Contents lists available at ScienceDirect

Expert Systems With Applications

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

Review

The Imitation Game revisited: A comprehensive survey on recent advances in AI-generated text detection

Zhiwei Yang[a], Zhengjie Feng[b], Rongxin Huo[a], Huiru Lin[c,d,*], Hanghan Zheng[b], Ruichi Nie[b], Hongrui Chen[b]

[a] *Guangdong Institute of Smart Education, Jinan University, Guangzhou, 510632, China*
[b] *International School, Jinan University, Guangzhou, 510632, China*
[c] *School of Physical Education, Jinan University, Guangzhou, 510632, China*
[d] *Guangdong Provincial Key Laboratory of Speed Capability Research, Guangzhou, 510632, China*

ARTICLE INFO

Keywords:
AI-generated text detection
Methodological review
Hierarchical analysis
Potential future research

ABSTRACT

In recent years, AI-generated text detection (AIGTD) has attracted more and more attention, with numerous novel methodologies being proposed. However, most existing reviews on this topic tend to be fragmented and incoherent in content, lacking a coherent and comprehensive framework for understanding. This paper comprehensively analyzes and summarizes the latest advancements and prominent technologies in this fast-moving field. In order to do that, we introduce a novel comprehensive multi-level taxonomy for AIGTD approaches, where the existing research can be broadly categorized into three directions, tackling the key challenges of classifier training, intrinsic attributes, and information embedding, respectively. To help researchers and practitioners understand and address detection and attack scenarios, we also introduce a classification of black-box and white-box models based on interpretability and transparency, as well as the computational requirements required to use the baseline methods. Moreover, we carefully provide a comprehensive performance comparison and analysis across several datasets for these methods, collect commonly used benchmark datasets, and outline potential future research directions in this field. To facilitate sharing, we consistently maintain the relevant materials at: https://github.com/Nicozwy/AIGTD-Survey.

Contents

[*] Corresponding author.
E-mail addresses: yangzw@jnu.edu.cn (Z. Yang), jerryfred@stu2021.jnu.edu.cn (Z. Feng), huorongxin1997@stu.jnu.edu.cn (R. Huo), linhuiru@jnu.edu.cn (H. Lin), zhenghanghan@stu2021.jnu.edu.cn (H. Zheng), nrc1112@stu2021.jnu.edu.cn (R. Nie), 2021102718lele@stu2021.jnu.edu.cn (H. Chen).

https://doi.org/10.1016/j.eswa.2025.126694
Received 14 August 2024; Received in revised form 7 November 2024; Accepted 26 January 2025
Available online 8 February 2025

1. Introduction

The identification of text generated by an artificial intelligence (AI) system is referred to as AI-generated text detection (AIGTD). This process leverages various technological approaches to automatically recognize, examine, and handle the content of the text. With the rapid advancements in generative AI technologies, such as ChatGPT and GPT-4 (Achiam et al., 2023), the ability to generate text content like humans has become increasingly powerful and accessible to a wide range of users. For example, AI-generated fake reviews can deceive and mislead consumers (Vidanagama, Silva, & Karunananda, 2022; Xylogiannopoulos, Xanthopoulos, Karampelas, & Bakamitsos, 2024). As a result, the internet has seen a significant proliferation of automatically generated text, making it increasingly challenging to differentiate between human-crafted and AI-originated content. Thus, the need to reliably identify AI-generated text has taken on heightened urgency to preserve the authenticity and credibility of information.

While previous surveys have provided valuable insights into AIGTD (Ghosal et al., 2023; Zhang, Ma et al., 2024), most of them are limited to some specific application areas, e.g., news generation and academic writing, failing to provide a coherent and comprehensive framework for understanding the broader progress and trends in this field. Given the rapidly evolving landscape of AI-powered text generation, it necessitates a fresh and comprehensive investigation into the current research hotspots and emerging challenges that have been studied and addressed in recent years. Therefore, we summarize the recent methods of AIGTD from classifier training, intrinsic attributes, and information embedding, as follows:

Classifier Training: Early solutions for AIGTD rely on machine learning and deep learning techniques. Being trained on a large amount of labeled data, classifiers can automatically differentiate human-written text from AI-generated text. The corresponding processing flow is roughly shown in Fig. 1(a). First, collected datasets are a key component and include large amounts of text data written by humans as well as text generated by AI systems. Rigorous data processing techniques are then applied to meticulously clean and organize the collected text data, including removing noise and irrelevant information (such as HTML tags, special characters, etc.) (Moreno & Redondo, 2016), standardizing text (such as lowercase, removing stops, stem extraction, etc.) (Almeida, Silva, Santos, & Hidalgo, 2016), and annotating data (Sleimi, Sannier, Sabetzadeh, Briand, & Dann, 2018). Afterward, the processed data is used to train machine learning models (e.g. SVM Abdullah & Abdulazeez, 2021, random forest Chen, Wu, Chen, Lu, & Ding, 2022, neural network Chakraborty et al., 2023) or deep learning models (e.g. BERT Szczepański, Pawlicki, Kozik, & Choraś, 2021, GPT Mitchell, Lee, Khazatsky, Manning, & Finn, 2023) for classification. Finally, the trained model is deployed to the production environment for real-time text detection. Generally, the trained model will be backed up and deployed to a server or cloud platform, or an API interface will be set up to enable application-level calls for text detection tasks. Thanks to the widespread application and continued refinement of pre-trained models, the accuracy of detecting AI-generated text has been significantly improved. Furthermore, the increasing availability of heavily annotated training data, coupled with advancements in computing resources, has also propelled the development of this research field.

Intrinsic Attributes: Driven by the proliferation of pre-trained large language models (LLMs), researchers have leveraged the inherent properties of these models for AIGTD without relying on specialized training data (Benaich & Hogarth, 2020; Mo, Qin, Dong, Zhu, & Li, 2024; Wang, Chen et al., 2024). Since LLMs such as ChatGPT have strong language understanding and in-context learning capabilities, they can quickly be adapted to different types and styles of text. Thus, the LLM-based methods can be regarded as a series of zero-shot detectors, directly determining the source of text by analyzing its representation or probability distribution. The corresponding processing flow is roughly shown in Fig. 1(b). The first step is selecting a pre-trained large language model, such as GPT, or BERT, which requires carefully evaluating the performance and suitability of different pre-trained models to determine the most appropriate ones, and then obtaining the necessary APIs or model files to integrate the selected pre-trained model(s) into the system. The passage to be detected is then input into the selected pre-trained model and outputs the probability distribution or representation vector. Afterward, the system performs feature analysis to analyze the generated probability distribution and calculate the textual representation vector, and then a preset threshold or classifier is used to judge between fact and fantasy. Finally, the detection algorithm and threshold will be adjusted based on the detection results, and the detection strategy optimized to improve the accuracy.

Information Embedding: With the widespread application of AI-generated text in content creation, news publishing, and other fields, researchers have discovered that it is possible to ensure the transparency and traceability of the generated content by processing the embedded information of the text. Watermarking technology is the most forward-looking and cutting-edge detection method at present, by embedding hidden identifying information (such as specific words or character sequences) in the generated text, to achieve explicit verification of the source of the text (Jiang, Guo, Hu, & Gong, 2024; Sadasivan, Kumar, Balasubramanian, Wang, & Feizi, 2023). The corresponding processing flow can be roughly shown in Fig. 1(c). The first step of watermark detection is watermark embedding (Jiang et al., 2024; Jiang, Zhang, & Gong, 2023), which involves designing hidden identifying signals, such as specific vocabulary or character sequences, and seamlessly incorporating this identifying information into the generated text. At the same time, the generated identification information must be hidden and reliable. Then, the watermark extraction algorithm is designed to verify whether it conforms to the preset pattern of identification (Jalil & Mirza, 2009). Afterward, the extracted identification information is compared with the preset watermark library to validate if the text was produced by the target generation model and to trace the generation source. Finally, the goal of validation is to ensure the inserted watermark has a minor influence on the naturalness and readability of the text while maintaining the overall flow and logical coherence.

This paper presents a comprehensive survey of how existing AIGTD methods have addressed the aforementioned challenges, while suggesting potential directions for future research in this domain, as shown in Fig. 2. Correspondingly, our survey is developed as follows:

- Section 2 introduces the problem definition of AIGTD and its multi-level taxonomic framework. The taxonomy consists of three levels of indicators, organized based on temporal development, information transparency, and information access. We also categorize the detection method as either a black-box or white-box approach to facilitate retrieval of related work.
- Section 3 summarizes detection methods that enhance classification training in AIGTD. It covers black-box methods like GCN, MPU, RADAR, transfer learning, agent model training, and white-box methods that are mainly partial access. A comparative analysis highlights their strengths and weaknesses.

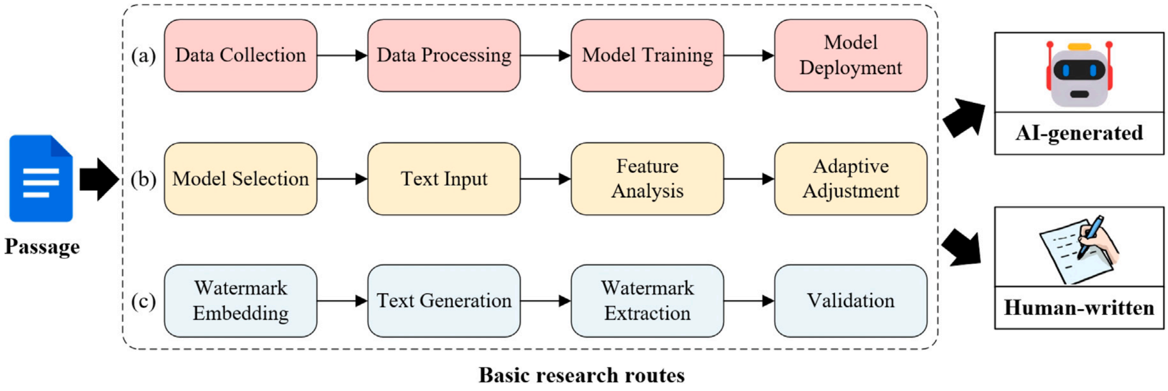

Fig. 1. Three key research directions in AIGTD, where (a), (b), and (c) describe the basic technical routes for tackling the challenges of classifier training, intrinsic attributes, and information embedding, respectively.

- Section 4 summarizes detection methods that leverage the inherent properties of pre-trained language models. These methods primarily focus on probability-based models with white-box partial access, including techniques such as average logarithm, probability divergence, and probability curvature.
- Section 5 summarizes existing methods that improve information embedding for AIGTD. Recent watermarking methods in the white-box context are roughly divided into three categories, i.e., training-free, training-based, and multi-bit embeddings.
- Section 6 provides the computational requirement.
- Section 7 provides the performance comparison and analysis for baseline methods.
- Section 8 introduces the benchmark datasets commonly used for AIGTD.
- Section 9 provides a conclusion and anticipates some promising directions for future AIGTD research.

2. Overview of aigtd

This paper presents a taxonomic framework to categorize the research challenges in AIGTD considering three key dimensions, i.e., temporal development, information transparency, and information access. While prior reviews have often classified challenges solely by the research timeline, our taxonomy provides a more nuanced and intuitive structure to contextualize the underlying technical approaches.

Considering the transparency and interpretability of the model, we further categorize the techniques as either black-box or white-box approaches (Tang, Chuang, & Hu, 2024; Wang, Mansurov, Ivanov, Su, Shelmanov, Tsvigun, Afzal et al., 2024). Black-box detection refers to systems without direct access to the internal mechanisms and details of the text generation model (Gao, Lanchantin, Soffa, & Qi, 2018). These detection methods can only analyze the input–output behavior and external characteristics of the generated text. In contrast, white-box detection implies that the system has a better understanding and access to the internal workings of the text generation model, including its architecture, training data, parameters, and generation process (Ebrahimi, Rao, Lowd, & Dou, 2018). This white-box approach inherently provides stronger interpretability and dependability, but may require more intimate knowledge of the target generation model. We also provide an overview of the computational requirement, as shown in Table 1.

Problem Definition. AIGTD is generally treated as a binary classification task, aiming to classify whether a candidate passage is human-written or AI-generated. Formally, given an input text x, an automatic detector $f(x) \in \mathbb{R}$ outputs a score, which is compared with a predetermined threshold θ for detection. If the output score exceeds the threshold $f(x) \geq \theta$, the text is identified as AI-generated. Conversely, the text is deemed to be human-written.

The subsequent sections will elaborate on the indicators at the sub-levels of this taxonomy, e.g., tackling classifier training (Section 3), tackling intrinsic attributes (Section 4), and information embedding (Section 5), respectively.

3. Tackling classifier training

Training-based classifiers are one of the first widely used methods for AIGTD (Rosenberg, Hebert, & Schneiderman, 2005). Enabled by the evolution of machine learning capabilities and natural language processing, especially the emergence of deep learning and pre-trained language models (e.g. BERT, GPT), their effectiveness has been improved significantly (Shrivastava, Gupta, & Girshick, 2016). These approaches aim to train a classification model to differentiate human-written text from AI-generated text, typically involving fine-tuning a pre-trained language model leveraging a collected dataset of binary data that includes human and AI-generated text. The key advantage of fine-tuned pre-trained models is their extensive training on large corpora, which imbues them with robust language understanding and generation capabilities, enabling better adaptation to specific tasks through fine-tuning (Gunel, Du, Conneau, & Stoyanov, 2021). *To cater to diverse application needs, the introduced methods can be sorted into black-box and white-box categories, as shown in the top of Table 1*. To address the issue of classifier training, the existing work generally employs feature analysis (Section 3.1), probability statistics (Section 3.2), and deep learning (Section 3.3), respectively.

3.1. Feature analysis

The method based on feature analysis mainly employs structural-based analysis, partial access, and network reconstruction to analyze and detect suspicious text. Thus, the structural features, some internal information, and network structure of the text can be obtained to exploit the characteristics of the text in AIGTD.

Structural-based Analysis. Structural information helps the model better understand the global characteristics and relationships of text data. Recent research has utilized graph convolutional networks (GCN) and tensor representations to build and analyze text structure. For example, Liu X et al. introduced the concepts of graph representation and structure entropy to discuss the model performance under the condition of data imbalance, to improve the accuracy of text classification and detection (Liu et al., 2022).

Partial Access. By analyzing the internal information of these models (such as logits, activation values, etc.), it is possible to effectively clarify the discrepancies between human and AI-generated text. A typical example is logits as waves (Xu, Hu, Gao, & Chen, 2022), which regarded logits in the process of model generation as a “waveform”, and detected the authenticity of the generated text by analyzing the characteristics of these waveforms. However, SeqXGPT (Wang

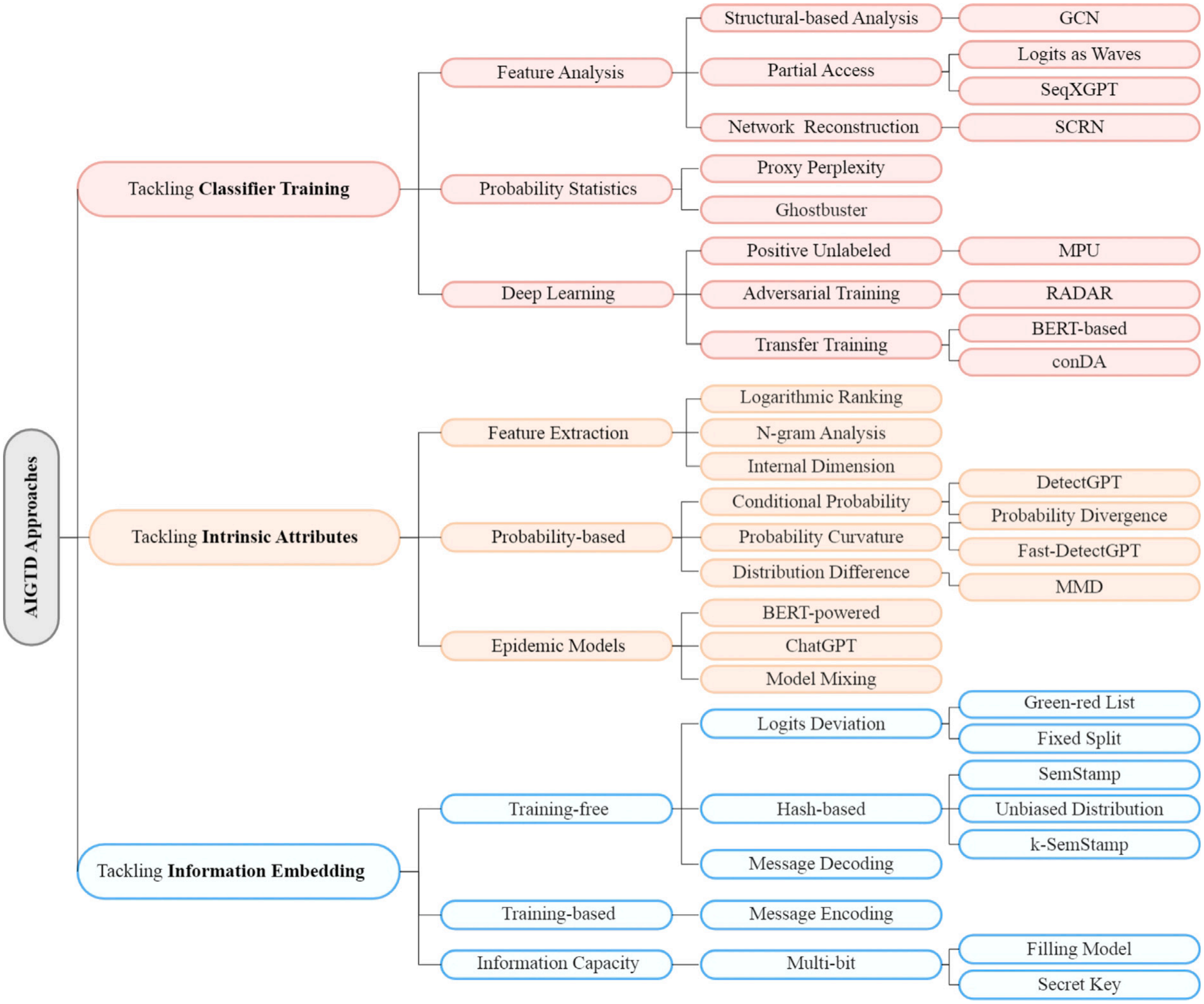

Fig. 2. Taxonomy of AIGTD methods.

et al., 2023) utilized the word-level logarithmic probability of the white-box language model as a feature, and processed these voice-like waveform timing features through convolutional networks and self-attention networks. This approach is also implemented on different data sets, and the results show that SeqXGPT outperforms existing methods in sentence-level and document-level AIGTD tasks, and shows strong generalization ability on out-of-distribution data sets.

Network Reconstruction. Interestingly, the SeqXGPT method mentioned above was quickly adopted by other researchers. To evaluate the robustness of the detector under mixed source generated text, Huang G et al. leveraged the SeqXGPT-Bench dataset and proposed a new model, the siamese calibrated reconstruction network (SCRN) (Huang et al., 2024). SCRN uses reconstruction networks to introduce and eliminate noise from text and extract semantic representations resilient to local perturbations, contributing to feature analysis.

In summary, structure-based methods are particularly effective at capturing structural features but have high computational costs and are sensitive to data imbalance. In contrast, partial access methods perform well by utilizing internal model outputs (e.g., logits) but are limited to white-box models. Finally, network reconstruction methods extract robust semantic representations, though they may suffer from information loss, affecting analysis accuracy.

3.2. Probability statistics

The method based on probability and statistics detects AI-generated text by analyzing the probability distribution and statistical characteristics of the text. These methods use probabilistic information in the process of text generation to determine the source of text.

Large language models (LLMs) generate text based on the probability distribution of the next word given the previous sequence. The proposed LLMDet collected text samples from diverse LLMs, constructing a dictionary of n-grams and associated next-word probabilities for each model (Wu, Pang, Shen, Cheng, & Chua, 2023). This probability distribution reflects the LLM’s capacity to predict the forthcoming word, which in turn correlates with the complexity of the input text. Accordingly, proxy model perplexity was proposed as an approximation, calculated as the negative log-likelihood of the text assessed by LLMs. The proxy perplexity of input texts was then calculated across the LLM set and incorporated as features into a text classification model, yielding the final probability distribution over source attributions.

Verma V et al. introduced Ghostbuster, a weak language model designed to detect AI-generated text, which predicted the likelihood of AI generation by training a linear classifier on features derived from a structured search of word probability vectors (Verma, Fleisig, Tomlin, & Klein, 2024). This method outperforms a series of prior weaker

Table 1
Overview of the computational requirement. We empirically estimated the overall cost for the following baselines, where $n(>1)$ in the Hardware column denotes the number of required GPUs. 'Black' and 'White' denote black-box and white-box, respectively.

Model	Black	White	Hardware (GPUs)	Software (LLM API)	Cost
GCN (Liu et al., 2022)	✓		8 × Nvidia A100 40 GB	GPT-3.5/4	High
Logits as waves (Xu et al., 2022)		✓	n × Nvidia V100 32 GB	–	Medium
SeqXGPT (Wang et al., 2023)		✓	n × Nvidia RTX 4090 24 GB	GPT-3.5-turbo	Medium
SCRN (Huang et al., 2024)		✓	8 × Nvidia V100 32 GB	–	High
Proxy perplexity (Wu et al., 2023)		✓	1 × Nvidia V100 32 GB	GPT-Zero	Medium
Ghostbuster (Verma et al., 2024)	✓		–	GPT-Zero	Medium
MPU (Tian et al., 2024)	✓		1 × Nvidia V100 32 GB	–	Medium
RADAR (Hu, Chen, & Ho, 2023)	✓		2 × Nvidia V100 32 GB	GPT-3.5-Turbo	Medium
BERT-based (Wang, Li and Li, 2024)	✓		1 × Nvidia RTX 3090 24 GB	–	Low
conDA (Bhattacharjee, Kumarage, Moraffah, & Liu, 2023)	✓		1 × Nvidia A100 40 GB	GPT-3.5-Turbo	High
Logarithmic ranking (Su, Zhuo, Wang and Nakov, 2023)		✓	4 × Nvidia A100 40 GB	–	High
N-gram analysis (Yang et al., 2024)	✓		2 × Nvidia A800 80 GB	GPT-3/3.5-Turbo/4	High
Internal dimension (Tulchinskii et al., 2024)		✓	2 × Nvidia V100 32 GB	GPT-3.5-Turbo/4/Zero	High
DetectGPT (Liu et al., 2024; Mitchell et al., 2023)	✓	✓	2 × Nvidia RTX 4090 24 GB	–	Medium
Probability divergence (Yang et al., 2024)		✓	2 × Nvidia A800 80 GB	GPT3/3.5-Turbo/4	High
Fast-DetectGPT (Bao, Zhao, Teng, Yang, & Zhang, 2024)	✓	✓	n × Nvidia A100 40 GB	GPT-3.5-Turbo/4/Zero	High
MMD (Zhang, Liu et al., 2024)	✓		1 × Nvidia A800 80 GB	GPT2/3/Neo/4	High
BERT-powered (Chakraborty, Gheewala, Degadwala, Vyas, & Soni, 2024)	✓		1 × Nvidia RTX 4090 24 GB	–	Low
ChatGPT (Markowitz, Hancock, & Bailenson, 2024)	✓		2 × Nvidia V100 32 GB	GPT-3.5-Turbo	Medium
Model mixing (Mo et al., 2024)	✓		1 × Nvidia RTX 3090 24G	–	Low
Green-red list (Kirchenbauer et al., 2023)		✓	–	–	Low
Fixed split (Zhao, Ananth, Li, & Wang, 2024)		✓	n × Nvidia A100 40G	GPT3	High
SemStamp (Hou et al., 2023)		✓	8 × Nvidia A40 48 GB	GPT-3.5-Turbo	High
Unbiased distribution (Wu, Hu, Zhang, & Huang, 2024)		✓	3 × Nvidia A6000 48 GB	GPT-4-0613	High
k-SemStamp (Hou, Zhang, Wang, Khashabi, & He, 2024)		✓	8 × A40 48 GB and 4 × A100 40 GB	GPT-3.5-Turbo	High
Message decoding (Zhao, Li and Wang, 2024)		✓	n × Nvidia A600 16 GB	–	High
Message encoding (Zhang, Hussain, Neekhara and Koushanfar, 2024)		✓	n × Nvidia A6000 48 GB	GPT-4	High
Filling model (Yoo, Ahn, Jang, & Kwak, 2023)		✓	1 × Nvidia RTX 3090 24 GB	–	Low
Secret key (Fernandez, Chaffin, Tit, Chappelier, & Furon, 2023)		✓	1 × Nvidia A800 80 GB	–	High

language models based on unigram, trigram, etc., by structured searching over combinations of these model outputs, which improves the generalization ability to detect unknown AI-generated text. However, it is often inferior to existing advanced methods based on large language models due to limitations in architecture, training data, computation, or algorithms, hindering the capture of complex language patterns and the generation of high-quality text.

In summary, these methods enhance speed and security by using precomputed probability data to track the origin of text or improve robustness to unknown prompts without requiring token probabilities. However, both methods share limitations in multilingual and stylistic versatility.

3.3. Deep learning

The deep learning paradigm has emerged as a prominent approach for AIGTD. Deep learning-based AIGTD methods leverage complex neural models trained on large data to learn effective discriminative features, outperforming traditional approaches reliant on manual feature engineering.

Positive Unlabeled. To address the challenge of performing AIGTD on short inputs without compromising performance on longer passages, Tian Y et al. proposed a multi-scale positive unlabeled (MPU) training framework (Tian et al., 2024). The core principle of MPU is to formulate the task as a partially positive-unlabeled (PU) problem and introduce a length-sensitive multi-scale PU loss function to jointly optimize the detector's performance on both short-form and long-form text inputs.

Adversarial Training. Texts generated by generative adversarial networks (GANs) are highly realistic and syntactically coherent, making them challenging to accurately distinguish from human-written text using traditional statistical or rule-based detection methods. GANs can learn and model complex data distributions, including the syntax and semantics of natural languages. This capacity allows GAN-generated text to closely mimic human-written content, making it challenging for detectors to reliably distinguish between AI-generated and human-authored text based on subtle differences alone. Although advanced trained detectors have achieved commendable performance on unseen test data, emerging studies highlight vulnerabilities in high-performing text detectors faced with adversarial tactics like paraphrasing (Zhou, He, & Sun, 2024). Hu X et al. proposed a RADAR framework that utilized adversarial learning techniques to train a robust AI text detector (Hu et al., 2023). The adversarial training between a rewriter and a detector further enhances robustness against text rewriting techniques compared with traditional statistical methods, e.g., log probability, rank, log rank, entropy, etc. Overall, generative models and adversarial attack techniques are constantly updated, making detection methods need to be updated promptly to maintain their effectiveness and accuracy. While existing text detectors exhibit robust performance, they still display notable vulnerabilities when confronted with adversarial attacks, and research in this domain remains limited.

Transfer Training. Media outlets wield a crucial influence in molding public sentiment and providing timely information. Some well-trained models can even predict the direction of public opinion. News outlets play a crucial role in shaping public discourse and disseminating timely information. Some well-trained models can even predict the direction of public opinion with notable accuracy (Chu, Andreas, Ansolabehere, & Roy, 2023; Hanley & Durumeric, 2024). Wang et al. built an efficient text detection model based on BERT (Wang, Li et al., 2024), which is pre-trained on an extensive training corpus and has excellent generalization performance in AIGTD. Facing the challenge of obtaining annotated data, Bhattacharjee A et al. proposed a conDA model through unsupervised domain adaptive and self-supervised representation learning (Bhattacharjee et al., 2023). This leverages labeled data from the source domain along with unlabeled data from the target domain, enabling effective transfer learning, migrating labeled data from source language models to unlabeled target data, enabling the learning of domain-invariant representations. Contrast learning is further used to enhance the presentation of text. The positive sample is usually obtained by transforming the original text, and the negative sample is formed with other samples so that the model can learn to distinguish the representation of the original text and the transformed text, and finally improve the detection performance of the data in the target domain.

In summary, positive unlabeled learning, particularly MPU, effectively detects short texts while maintaining performance on longer ones, though it relies heavily on incomplete data. Adversarial training methods like RADAR improve robustness against sophisticated text generation but may struggle with native LLM-generated texts, requiring continuous updates to remain effective. Additionally, conDA's integration of domain adaptation and contrastive learning shows promise but lacks explainability. However, further research is needed to improve their robustness, adaptability, and interpretability across various text formats.

4. Tackling intrinsic attributes

Leveraging the inherent attributes of large pre-trained language models, such as their probability distributions and representation vectors, to detect AI-generated text is an effective white-box approach that circumvents the need for specialized training data. To address this challenge, a variety of efficient detection algorithms have been developed, with many recent methods incorporating zero-shot techniques to some degree. A brief introduction to the use of this technology in AIGTD will be provided.

The zero-shot technology is currently experiencing rapid development, aiming to enable models to identify, classify, or understand categories or concepts that are not included during training (Pourpanah et al., 2022). The basic principles of the zero-shot technology can be divided into three main aspects: knowledge transfer, attribute learning, and semantic embedding. Specifically, the core of zero-shot learning lies in the ability to transfer knowledge learned from training data (such as patterns, features, or relationships) to previously unseen categories (Wang, Zheng, Yu, & Miao, 2019). In the process of zero-shot learning, attributes are often used as intermediaries, and these attributes are shared between the training classes and the previously unseen classes. Finally, semantic embedding is utilized to establish the relationship between these categories.

In a zero-shot setting, the discriminator can effectively differentiate AI-generated text from human-written content by drawing upon the inherent distinctions between the two, circumventing the need for extensive training on labeled data (Wang et al., 2019). The paramount advantage of zero-shot detection is its proficiency in adapting to new data, eliminating the requirement for supplementary data collection or model optimization. Although watermarking approaches may be considered zero-shot, we treat them as a separate category in this discussion. *Furthermore, the introduced methods can also be sorted into black-box and white-box categories, as shown in the middle of Table 1.* To address the issue of intrinsic attributes, the existing work generally uses feature extraction (Section 4.1), probability-based (Section 4.2), and epidemic model (Section 4.3), respectively.

4.1. Feature extraction

Feature extraction is the process of extracting useful information from raw data. In AIGTD, relevant text features can include lexical frequency, grammatical structure, and contextual cues, etc. Employing diverse feature extraction methods can capture the distinctive characteristics and behavioral patterns of AIGTD models, thus contributing to distinguishing human-written text from AI-generated text.

Logarithmic Ranking. Su J et al. leveraged a large language model to calculate the log-likelihood log-rank ratio (LRR) of each word in the target text (Su, Zhuo et al., 2023). They obtained the LRR by using the complementary information of log-likelihood and log-rank, and took the average LRR of all words in the target text as the feature for detection. Since logarithmic likelihood reflects the absolute confidence of the model for the correct word and logarithmic ranking reflects the relative confidence, thus the combination of them can better characterize the text.

N-gram Analysis (BScore). In addition to comparing ranking, Yang X et al. utilized N-gram distances to calculate a BScore (Yang et al., 2024). Specifically, given a piece of text, they divided it into two parts, used only the first half as input, and then allowed a pre-trained language model to generate the second half. The resulting BScore is often used to determine whether the text was machine-generated.

Internal Dimension. Recently, the popular black-box unknown source detection method has primarily been the single feature classifier for internal dimension proposed by existing work (Tulchinskii et al., 2024). They argued that text samples written by humans had roughly the same intrinsic dimensions in a given language, whereas text samples generated by modern large language models had lower intrinsic dimensions on average. Therefore, they used persistent homology theory in topological data analysis to estimate the intrinsic dimension of the point cloud formed by the embedding of each text sample in Euclidean space. Their work found that the intrinsic dimensions of text written by humans were significantly higher, by an average of 2–3 points, compared with those generated by AI. This finding presents a novel avenue for advancing black-box unknown source detection techniques.

In summary, these methods can effectively distinguish between AI-generated and human-written texts by identifying intrinsic features. However, when used with modern LLMs, Logarithmic Ranking requires large perturbation functions and multiple perturbations, N-gram relies on the ability to generate new text based on a given prompt, and the internal dimension is restricted to PH dimension, small generation temperature, and high-resource domain.

4.2. Probability-based

Conditional Probability. Traditional conditional probability approaches typically rely on analyzing the behavior and inherent properties of a text generation model under specific contextual conditions, such as predicting the probability distribution of the next word given a particular input context. In contrast to general probabilistic analysis, this more targeted conditional methodology can provide a more granular and accurate assessment of the underlying text generation process. Mitchell E et al. introduced the probability curve and the DetectGPT framework (Mitchell et al., 2023), which first leveraged a generic pre-trained language model, such as T5 (Raffel et al., 2020), to introduce minor perturbations to the candidate text, and then compared the average logarithmic probability of the original and perturbed text under the target model. If the logarithmic probability of the disturbed text was significantly lower than that of the original text, it was likely that the text came from the target model, without the need to train specialized detectors or collect training data.

However, the random perturbation strategy employed in DetectGPT introduces unwanted noise, while the reliance on logit regression thresholds in that approach compromises the generalization and applicability of the method to single or small batches of inputs. To address these limitations, Liu S et al. proposed a novel fine-tuning detector, Pecola, to connect the merits of metric-based detectors and fine-tuned detectors through a contrast learning framework that leveraged selective perturbation strategies (Liu et al., 2024). Specifically, the selective perturbation strategies employed by Pecola retain significant tokens during the perturbation process and maintain weights for multiple pairs of contrast learning, thereby enhancing the generalization capabilities of the detector.

Probability Curvature. Previous research has established that AI-generated text often resides in the regions of negative curvature in the probability function. To detect whether a given text originates from a specific model, Mireshghallah F et al. have demonstrated that large models are more conservative in assigning curvature and likelihood values to the generated text of other models (Mireshghallah, Mattern, Gao, Shokri, & Berg-Kirkpatrick, 2023). Leveraging this observation, they proposed using smaller models, such as OPT-125M (Zhang et al., 2022), as proxies for a more comprehensive detection approach. The rationale is that these smaller models can assign higher curvature values to text generated by models of the same size or larger, thereby enabling more effective identification of AI-generated content.

Moreover, Mitchell E et al. have observed that LLM-generated text tends to occupy the regions of negative curvature through the model's logarithmic probability function (Mitchell et al., 2023). Thus, they introduce conditional probability divergence to assess if the text was produced by the target model based on the perturbation deviation. The perturbation deviation refers to the logarithmic probability change of the text in the LLM before and after applying a random perturbation, where the perturbation is guided by the curvature criterion of the model's probability function. However, the results of this exercise are not always accurate. Afterward, Yang X et al. proposed conditional probability curvature based on conditional probability divergence, utilizing WScore (Yang et al., 2024). They split a passage into two parts and then inputted the first half into the model to generate the second half, judging the category of the text by the probability difference between the generated text and the original second half.

Bao G et al. achieved significant advancements over the previous DetectGPT approach (Bao et al., 2024). Specifically, they used conditional probability to build a more effective zero-emission detector and replaced the perturbation-based probability measure with conditional probability curvature, which utilized sampling instead of the original perturbation step. These modifications have resulted in their detector being substantially more accurate than DetectGPT, in both white-box and black-box settings. Besides, their approach achieved a hundredfold acceleration compared to the original DetectGPT method for AIGTD.

Distribution Difference. Another promising approach to differentiate AI-generated text from human-written text is to identify the maximum mean discrepancy (MMD) (Gretton, Borgwardt, Rasch, Schölkopf, & Smola, 2012) between the two distributions. However, training a detector directly with MMD on text produced by diverse language models has the potential to significantly amplify MMD variance. As diverse language models may be employed, including multiple text populations would significantly impair MMD's facility for evaluating the difference separating two probability distributions. To address this problem, Zhang et al. proposed MMD-mp, a multi-population-aware optimized MMD (Zhang, Liu et al., 2024), where paragraph-based and sentence-based methods were used to improve the accuracy and stability of measuring the distribution differences between AI-generated and human-written text. Additionally, it addresses the challenge of variance in AI-generated data from multiple sources, serving as a valuable tool for content authenticity and mitigating the risks associated with AI-generated texts.

In summary, these methods enhance the precision and generalization capabilities of AIGTD systems based on conditional probabilities, probability curvature, and distribution differences. However, conditional probability methods generally require a white-box assumption and a reasonable perturbation function. Probability curvature achieves high AUC performances in detecting machine-generated text but does not guarantee effectiveness across all models. Although distribution-based approaches assess distribution differences between AI-generated and human-written text to improve detection stability, there is still a high time complexity.

4.3. Epidemic models

BERT & ChatGPT. Chakraborty et al. proposed an approach that leveraged contextual embeddings from the BERT language model to accurately identify AI-generated text (Chakraborty et al., 2024), which was designed to uncover complex patterns within the text that can serve as indicators of AI sources. Moreover, Markowitz et al. examined the differences in linguistic properties between AI-generated and human-generated texts, where ChatGPT was utilized as a comparative reference, evaluating aspects such as content (emotional expression), style (analytical writing, adjectival usage), and structural features (readability level) (Markowitz et al., 2024). Despite achieving over 80% accuracy in distinguishing AI-generated from human-generated texts, AI-generated content generally displays a more analytical tone, heightened descriptiveness, and reduced readability relative to handwritten material. This suggests that the distinct stylistic and structural properties of AI-generated text, despite its high accuracy, can serve as differentiating features from human-written content.

Model Mixing. To improve the detection performance, Mo et al. presented an effective tool for detecting AI-generated text (Mo et al., 2024). The detection methods utilized deep learning techniques, integrating layers of Transformer, long short-term memory (LSTM), and convolutional neural network (CNN) for effective text classification and sequence labeling tasks. Besides, the preprocessing steps were comprehensive, involving Unicode normalization, case conversion, removal of non-alphabetic characters and extra whitespace, and specific delimiter connections. Overall, this rigorous preprocessing ensures the data fed into the model is clean and consistent, and the thorough methodology and promising results underscore the potential for broad application and future advancements in AIGTD.

In summary, recent models like BERT and ChatGPT enhance AI-generated text detection by leveraging semantic and linguistic features. Additionally, model mixing techniques that combine various architectures may further improve detection performance, contributing to more robust and adaptable systems across domains.

5. Tackling information embedding

The development of detection techniques based on information embedding has advanced significantly. Detecting or analyzing specific markers embedded during the text generation process contributes to determining whether the text is generated by AI. One significant application of this approach is watermark detection technology, which incorporates a watermark, such as a bit string, into LLM-generated text, facilitating the identification of AI-authored content and its tracing to a particular model or user. Traditional watermark detection methods are relatively straightforward, as they determine the presence of a watermark by extracting a fixed set of semantic features (Topkara, Taskiran, & Delp III, 2005; Topkara, Topkara, & Atallah, 2006; Yang et al., 2023, 2022). However, with the continuous advancements in generation technology, detecting specific information markers is becoming increasingly challenging. A major limitation of current information embedding detection techniques is their inability to accurately and effectively extract watermarks from text, particularly when the watermark consists of a long bit string.

To address the extraction and analysis of directly embedded information, Qu et al. employed error correction codes to enhance the accuracy and robustness of watermark extraction in the context of LLM-generated text (Qu et al., 2024). This approach maintains high accuracy even under bounded adversarial text/token editing (insertion, deletion, and replacement), thereby providing a reliable method for countering adversarial text interference. *Since the core idea of information embedding is to embed a distinctive signal in the content and enable subsequent detection of the "watermark", these methods are generally regarded as white-box approaches.* This study underscores the importance of developing detection techniques for indirect information embedding, which addresses more complex and nuanced challenges in the field. To address the issue of information embedding, the existing work generally employs training-free (Section 5.1), training-based (Section 5.2), and information capacity (Section 5.3), respectively.

5.1. Training-free

A notable feature of training-free watermarking detection methods is that they do not require embedding watermarks during the model training phase. Instead, these methods typically embed the watermark after the model has been trained by directly modifying the model parameters or adding additional structures. The primary advantage of this approach is its ability to quickly embed and detect watermarks, making it particularly suitable for the rapid deployment and application of pre-trained models. Moreover, these methods often achieve watermark embedding without sacrificing the model's performance, offering high practicality and flexibility.

Logit Deviation. To study the watermarking of model outputs, Kirchenbauer et al. proposed the logit deviation with the green-red list (Kirchenbauer et al., 2023), which doesn not necessitate access to the model's internal parameters. Instead, this method directly analyzed the generated text to determine whether the words conform to predefined green and red lists. Initially, the entire vocabulary was divided into two subsets: the green list, containing a randomly selected subset of words, and the red list, containing all remaining words. During text generation, the original logit distribution was modified. For each term in the green list, a small offset was added to its logit, while the logit of terms in the red list remained unchanged. Such a mechanism increases the likelihood of the model selecting words from the green list. So the generated text was more likely to contain words from the green list by sampling from this modified logit distribution. If most of the words in the generated text were from the green list, it could be inferred that the text is AI-generated. Building on this foundation, Zhao X et al. proposed a logit deviation with the fixed splits, which simplified the scheme by utilizing a constant green-red list separation while maintaining the guaranteed generation quality of the watermark, improving its robustness to text editing (Zhao et al., 2024).

Hash-based. Existing watermarking methods are vulnerable to paraphrase attacks owing to their token-level structure, thus Hou et al. introduced a sentence-level semantic watermarking algorithm SemStamp that leverages locality-sensitive hashing (LSH) to divide the semantic space of sentences (Hou et al., 2023). The approach worked by encoding the candidate sentence generated by a large language model and then applying LSH hashing to it, and then performing sentence-level rejection sampling until the sampled sentence lies inside the watermarked partitions in the semantic embedding space. Finally, a margin-based constraint was applied to enhance the robustness of the watermark against paraphrastic changes. Wu Y et al. proposed an unbiased, distribution-preserving watermarking technique called the reweight strategy with ciphers (Wu et al., 2024), which preserved the original token distribution during the watermarking process by allocating unique, i.i.d. weights based on the context, using a heavy-weight strategy. The hash function, which is robust to moderate token changes, maintains the distribution of the original text, rendering the watermarked text indistinguishable from the unwatermarked version.

As previously noted, the watermark generation algorithm injects detectable signatures into the language generation process. However, this token-level watermarking approach is vulnerable to interpretation attacks. Although the proposed robust watermarking solution SemStamp (Hou et al., 2023) contributes to addressing this problem, it suffers from a suboptimal trade-off between speed and resilience due to the arbitrary hyperplane divisions in the underlying LSH technique. Thus, Hou et al. further proposed k-SemStamp, which represents a streamlined and impactful improvement upon the existing SemStamp watermarking solution (Hou et al., 2024). In addition, K-means clustering was utilized as a substitute for LSH to divide the embedded space, facilitating a more accurate representation of the underlying semantic structure and improving the trade-off between speed and resilience compared with the original LSH-based approach.

Message Decoding. Zhao et al. proposed a novel permutation-and-flip (PF) decoder, which demonstrates higher robustness during the sampling phase (Zhao, Li et al., 2024). Additionally, they have designed a cryptographic watermarking framework akin to Aaronson's Gumbel watermarking (Aaronson, Liu, Liu, Zhandry, & Zhang, 2021), but specifically tailored for the decoders. This framework significantly outperforms naive sampling and its Gumbel watermarking counterpart regarding confusion while preserving the original sample distribution. In addition, the PF method can be combined with other advanced decoding techniques, e.g., Bayes minimum risk, constrained decoding, etc.

In summary, these methods provide a flexible and efficient approach to embedding watermarks in AI-generated text without modifying the model during training. Logit deviation improves robustness against text editing by simplifying green-red list separation, though its reliance on a fixed segmentation may not be universally optimal. Hash-based methods enhance attack resilience by leveraging hash functions, balancing speed and robustness. Additionally, message decoding methods bolster robustness during sampling and can be integrated with advanced decoding techniques to improve text quality and diversity.

5.2. Training-based

Training-based watermarking refers to the technique of embedding a watermark during the model training process. This method adjusts the training data or training process to imbue the model with specific watermark information. For example, modifying the training dataset by adding watermark samples, and incorporating watermarking-related penalties or regularization terms into the training loss function, guiding the model to embed the watermark more effectively (Fang, Jia, Zhou, Ma, & Zhang, 2022). These strategies ensure that the watermark is an integral part of the model, allowing for reliable detection and verification of the model's outputs.

Message Encoding. Zhang R et al. proposed message encoding with parameterization, involving generating a dense token distribution by injecting a watermark into the message encoding module (Zhang, Hussain et al., 2024). The message decoding module was used to extract the signatures from the watermarked texts, and the reparameterization technique functioned as a link, connecting the dense distribution and the one-hot encoding of the token. Training-based watermarking shows promise but is limited by its training requirement for source data, which may hinder generalization to unseen text.

In summary, these inject the watermark into the message encoding module, ensuring that the watermark becomes an integral part of the model. While promising, training-based watermarking faces limitations, particularly the requirement for source data, which may restrict its generalization ability.

5.3. Information capacity

The classification dimension of information capacity pertains to the amount of information that can be embedded using watermarking technology. This technology can be further subdivided based on the quantity and complexity of the embedded information. A key challenge in this approach is to maximize the embedding of useful information without significantly compromising the quality of text generation. Additionally, it is crucial to ensure that the embedded information remains intact and valid despite various potential attacks and modifications.

Multi-bit. Multi-bit watermarking enables the embedding of multi-bit information, allowing for the incorporation of rich information into the model rather than just a simple identifier. This approach is compatible with the previously mentioned methods and is suitable for scenarios requiring the embedding of large amounts of information or complex copyright protection and tracking. Yoo et al. proposed a corruption-resistant filling model and designed a multi-bit watermark called invariant features (Yoo et al., 2023), which followed a well-known proposition of image watermarking, recognizing natural language features that are invariant to slight damage. This approach

Table 2
Performance comparison of methods shown in Fig. 2 across commonly used datasets. Bold values indicate the best performance.

Model	HC3			GPT-2-human		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
GCN[a]	–	–	–	–	–	94.52
SeqXGPT	**99.70**	**99.75**	**99.72**	**97.52**	**96.61**	**97.06**
Proxy perplexity	79.52	79.19	79.35	76.13	75.00	74.75
MPU	98.36	96.98	97.67	–	–	–
BERT-based	99.10	98.86	98.98	95.22	94.78	94.89
Model	**Xsum**			**WritingPrompts**		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
Logarithmic ranking	72.17	88.20	79.38	82.87	90.00	86.28
DetectGPT (10 Perturbations)	94.25	94.07	93.96	96.82	96.96	96.39
DetectGPT (1 Perturbation)	71.12	72.65	71.58	81.20	81.63	81.42
Fast-DetectGPT (Neo-2.7/Neo-2.7)	84.68	83.00	82.79	91.71	91.67	91.66
BERT-powered	**96.82**	**96.80**	**96.80**	**98.23**	**98.23**	**98.23**
Model mixing	86.19	82.03	84.06	89.18	90.12	89.65
Model	RealNews			BookSum		
	AUC (%)	TP@1 (%)	TP@5 (%)	AUC (%)	TP@1 (%)	TP@5 (%)
Fixed split (+GPT2)	89.4	63.3	67.7	99.3	86.7	87.3
SemStamp (+Pegasus)[a]	97.8	83.7	92.0	99.2	90.1	96.8
k-SemStamp (+Pegasus)[a]	**99.5**	**92.7**	**96.5**	**99.3**	**94.1**	**97.3**
SemStamp (+GPT3.5)[a]	83.3	33.9	52.9	89.6	45.6	62.4
k-SemStamp (+GPT3.5)[a]	90.8	55.5	71.8	95.6	65.7	83.0
Model	WikiText-2					
	WER (%)		Semantic similarity		BLEU-4	
Message encoding (4 bits)[a]	**97.23**		0.92		**0.33**	
Message encoding (8 bits)[a]	89.57		0.89		0.23	
Message encoding (16 bits)[a]	76.37		0.89		0.19	
Filling model[a]	–		**0.99**		–	

[a] Indicates the performance reported in the original paper.

achieves robustness against multiple types of disturbances by utilizing the semantic and syntactic invariant features of the text, providing a high payload and offering an effective solution for text watermarking. Furthermore, Fernandez et al. utilized a secret key per message to enhance the watermarking of LLM leveraging statistical analysis methods and advanced multi-bit watermarking approaches (Fernandez et al., 2023). This approach further strengthens the integrity and reliability of embedded watermarks.

Afterward, Boroujeny et al. proposed a non-distortion watermarking method that requires a key for watermark detection (Boroujeny, Jiang, Zeng, & Mark, 2024). Unlike traditional zero-bit watermarks, which only mark text as AI-generated, this method embeds multi-bit information, enhancing informational capacity. They also developed an optimized decoder that captures embedded signals from the watermark information with a low error rate, improving practicality and effectiveness.

In summary, these methods allow for embedding richer information beyond simple identifiers, playing a critical role in preserving the robustness and information capacity of watermarked models. However, they may struggle with semantic accuracy, adversarial robustness, paraphrasing, and potential factual errors undetected by standard metrics.

6. Computational requirement

As AI models advance in complexity and capability, the associated computational requirements, such as hardware and software, have become increasingly critical. We have carefully collected the GPU and LLM API parameters that users generally care about and provided empirically estimated overall costs for baseline methods, as shown in Table 1. We customize the following rules: (1) A single GPU of 24G or less memory is considered Low, while multiple such GPUs are classified as Medium; (2) A single GPU of 32G memory is Medium, while multiple such GPUs are High; (3) A single or multiple GPUs of 40G or more memory are considered High; (4) Due to the high cost of the GPT-4 API, any use of GPT-4 under the above rules is also classified as High. Overall, the cost of the baseline method for classifier training is lower than that of the other two categories, and the prevailing trend in this task is to use GPUs of larger memory capacities and more powerful LLM APIs. Please refer to Table 1 for more details.

7. Performance comparison and analysis

For the performance comparison and analysis, we have gathered or reproduced the performance of the representative baselines shown in Fig. 2, as detailed in Table 2. We use precision (P), recall (R), and F1 score metrics for HC3, GPT-2-human, Xsum, and WritingPrompts, and adopt the area under the receiver operating characteristic curve (AUC) and the true positive rate (TP) when the false positive rate is at 1% and 5% (TP@1%, TP@5%) for RealNews and BookSum. Besides, we further adopt the watermark extraction rate (WER), semantic similarity, and BLEU-4 metrics for WikiText-2.

To evaluate methods for addressing the classifier training issue, we selected the HC3 and GPT-2-human datasets, with results presented at the top of Table 2. Seqxgpt achieves the state-of-the-art performance in binary classification by leveraging the strengths of both Transformer and CNN architectures, where the Transformer captures contextual features, and CNN extracts local features. Other Transformer-based methods, e.g., MPU and BERT-based, further stress the importance of contextual semantic features, and the linguistic structure is also effective for this task, e.g., GCN.

To evaluate methods for addressing the intrinsic attributes issue, we selected Xsum and WritingPrompts datasets, with results shown at the middle of Table 2. For these datasets, we selected the first 30 words of each sample as prompts for GPT-2 to obtain the corresponding AI-generated text. Interestingly, BERT-powered outperformed other baselines, including GPT-based methods, demonstrating that BERT has a more significant advantage in explicit and straightforward tasks. GPT-based methods achieve promising performances thanks to the powerful reasoning ability of LLMs, and increasing the number of perturbations

Table 3
Overview of commonly used datasets for AIGTD in recent years. – denotes the size of datasets is not available.

Datasets	Source	Size	Data Description
TuringBench	Uchendu, Ma, Le, Zhang, and Lee (2021)	200,000	10K news articles and 10K articles generated by 19 AI text-generators
HC3	Guo et al. (2023)	37,175	Responses from human experts and ChatGPT across open-domain, financial, etc.
GPT-2-human	Wang et al. (2023)	70,000	A collection of GPT-2-generated text and human-written text
WritingPrompts	Fan, Lewis, and Dauphin (2018)	303,358	Human-written stories paired with writing prompts from an online forum
CHEAT	Yu, Chen, Feng, and Xia (2023)	35,304	35,304 synthetic abstracts, including Generation,Polish and Mix
OpenGPTText	Chen et al. (2023)	29,395	GPT-3.5-turbo paraphrases of the OpenWebText dataset
MGTBench	He, Shen, Chen, Backes, and Zhang (2023)	3000	Containing three enhanced datasets, i.e., Essay, WP, and Reuters
HC3 Plus	Su, Wu, Zhou, Ma and Hu (2023)	214,498	Summarization, Translation, and Paraphrasing
MULTITuDE	Macko et al. (2023)	74,081	Authentic and AI-generated texts in 11 languages generated by 8 LLMs
M4	Wang, Ivanov, Su, Shelmanov, Tsvigun, Whitehouse et al. (2024)	247,000	Multi-generator, Multi-domain, and Multi-lingual
M4GT-Bench	Wang, Mansurov, Ivanov, Su, Shelmanov, Tsvigun, Afzal et al. (2024)	138,465	Multi-lingual, Multi-domain, and Multi-generator
GTD[a]	Mo et al. (2024)	1378	Open-source dataset, including manually written texts and AI-generated texts
Reviews24[a]	Markowitz et al. (2024)	1200	Self-built hotel reviews produced by humans and GPT-3.5
OpenGen	Krishna, Song, Karpinska, Wieting, and Iyyer (2023)	3000	3000 randomly selected two-sentence blocks
Alpaca	Taori et al. (2023)	52,000	Instructions and demonstrations generated by OpenAI's engine
C4	Raffel et al. (2020)	–	English corpus, used for text generation tasks
C4 News	Qu et al. (2024)	–	A variant of C4 contains 15 GB news crawled from the internet
XSum	Narayan, Cohen, and Lapata (2018)	226,711	BBC articles and accompanying single sentence summaries
WikiText-2	Merity, Xiong, Bradbury, and Socher (2017)	720	Wikimedia corpus for training and evaluating language models
WikiText-103	Merity et al. (2017)	28,595	Wikimedia corpus for training and evaluating language models
LWD	Soto et al. (2024)	234,593	Text generated using Llama-2, GPT-4, and ChatGPT
AAC	Soto et al. (2024)	1,259,286	Text generated by GPT-2 and OPT models
HEIs[a]	Perkins, Roe, Postma, McGaughran, and Hickerson (2024)	–	963 student submissions of essays, reports, and case studies
DAIGT	Lai, Zhang, and Chen (2024)	44,206	The ratio of human-written text to LLM-generated text is 2:1
Deepfake	Lai et al. (2024)	1562	LLM-generated text encompassing broader domains
BookSum	Kryściński, Rajani, Agarwal, Xiong, and Radev (2022)	12,515	A collection of datasets for long-form narrative summarization
RealNews	Zellers et al. (2019)	–	Filtering C4 to only include news content
Creative Writing	Verma et al. (2024)	7000	Creative writing based on community tips
Student Essay	Verma et al. (2024)	7000	Essays based on the British Academic Written English corpus
News	Verma et al. (2024)	7000	Based on the Reuters 50-50 authorship identification dataset
Code	Mao, Vondrick, Wang, and Yang (2024)	328	GPT-written Python code detection with HumanEval dataset
Yelp Review	Mao et al. (2024)	4000	Raw Yelp reviews and AI-generated reviews via GPT-3.5-turbo
ArXiv Paper	Mao et al. (2024)	700	Contains 350 abstracts of ICLR papers from 2015 to 2021

[a] Denotes the dataset name is not provided in the original paper, but is assigned here.

may contribute to the final performance. This is also useful for zero-shot detection, e.g., Logarithmic Ranking. Moreover, simply combining epidemic methods may fail to produce effective results, requiring more robust strategies for improvement.

To evaluate methods for addressing the information embedding issue, we selected RealNews, BookSum, and WikiText-2 datasets, with results presented at the bottom of Table 2. We paraphrased the watermarked generation for each sentence using the Pegasus, GPT-3.5, and GPT-2 models, potentially varying in task difficulty. In the upper part, k-SemStamp achieves the state-of-the-art performance, indicating the advantages of combining locality-sensitive hashing and clustering in partitioning the embedding space. In addition, the simplified fixed grouping strategy proposed by Fixed Split is also effective for robust watermarking. In the lower part, the Filling Model outperforms Message Encoding, demonstrating the importance of using invariant features of natural language to embed robust watermarks against corruption. Additionally, the inserted signature length is increased from 4-bit to 16-bit. Semantic similarity remains stable as the watermark length increases; however, WER and BLEU-4 decline significantly, indicating that longer watermark insertions impact semantic integrity.

8. Benchmark datasets

To support research in this area, several benchmark datasets have been developed to evaluate the performance of AIGTD algorithms. As illustrated in Table 3, we have identified the commonly used detection datasets in recent years, including the name, source, size, and data description of datasets. For example, OpenGPTText (Chen et al., 2023), sourced from open web text, is one of the most frequently utilized datasets. It is often used to evaluate a model's generalization capability across a spectrum of unrestricted text sources. These datasets typically comprise a collection of human-written and AI-generated text samples spanning various domains and genres, enabling the development, evaluation, and comparison of novel detection algorithms. The availability of such comprehensive benchmarks has been instrumental in driving progress and innovation in this research task.

9. Conclusion and future research

Based on an in-depth investigation of AIGTD methods, this paper comprehensively summarizes and discusses the advancements of AIGTD technology in recent years, and constructs the classification topology of AIGTD based on addressing three key challenges: classifier training, inherent model attributes, and information embedding. Additionally, we categorize these technologies into black-box and white-box approaches, and provide the computational requirements for each baseline to facilitate quick reference for researchers across various fields.

Although there has been significant progress in AIGTD through machine learning, large language models, and innovative watermarking techniques, interpreting and understanding these outputs remains challenging. This limitation affects the applicability and reliability of black-box detection methods in practical applications. Furthermore, the continuous advancement of generative models has led to the emergence of adversarial generative texts, which complicate traditional detection methods. These adversarial texts can deceive existing detectors, thereby increasing the false-negative rate. In this context, we try to provide potential future research directions for the field as follows:

For classifier training. In the present study, there was little detection of adversarial generated text. However, as adversarial networks continue to grow in popularity, future research will likely focus on developing detection and recognition methods to effectively address complex adversarial generated text (Zhou et al., 2024). This includes enhancing the ability to detect adversarial attacks and designing more robust and reliable detection algorithms.

For intrinsic attributes. The challenges associated with black-box detection have been previously discussed. Future research should focus on enhancing model interpretation and transparency, e.g., (1) developing advanced techniques to reveal the intricate generation mechanisms and internal features of large pre-trained models. (2) Integrating explainable AI (XAI) methods, such as attention visualization and feature attribution to demystify these models' decision-making processes. (3) Establishing standardized metrics and benchmarks for assessing transparency and interpretability will enable more effective comparisons and improvements.

For information embedding. Watermark technology has been a prominent focus in AI-generated text detection, showing significant potential. Current literature indicates a gap in developing watermark detection technologies with high information capacity and robust anti-attack capabilities. Future research should explore advanced methods such as adversarial information embedding and multimodal data analysis to enhance detection effectiveness. These approaches could improve the resilience and accuracy of watermark detection in sophisticated AI-generated texts.

CRediT authorship contribution statement

Zhiwei Yang: Methodology, Review & editing, Funding acquisition, Supervision. **Zhengjie Feng:** Writing draft. **Rongxin Huo:** Validation, Resource update. **Huiru Lin:** Formal analysis, Funding acquisition, Supervision. **Hanghan Zheng:** Investigation. **Ruichi Nie:** Resources. **Hongrui Chen:** Investigation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Zhiwei Yang reports financial support and article publishing charges were provided by Fundamental Research Funds for the Central Universities. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We truly thank the reviewers and editors for their great efforts. This work is partially supported by the research project funded by the Fundamental Research Funds for the Central Universities, China (21624325, 21624338), Ministry of Education of the People's Republic of China Humanities and Social Sciences Youth Foundation (24YJC890034), and the Key Laboratory of Smart Education of Guangdong Higher Education Institute, Jinan University (2022LSYS003).

Data availability

No data was used for the research described in the article.

References

Aaronson, S., Liu, J., Liu, Q., Zhandry, M., & Zhang, R. (2021). New approaches for quantum copy-protection. In *Advances in cryptology–CRYPTO 2021: 41st annual international cryptology conference* (pp. 526–555). Springer.

Abdullah, D. M., & Abdulazeez, A. M. (2021). Machine learning applications based on svm classification a review. *Qubahan Academic Journal*, *1*(2), 81–90.

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., et al. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Almeida, T. A., Silva, T. P., Santos, I., & Hidalgo, J. M. G. (2016). Text normalization and semantic indexing to enhance instant messaging and sms spam filtering. *Knowledge-Based Systems*, *108*, 25–32.

Bao, G., Zhao, Y., Teng, Z., Yang, L., & Zhang, Y. (2024). Fast-detectgpt: efficient zero-shot detection of machine-generated text via conditional probability curvature. In *Proceedings of the international conference on learning representations* (pp. 1–23).

Benaich, N., & Hogarth, I. (2020). *State of AI report*. London.

Bhattacharjee, A., Kumarage, T., Moraffah, R., & Liu, H. (2023). Conda: contrastive domain adaptation for ai-generated text detection. In *Proceedings of the international joint conference on natural language processing and the conference of the Asia-Pacific chapter of the association for computational linguistics* (pp. 598–610).

Boroujeny, M. K., Jiang, Y., Zeng, K., & Mark, B. (2024). Multi-bit distortion-free watermarking for large language models. arXiv preprint arXiv:2402.16578.

Chakraborty, S., Bedi, A. S., Zhu, S., An, B., Manocha, D., & Huang, F. (2023). On the possibilities of ai-generated text detection. arXiv preprint arXiv:2304.04736.

Chakraborty, U., Gheewala, J., Degadwala, S., Vyas, D., & Soni, M. (2024). Safeguarding authenticity in text with bert-powered detection of ai-generated content. In *Proceedings of the international conference on inventive computation technologies* (pp. 34–37). IEEE.

Chen, Y., Kang, H., Zhai, V., Li, L., Singh, R., & Raj, B. (2023). Gpt-sentinel: distinguishing human and chatgpt generated content. arXiv preprint arXiv:2305.07969.

Chen, H., Wu, L., Chen, J., Lu, W., & Ding, J. (2022). A comparative study of automated legal text classification using random forests and deep learning. *Information Processing & Management*, *59*(2), Article 102798.

Chu, E., Andreas, J., Ansolabehere, S., & Roy, D. (2023). Language models trained on media diets can predict public opinion. arXiv preprint arXiv:2303.16779.

Ebrahimi, J., Rao, A., Lowd, D., & Dou, D. (2018). HotFlip: white-box adversarial examples for text classification. In *Proceedings of the annual meeting of the association for computational linguistics* (pp. 31–36).

Fan, A., Lewis, M., & Dauphin, Y. (2018). Hierarchical neural story generation. In I. Gurevych, & Y. Miyao (Eds.), *vol. 1*, *Proceedings of the 56th annual meeting of the association for computational linguistics*. Association for Computational Linguistics.

Fang, H., Jia, Z., Zhou, H., Ma, Z., & Zhang, W. (2022). Encoded feature enhancement in watermarking network for distortion in real scenes. *IEEE Transactions on Multimedia*.

Fernandez, P., Chaffin, A., Tit, K., Chappelier, V., & Furon, T. (2023). Three bricks to consolidate watermarks for large language models. In *IEEE international workshop on information forensics and security* (pp. 1–6). IEEE.

Gao, J., Lanchantin, J., Soffa, M. L., & Qi, Y. (2018). Black-box generation of adversarial text sequences to evade deep learning classifiers. In *IEEE security and privacy workshops* (pp. 50–56). IEEE.

Ghosal, S. S., Chakraborty, S., Geiping, J., Huang, F., Manocha, D., & Bedi, A. S. (2023). Towards possibilities & impossibilities of ai-generated text detection: a survey. arXiv preprint arXiv:2310.15264.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *Journal of Machine Learning Research*, *13*(1), 723–773.

Gunel, B., Du, J., Conneau, A., & Stoyanov, V. (2021). Supervised contrastive learning for pre-trained language model fine-tuning. In *Proceedings of the international conference on learning representations* (pp. 1–15).

Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., et al. (2023). How close is chatgpt to human experts? comparison corpus, evaluation, and detection. arXiv preprint arXiv:2301.07597.

Hanley, H. W., & Durumeric, Z. (2024). Machine-made media: monitoring the mobilization of machine-generated articles on misinformation and mainstream news websites. *vol. 18*, In *Proceedings of the international AAAI conference on web and social media* (pp. 542–556).

He, X., Shen, X., Chen, Z., Backes, M., & Zhang, Y. (2023). Mgtbench: benchmarking machine-generated text detection. arXiv preprint arXiv:2303.14822.

Hou, A. B., Zhang, J., He, T., Wang, Y., Chuang, Y.-S., Wang, H., et al. (2023). Semstamp: a semantic watermark with paraphrastic robustness for text generation. arXiv preprint arXiv:2310.03991.

Hou, A. B., Zhang, J., Wang, Y., Khashabi, D., & He, T. (2024). K-semstamp: a clustering-based semantic watermark for detection of machine-generated text. arXiv preprint arXiv:2402.11399.

Hu, X., Chen, P.-Y., & Ho, T.-Y. (2023). Radar: robust ai-text detection via adversarial learning. *Advances in Neural Information Processing Systems*, *36*, 15077–15095.

Huang, G., Zhang, Y., Li, Z., You, Y., Wang, M., & Yang, Z. (2024). Are ai-generated text detectors robust to adversarial perturbations?. arXiv preprint arXiv:2406.01179.

Jalil, Z., & Mirza, A. M. (2009). A review of digital watermarking techniques for text documents. In *Proceedings of the international conference on information and multimedia technology* (pp. 230–234). IEEE.

Jiang, Z., Guo, M., Hu, Y., & Gong, N. Z. (2024). Watermark-based detection and attribution of ai-generated content. arXiv preprint arXiv:2404.04254.

Jiang, Z., Zhang, J., & Gong, N. Z. (2023). Evading watermark based detection of ai-generated content. In *Proceedings of the ACM SIGSAC conference on computer and communications security* (pp. 1168–1181).

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., & Goldstein, T. (2023). A watermark for large language models. In *Proceedings of the international conference on machine learning* (pp. 17061–17084). PMLR.

Krishna, K., Song, Y., Karpinska, M., Wieting, J. F., & Iyyer, M. (2023). Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In *Proceedings of the conference on neural information processing systems* (pp. 1–32).

Kryściński, W., Rajani, N., Agarwal, D., Xiong, C., & Radev, D. (2022). Booksum: a collection of datasets for long-form narrative summarization. In *Findings of the association for computational linguistics: EMNLP 2022* (pp. 6536–6558).

Lai, Z., Zhang, X., & Chen, S. (2024). Adaptive ensembles of fine-tuned transformers for llm-generated text detection. arXiv preprint arXiv:2403.13335.

Liu, S., Liu, X., Wang, Y., Cheng, Z., Li, C., Zhang, Z., et al. (2024). Does\textsc $\{DetectGPT\}$ fully utilize perturbation? selective perturbation on model-based contrastive learning detector would be better. arXiv preprint arXiv:2402.00263.

Liu, X., Zhang, Z., Wang, Y., Pu, H., Lan, Y., & Shen, C. (2022). Coco: coherence-enhanced machine-generated text detection under data limitation with contrastive learning. arXiv preprint arXiv:2212.10341.

Macko, D., Moro, R., Uchendu, A., Lucas, J., Yamashita, M., Pikuliak, M., et al. (2023). Multitude: large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 9960–9987).

Mao, C., Vondrick, C., Wang, H., & Yang, J. (2024). Raidar: generative ai detection via rewriting. In *Proceedings of the international conference on learning representations* (pp. 1–18).

Markowitz, D. M., Hancock, J. T., & Bailenson, J. N. (2024). Linguistic markers of inherently false ai communication and intentionally false human communication: evidence from hotel reviews. *Journal of Language and Social Psychology*, *43*(1), 63–82.

Merity, S., Xiong, C., Bradbury, J., & Socher, R. (2017). Pointer sentinel mixture models. In *Proceedings of the international conference on learning representations* (pp. 1–15).

Mireshghallah, N., Mattern, J., Gao, S., Shokri, R., & Berg-Kirkpatrick, T. (2023). Smaller language models are better black-box machine-generated text detectors. arXiv preprint arXiv:2305.09859.

Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., & Finn, C. (2023). Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the international conference on machine learning* (pp. 24950–24962). PMLR.

Mo, Y., Qin, H., Dong, Y., Zhu, Z., & Li, Z. (2024). Large language model (llm) ai text generation detection based on transformer deep learning algorithm. *International Journal of Engineering and Management Research*, *14*(2), 154–159.

Moreno, A., & Redondo, T. (2016). Text analytics: the convergence of big data and artificial intelligence. *IJIMAI*, *3*(6), 57–64.

Narayan, S., Cohen, S. B., & Lapata, M. (2018). Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 1797–1807).

Perkins, M., Roe, J., Postma, D., McGaughran, J., & Hickerson, D. (2024). Detection of gpt-4 generated text in higher education: combining academic judgement and software to identify generative ai tool misuse. *Journal of Academic Ethics*, *22*(1), 89–113.

Pourpanah, F., Abdar, M., Luo, Y., Zhou, X., Wang, R., Lim, C. P., et al. (2022). A review of generalized zero-shot learning methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(4), 4051–4070.

Qu, W., Yin, D., He, Z., Zou, W., Tao, T., Jia, J., et al. (2024). Provably robust multi-bit watermarking for ai-generated text via error correction code. arXiv preprint arXiv:2401.16820.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., et al. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, *21*(140), 1–67.

Rosenberg, C., Hebert, M., & Schneiderman, H. (2005). Semi-supervised self-training of object detection models. In *Proceedings of the IEEE workshops on application of computer vision* (pp. 29–36).

Sadasivan, V. S., Kumar, A., Balasubramanian, S., Wang, W., & Feizi, S. (2023). Can ai-generated text be reliably detected?. arXiv preprint arXiv:2303.11156.

Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 761–769).

Sleimi, A., Sannier, N., Sabetzadeh, M., Briand, L., & Dann, J. (2018). Automated extraction of semantic legal metadata using natural language processing. In *Proceedings of the international requirements engineering conference* (pp. 124–135). IEEE.

Soto, R. A. R., Koch, K., Khan, A., Chen, B. Y., Bishop, M., & Andrews, N. (2024). Few-shot detection of machine-generated text using style representations. In *Proceedings of the international conference on learning representations*.

Su, Z., Wu, X., Zhou, W., Ma, G., & Hu, S. (2023). Hc3 plus: a semantic-invariant human chatgpt comparison corpus. arXiv preprint arXiv:2309.02731.

Su, J., Zhuo, T., Wang, D., & Nakov, P. (2023). Detectllm: leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the association for computational linguistics: EMNLP* (pp. 12395–12412).

Szczepański, M., Pawlicki, M., Kozik, R., & Choraś, M. (2021). New explainability method for bert-based model in fake news detection. *Scientific Reports*, *11*(1), 23705.

Tang, R., Chuang, Y.-N., & Hu, X. (2024). The science of detecting llm-generated text. *Communications of the ACM*, *67*(4), 50–59.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., et al. (2023). An instruction-following llama model. URL https://github.com/tatsu-lab/stanford_alpaca.

Tian, Y., Chen, H., Wang, X., Bai, Z., Zhang, Q., Li, R., et al. (2024). Multiscale positive-unlabeled detection of ai-generated texts. In *Proceedings of the international conference on learning representations*.

Topkara, M., Taskiran, C. M., & Delp III, E. J. (2005). Natural language watermarking. *vol. 5681*, In *Security, steganography, and watermarking of multimedia contents VII* (pp. 441–452). SPIE.

Topkara, U., Topkara, M., & Atallah, M. J. (2006). The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions. In *Proceedings of the workshop on multimedia and security* (pp. 164–174).

Tulchinskii, E., Kuznetsov, K., Kushnareva, L., Cherniavskii, D., Nikolenko, S., Burnaev, E., et al. (2024). Intrinsic dimension estimation for robust detection of ai-generated texts. *Advances in Neural Information Processing Systems*, *36*.

Uchendu, A., Ma, Z., Le, T., Zhang, R., & Lee, D. (2021). Turingbench: a benchmark environment for turing test in the age of neural text generation. In *Findings of the association for computational linguistics: EMNLP* (pp. 2001–2016).

Verma, V., Fleisig, E., Tomlin, N., & Klein, D. (2024). Ghostbuster: detecting text ghostwritten by large language models. In *Proceedings of the conference of the North American chapter of the association for computational linguistics* (pp. 1702–1717).

Vidanagama, D. U., Silva, A., & Karunananda, A. S. (2022). Ontology based sentiment analysis for fake review detection. *Expert Systems with Applications*, *206*, Article 117869.

Wang, R., Chen, H., Zhou, R., Ma, H., Duan, Y., Kang, Y., et al. (2024). LLM-detector: improving ai-generated chinese text detection with open-source llm instruction tuning. arXiv preprint arXiv:2402.01158.

Wang, H., Li, J., & Li, Z. (2024). Ai-generated text detection and classification based on bert deep learning algorithm. arXiv preprint arXiv:2405.16422.

Wang, P., Li, L., Ren, K., Jiang, B., Zhang, D., & Qiu, X. (2023). Seqxgpt: sentence-level ai-generated text detection. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 1144–1156).

Wang, Y., Mansurov, J., Ivanov, P., Su, J., Shelmanov, A., Tsvigun, A., et al. (2024). M4gt-bench: evaluation benchmark for black-box machine-generated text detection. arXiv preprint arXiv:2402.11175.

Wang, Y., Mansurov, J., Ivanov, P., Su, J., Shelmanov, A., Tsvigun, A., et al. (2024). M4: multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the conference of the European chapter of the association for computational linguistics* (pp. 1369–1407).

Wang, W., Zheng, V. W., Yu, H., & Miao, C. (2019). A survey of zero-shot learning: settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology*, *10*(2), 1–37.

Wu, Y., Hu, Z., Zhang, H., & Huang, H. (2024). Dipmark: a stealthy, efficient and resilient watermark for large language models. In *Proceedings of the international conference on learning representations* (pp. 1–27).

Wu, K., Pang, L., Shen, H., Cheng, X., & Chua, T.-S. (2023). Llmdet: a third party large language models generated text detection tool. In *Findings of the association for computational linguistics: EMNLP* (pp. 2113–2133).

Xu, Y., Hu, J., Gao, Z., & Chen, J. (2022). Ucl-ast: active self-training with uncertainty-aware clouded logits for few-shot text classification. In *Proceedings of the IEEE international conference on tools with artificial intelligence* (pp. 1390–1395). IEEE.

Xylogiannopoulos, K. F., Xanthopoulos, P., Karampelas, P., & Bakamitsos, G. A. (2024). ChatGPT paraphrased product reviews can confuse consumers and undermine their trust in genuine reviews. can you tell the difference? *Information Processing & Management*, *61*(6), Article 103842.

Yang, X., Chen, K., Zhang, W., Liu, C., Qi, Y., Zhang, J., et al. (2023). Watermarking text generated by black-box language models. arXiv preprint arXiv:2305.08883.

Yang, X., Cheng, W., Wu, Y., Petzold, L., Wang, W. Y., & Chen, H. (2024). Dna-gpt: divergent n-gram analysis for training-free detection of gpt-generated text. In *Proceedings of the international conference on learning representations* (pp. 1–26).

Yang, X., Zhang, J., Chen, K., Zhang, W., Ma, Z., Wang, F., et al. (2022). Tracing text provenance via context-aware lexical substitution. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 11613–11621).

Yoo, K., Ahn, W., Jang, J., & Kwak, N. (2023). Robust multi-bit natural language watermarking through invariant features. In *Proceedings of the annual meeting of the association for computational linguistics* (pp. 2092–2115).

Yu, P., Chen, J., Feng, X., & Xia, Z. (2023). Cheat: a large-scale dataset for detecting chatgpt-written abstracts. arXiv preprint arXiv:2304.12008.

Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., et al. (2019). Defending against neural fake news. In *Proceedings of the international conference on neural information processing systems* (pp. 9054–9065).

Zhang, R., Hussain, S. S., Neekhara, P., & Koushanfar, F. (2024). Remark-llm: a robust and efficient watermarking framework for generative large language models. In *USeNIX security symposium*.

Zhang, S., Liu, F., Yang, J., Yang, Y., Li, C., Han, B., et al. (2024). Detecting machine-generated texts by multi-population aware optimization for maximum mean discrepancy. arXiv preprint arXiv:2402.16041.

Zhang, Y., Ma, Y., Liu, J., Liu, X., Wang, X., & Lu, W. (2024). Detection vs. anti-detection: is text generated by ai detectable? In *Proceedings of the international conference on information* (pp. 209–222). Springer.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., et al. (2022). Opt: open pre-trained transformer language models. arXiv preprint arXiv:2205.01068.

Zhao, X., Ananth, P. V., Li, L., & Wang, Y.-X. (2024). Provable robust watermarking for ai-generated text. In *Proceedings of the international conference on learning representations* (pp. 1–35).

Zhao, X., Li, L., & Wang, Y.-X. (2024). Permute-and-flip: an optimally robust and watermarkable decoder for llms. arXiv preprint arXiv:2402.05864.

Zhou, Y., He, B., & Sun, L. (2024). Humanizing machine-generated content: evading ai-text detection through adversarial attack. In *Proceedings of the joint international conference on computational linguistics, language resources and evaluation* (pp. 8427–8437).