

Dataset	Records	TPR (%)	TNR(%)
Medical	4,992	96.6	96.4
Financial	15,732	94.6	92.7
Wiki	3,368	78.5	85.8
Open Q&A	4,748	73.9	75.4
Reddit	99,054	100.0	100.0
Political	3,620	95.2	87.7
arXiv	2,664	87.1	91.7
Combined	134,178	97.0	96.5

TABLE III: Statistical results of the *DEMASQ* in other domains.

Dataset	Records	TPR (%)	TNR(%)
Task1	100000	94.7	97.9
Task2	100000	88.7	88.4
Task3	100000	84.3	81.0

ing the energy plots for each dataset, as depicted in Figure 5, we can clearly observe distinct disparities in energy levels between human-generated text and ChatGPT-generated content. These plots specifically represent the perturbed energies of individual samples within the test set of each dataset. Figures 5a, 5c, 5e, 5g, 5i, 5k, 5m, and 5o illustrate the energy distribution

for human-generated content and ChatGPT-generated content, showcasing two distinct levels. The lower values, which are highly negative, correspond to label 0 (ChatGPT-generated content), while the larger values belong to label 1 (human-generated content). Therefore, by analyzing the energy values, it becomes relatively easy to differentiate between human-generated and ChatGPT-generated content. Figures 5b, 5d, 5f, 5h, 5j, 5l, 5n, and 5p represents the energy distribution of human- and ChatGPT-generated content in relation to the EBM prediction. It is important to highlight that misclassifications were observed for certain samples across different datasets, which introduces the possibility of reverse perturbed energies for both label options (0 or 1). Consequently, confusion matrices for both label 0 and label 1, with distinct true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), are depicted in Figures 5b, 5f, 5h, 5j, 5l, 5n, and 5p. However, for the reddit dataset, the EBM achieved a testing accuracy of 100%, as evidenced by Figure 5c, where both true positive rate (TPR) and true negative rate (TNR) are equal to 100%.

TABLE IV: Rephrasing statistical results for CheckGPT.

Dataset	#Records	TPR (%)	TNR (%)
Task1	100,000	82.7	99.7
Task2	100,000	56.3	96.1
Task3	100,000	3.8	99.0

D. Rephrasing detection

In the following, we evaluate the effectiveness of our approach, *DEMASQ*, by comparing it to the rephrasing of content generated by ChatGPT. As mentioned earlier, our benchmark dataset has been carefully curated to encompass various domains, and *DEMASQ* demonstrates remarkable performance on it. However, to evaluate the effectiveness of *DEMASQ* in a domain that differs from those covered in our dataset, we utilize the Task1, Task2, and Task3 datasets referenced in the research conducted by Liu et al. [34]. We observe that in these datasets the ChatGPT prompts predominantly begin with phrases like "In this work," "In this paper," or "This paper," which would highly introduce bias into the training of naive classifier. Therefore, we will discuss that the detector proposed by Liu et al. does not accurately function as demonstrated in their research, as we will showcase in the upcoming analysis. Consequently, in this study, we rephrase the samples of Task1, Task2, and Task3 that contains these specific wordings and evaluate the EBM model trained on these updated datasets, respectively, as shown in Section VI-B.

The effectiveness of CheckGPT [34] was examined by testing it against the rephrased datasets of Task1, Task2, and Task3, and the results are summarized in Table IV. The accuracy of CheckGPT decreased progressively as we evaluated it on the rephrased datasets. Specifically, for rephrased Task1, CheckGPT achieved an accuracy of 82.72%, for rephrased Task2, it obtained an accuracy of 56.35%, and for rephrased Task3, it achieved an accuracy of 3.76%. These results indicate that CheckGPT does not deliver the expected performance when tasked with detecting the rephrased versions of academic abstracts. This discrepancy arises because CheckGPT was originally designed to identify academic abstracts in their original form, thus, introducing bias for the CheckGPT detection. It is worth noting that the authors in [34] developed distinct detectors for different tasks, training each task on separate models. In contrast, our approach involved testing the rephrased Task1, Task2, and Task3 against the EBM model trained on our combined benchmark dataset, to understand and measure how *DEMASQ* behaves when presented with never seen before samples. The accuracy achieved for rephrased Task1 was 76.9%, for rephrased Task2 it was 68.7%, and for rephrased Task3 it was 58.3%, as shown in Table V. However, when we trained separate EBMs for each original task and tested them on the rephrased versions, the true positive rates (TPR) obtained were 93% for rephrased Task1, 83.4% for rephrased Task2, and 75.4% for rephrased Task3, as shown in Table VI. Thus, in comparison to CheckGPT's experimental evaluation results of 82.72% for rephrased Task1, 56.35% for rephrased Task2, and 3.76% for rephrased Task3 (Table IV),

TABLE V: Statistical results for *DEMASQ* in other domains.

Dataset	# Records	TNR (%)	TPR (%)
Task1	100,000	76.9	76.6
Task2	100,000	68.7	68.4
Task3	100,000	58.3	58.9

TABLE VI: Rephrasing statistical results for *DEMASQ*

Dataset	#Records	TPR (%)	TNR (%)
Task1	100,000	93.0	94.1
Task2	100,000	83.4	84.5
Task3	100,000	74.9	75.1

our proposed approach, *DEMASQ*, outperforms CheckGPT by a significant margin.

Therefore, when we compare *DEMASQ* and *CheckGPT* [34] under the same realistic settings, *DEMASQ* outperforms other detectors for the case of hybrid text, even if notably the accuracy is not as high as in our benchmark dataset.

VII. RELATED WORKS

In general, automated machine learning techniques for identifying synthetic texts can be categorized into three primary groups: i) Basic classifiers [23], [50], ii) Zero-shot detection approaches [31], [36], [59], and iii) Detection methods based on fine-tuning [37].

Basic classifiers. The first category, simple classifiers, includes various detection techniques. For instance, OpenAI's logistic regression model [50] was trained on TF-IDF, unigram, and bigram features and specifically analyzed text generated by *GPT-2*. Through exploration of different generation strategies and model parameters, it was discovered that these classifiers can achieve accuracy levels of up to 97%. However, detecting shorter outputs proves more challenging for these models compared to longer ones. Another simple classifier proposed by Guo et al. [23] analyzed linguistic and stylistic characteristics of responses from ChatGPT and human experts. Unfortunately, these detection models were ineffective in identifying ChatGPT-generated text due to the imbalanced nature of the dataset they were trained on, failing to capture all of ChatGPT's text generation styles [42]. In a separate study, Kushnareva et al. [31] trained a logistic regression model using *Topological Data Analysis* (TDA) to extract interpretable topological features, such as the number of connected components, edges, and cycles in the graph, for recognizing artificial text. However, this approach is unlikely to be effective for ChatGPT as it was not specifically tested on that particular model, but instead was conducted on datasets from WebText & *GPT-2*, Amazon Reviews & *GPT-2*, RealNews & *GROVER* (FakeNews) [59]. Furthermore, prior work [42] tested the authors [31] designed model and got an accuracy of 25.1%.

Zero-shot detection. Zero-shot detection techniques also includes a variety of approaches for identifying AI-generated text. For instance, Mitchell et al. [36] introduced a method that leverages the log probabilities of the generative model to detect AI-generated text. However, this approach is primarily tailored to *GPT-2* prompts and our evaluation indicates that it

does not generalize well to *GPT-3* models. In another study, the authors in [59] utilized a transformer similar to the one employed in *GPT-2* to create a tool called *Grover*. *Grover* is capable of generating text such as fake news and is also able to detect its own generated text. The study demonstrated the effectiveness of *Grover* in verifying self-generated fake news. Nevertheless, it remains uncertain how well *Grover* would perform on text generated by other *GPT* models. Additionally, OpenAI [50] developed a *GPT-2* detector using a 1.5 billion parameter model that can accurately identify the top 40 generated outputs with an accuracy ranging from 83% to 85%. However, when fine-tuned on the Amazon reviews dataset, the accuracy dropped to 76%.

Fine-tuned detection. In the case of fine-tuning based detection mechanism, Mitrović et al. [37] conducted a study exploring the training of a machine learning model that distinguishes ChatGPT-generated two-line restaurant reviews generated from real human-written reviews. They employed a framework based on *DistilBERT*, which is a lightweight model based on *BERT*. In another investigation, Solaiman et al. [50] performed experiments to fine-tune pre-trained language models for the detection of AI-generated texts. They utilized *RoBERTa_{BASE}* and *RoBERTa_{LARGE}* as the basis for their classifiers. However, it is important to note that this approach failed to detect text generated by ChatGPT, as illustrated in [46]. As a result, all the aforementioned detection methods are either exclusively applicable to prompts generated by the *GPT-2* model or tailored for specific tasks.

Other methods. Alternative methods exist outside the three main categories of detection approaches that deserve attention. These methods involve testing ChatGPT-generated text against plagiarism tools, developing Deep Neural Network-based detection tools, employing sampling-based approaches, and utilizing various online detection tools. Existing tools such as *RoBERTa*, *Grover*, and *GPT-2* have been used to assess the originality of educational content compared to ChatGPT-generated text. For instance, a study evaluated the effectiveness of popular plagiarism-detection tools, *iThenticate* and *Turnitin*, in identifying plagiarism in 50 essays generated by ChatGPT [28]. Another study by Gao et al. [20] aimed to compare ChatGPT-generated academic paper abstracts using a *GPT-2* Output Detector *RoBERTa*, a plagiarism checker, and human review. The authors collected ten research abstracts from five high-impact medical journals and then used ChatGPT to output research abstracts based on their titles and journals. In a different study, Liu et al.[34] developed a tool called CheckGPT, which is a content detector for LLMs aimed at determining whether academic abstracts in the fields of computer science (CS), physics, humanities, and social sciences (HSS) were generated by ChatGPT. The authors achieved a detection accuracy ranging from 98% to 99% for identifying ChatGPT-generated academic abstracts. Yet, as discussed in Section VI, when their CS Task1, Task2, and Task3 datasets underwent paraphrasing to eliminate potential biases stemming from phrases like "In this work," "In this paper," or "This paper," CheckGPT showed reduced true positive rates (TPR).

Nonetheless, prior research [42] has revealed the shortcomings of the discussed detection methods in accurately identifying text generated by ChatGPT. Additionally, in the same study, the authors evaluated multiple online tools [41], [4], [14], [18], [22], [39], [43], [55], [60], but the most effective tool for detecting generated text was found to have a success rate of less than 50%.

In contrast, we propose a detection tool that significantly distinguishes itself from previous methodologies by incorporating the characteristics of human speech and the thought process involved in communication with both humans and machines. Consequently, our approach outperforms other existing methods in detecting ChatGPT-generated text.

VIII. CONCLUSION

We propose the design and implementation of a novel detection tool named *DEMASQ*, which accurately determines the source of a given text by effectively differentiating between responses generated by the ChatGPT and those generated by humans. Unlike existing detection methods that focus solely on prompts generated by specific GPT models or serve specific purposes within certain fields (often yielding low accuracy), our approach is versatile and can successfully distinguish any text produced by other AI tools, especially ChatGPT, from human-generated text. By leveraging the intricate dynamics of human-to-human and human-to-machine communication, we incorporate the Doppler effect technique into our detection mechanism to account for inherent biases. Additionally, we address ChatGPT's rephrasing techniques by employing the explainable AI Integrated Gradients (IG) method. By integrating all these components, we construct an energy-based model that significantly enhances the detector's ability to accurately differentiate between responses generated by humans and ChatGPT. *DEMASQ* significantly outperforms the existing ChatGPT detectors. Our approach goes beyond the limitations of existing methods, offering a more comprehensive and effective solution for source identification in the ever-evolving landscape of AI-generated text. Our extensive evaluations of *DEMASQ* on the proposed benchmark dataset demonstrate that *DEMASQ* significantly outperforms the existing ChatGPT detectors. Our approach goes beyond the limitations of existing methods, offering a more comprehensive and effective solution for source identification in the ever-evolving landscape of AI-generated text.

ACKNOWLEDGMENT

This research received funding from the following organization: Intel Private AI Collaborate Research Institute, Deutsche Forschungsgemeinschaft (DFG) SFB-1119 CROSS-ING/236615297, the Horizon programme of the European Union under the grant agreement No. 101093126 (ACES) No. 101070537 (CROSSCON).

REFERENCES

- [1] Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *International Conference on Learning Representations*, 2018.

- [2] EN da C Andrade. Doppler and the doppler effect, 1959.
- [3] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. What is relevant in a text document?: An interpretable machine learning approach. *PLoS one*, 12(8):e0181142, 2017.
- [4] Content at Scale. AI DETECTOR. <https://contentatscale.ai/ai-content-detector/>, 2023. [Online; accessed 23-Mar-2023].
- [5] Microsoft Azure. Model interpretability in Azure Machine Learning (preview). <https://docs.microsoft.com/en-us/azure/machine-learning/how-to-machine-learning-interpretability>, 2022. [Online; accessed 13-Mar-2022].
- [6] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.
- [7] Som Biswas. Chatgpt and the future of medical writing, 2023.
- [8] Arend Groot Bleumink and Aaron Shikhule. Keeping ai honest in education: Identifying gpt-generated text, 2023. <https://www.aicheatcheck.com/>.
- [9] Frank Bowman. *Introduction to Bessel functions*. Courier Corporation, 2012.
- [10] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.
- [11] Google Cloud. Explainable AI: Tools and frameworks to understand and interpret your machine learning models. <https://cloud.google.com/explainable-ai>, 2022. [Online; accessed 13-Mar-2022].
- [12] Debby RE Cotton, Peter A Cotton, and J Reuben Shipway. Chatting and cheating: Ensuring academic integrity in the era of chatgpt. *Innovations in Education and Teaching International*, pages 1–12, 2023.
- [13] Luigi De Angelis, Francesco Baglivo, Guglielmo Arzilli, Gaetano Pierpaolo Privitera, Paolo Ferragina, Alberto Eugenio Tozzi, and Caterina Rizzo. Chatgpt and the rise of large language models: The new ai-driven infodemic threat in public health. *Available at SSRN 4352931*, 2023.
- [14] Copyleaks AI Content Detector. Copyleaks. <https://copyleaks.com/ai-content-detector>, 2023. [Online; accessed 23-Mar-2023].
- [15] docs.scipy.org. Bessel Functions. https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.jn_zeros.html#scipy.special.jn_zeros, 2023. [Online; accessed 15-Oct-2023].
- [16] Michael Dowling and Brian Lucey. Chatgpt for (finance) research: The bananarama conjecture. *Finance Research Letters*, page 103662, 2023.
- [17] Draft and Goal. ChatGPT - GPT3 Content Detector. <https://detector.dng.ai/>, 2023. [Online; accessed 23-Mar-2023].
- [18] Hugging Face. Hugging Face ChatGPT-Detection . <https://huggingface.co/spaces/imseldrith/ChatGPT-Detection>, 2023. [Online; accessed 23-Mar-2023].
- [19] Leon Fröhling and Arkaitz Zubiaga. Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover. *PeerJ Computer Science*, 7:e443, April 2021. <https://peerj.com/articles/cs-443/#supplementary-material>.
- [20] Catherine A Gao, Frederick M Howard, Nikolay S Markov, Emma C Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T Pearson. Comparing scientific abstracts generated by chatgpt to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. *bioRxiv*, pages 2022–12, 2022.
- [21] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*, 2019. <http://gltr.io/>.
- [22] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. Chatgpt Detector Using Linguistic Features. <https://huggingface.co/spaces/Hello-SimpleAI/chatgpt-detector-ling>, 2023. [Online; accessed 23-Mar-2023].
- [23] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*, 2023. <https://github.com/Hello-SimpleAI/chatgpt-comparison-detection>.
- [24] Philipp Hacker, Andreas Engel, and Marco Mauer. Regulating chatgpt and other large generative ai models. *arXiv preprint arXiv:2302.02337*, 2023.
- [25] Tim Ingold. Culture and the perception of the environment. In *Bush base, forest farm*, pages 38–56. Routledge, 2002.
- [26] Alon Jacovi, Oren Sar Shalom, and Yoav Goldberg. Understanding convolutional neural networks for text classification. *arXiv preprint arXiv:1809.08037*, 2018.
- [27] Christopher HM Jenkins and Umesh A Korde. Membrane vibration experiments: An historical review and recent results. *Journal of Sound and Vibration*, 295(3-5):602–613, 2006.
- [28] Mohammad Khalil and Erkan Er. Will chatgpt get you caught? rethinking of plagiarism detection. *arXiv preprint arXiv:2302.04335*, 2023.
- [29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [30] Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, Kirill Trapeznikov, Scott Ruston, and Huan Liu. Stylometric detection of ai-generated text in twitter timelines. *arXiv preprint arXiv:2303.03697*, 2023. <https://github.com/TSKumarage/Stylo-Det-AI-Gen-Twitter-Timelines>.
- [31] Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. Artificial text detection via examining the topology of attention maps. *arXiv preprint arXiv:2109.04825*, 2021. <https://github.com/danchern97/tda4atd>.
- [32] Nathan Lambert, Louis Castricato, Leandro von Werra, and Alex Havrilla. Illustrating reinforcement learning from human feedback (rlhf). *Hugging Face Blog*, 2022. <https://huggingface.co/blog/rlhf>.
- [33] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fugie Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.
- [34] Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. Check me if you can: Detecting chatgpt-generated academic writing using checkgpt. *arXiv preprint arXiv:2306.05524*, 2023.
- [35] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777, 2017.
- [36] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*, 2023. <https://github.com/eric-mitchell/detect-gpt> <https://detectgpt.ericmitchell.ai/>.
- [37] Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *arXiv preprint arXiv:2301.13852*, 2023.
- [38] Reham Omar, Omij Mangukiya, Panos Kalnis, and Essam Mansour. Chatgpt versus traditional question answering for knowledge graphs: Current status and future directions towards knowledge graph chatbots. *arXiv preprint arXiv:2302.06466*, 2023.
- [39] OpenAI. <https://beta.openai.com/ai-text-classifier>, January 2023.
- [40] OpenAI. Gpt-4 technical report, 2023.
- [41] Originality.ai. <https://originality.ai/>, 2022. [Online; accessed 28-Mar-2023].
- [42] Alessandro Pegoraro, Kavita Kumari, Hossein Fereidooni, and Ahmad-Reza Sadeghi. To chatgpt, or not to chatgpt: That is the question!, 2023.
- [43] Edward Tian (Princeton). GPTZero. <https://gptzero.me/>, 2023. [Online; accessed 23-Mar-2023].
- [44] Pytorch, 2019. <https://pytorch.org>.
- [45] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " 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, 2016.
- [46] Jürgen Rudolph, Samson Tan, and Shannon Tan. Chatgpt: Bullshit spewer or the end of traditional assessments in higher education? *Journal of Applied Learning and Teaching*, 6(1), 2023.
- [47] Bernard Sapoval, Th Gobron, and A Margolina. Vibrations of fractal drums. *Physical review letters*, 67(21):2974, 1991.
- [48] Yiqiu Shen, Laura Heacock, Jonathan Elias, Keith D Hentel, Beatriu Reig, George Shih, and Linda Moy. Chatgpt and other large language models are double-edged swords, 2023.
- [49] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.
- [50] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. Release strategies and the social

- impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019. <https://openai-openai-detector.hf.space/> <https://github.com/openai/gpt-2-output-dataset/tree/master/detector>.
- [51] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.
- [52] Teo Susnjak. Applying bert and chatgpt for sentiment analysis of lyme disease in scientific literature. *arXiv preprint arXiv:2302.06474*, 2023.
- [53] Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. The science of detecting llm-generated texts. *arXiv preprint arXiv:2303.07205*, 2023.
- [54] Wikipedia. https://en.wikipedia.org/wiki/Vibrations_of_a_circular_membrane#References, June 2023.
- [55] writefull. GPT Detector. <https://x.writefull.com/gpt-detector>, 2023. [Online; accessed 23-Mar-2023].
- [56] Writer.com. AI Content Detector. <https://writer.com/ai-content-detector/>, 2023. [Online; accessed 23-Mar-2023].
- [57] Agnieszka Wykowska, Jasmin Kajopoulos, Miguel Obando-Leiton, Sushil Singh Chauhan, John-John Cabibihan, and Gordon Cheng. Humans are well tuned to detecting agents among non-agents: examining the sensitivity of human perception to behavioral characteristics of intentional systems. *International Journal of Social Robotics*, 7:767–781, 2015.
- [58] Yee Hui Yeo, Jamil S Samaan, Wee Han Ng, Peng-Sheng Ting, Hirsh Trivedi, Aarshi Vipani, Walid Ayoub, Ju Dong Yang, Omer Liran, Brennan Spiegel, et al. Assessing the performance of chatgpt in answering questions regarding cirrhosis and hepatocellular carcinoma. *medRxiv*, pages 2023–02, 2023.
- [59] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. *Advances in neural information processing systems*, 32, 2019. <https://rowanzellers.com/grover/>.
- [60] ZeroGPT. <https://www.zerogpt.com>, January 2023.

APPENDIX

A. Large Language Models (LLMs)

Large Language Models represent sophisticated artificial intelligence systems specifically developed to comprehend and generate text that closely resembles human language. These models undergo extensive training on vast datasets, incorporating diverse sources such as books, articles, websites, and more. This training equips LLMs with the ability to generate coherent and contextually appropriate responses. Some notable examples of LLMs include: GPT-3 (Generative Pre-trained Transformer 3), GPT-2 (Generative Pre-trained Transformer 2), Transformer-XL, BERT (Bidirectional Encoder Representations from Transformers), and ChatGPT. The aforementioned models are merely a few examples of LLMs, as there exist numerous other variations and models crafted by different organizations and researchers. Each of these models possesses unique strengths and specializes in specific areas, contributing to the diverse landscape of LLM development.

B. Drumhead vibrations

When the head of a drum is struck, it changes shape and compresses the air inside the shell. This compressed air then exerts pressure on the bottom head, causing it to change shape. These shape changes are transmitted to the drum shell, resulting in reflections and a repeated cycle that generates vibrations. The top and bottom head vibrations propagate into the surrounding air, becoming audible. As the vibrations of the drum heads gradually dampen, the sound diminishes. The behavior of an idealized drumhead can be understood by examining the vibrations of a two-dimensional elastic membrane under tension [47] [27]. This membrane, which

can be modeled as a circular surface with uniform thickness connected to a rigid frame, exhibits fascinating properties. The membrane can store vibrational energy at specific resonant frequencies through the resonance phenomenon. The surface of the membrane moves in distinct patterns characterized by standing waves, referred to as normal modes. The lowest frequency normal mode is called the fundamental mode, and the membrane possesses an infinite number of these normal modes.