Humanizing Machine-Generated Content: Evading AI-Text Detection through Adversarial Attack

Ying Zhou[1,2], **Ben He**[1,2*], **Le Sun**[2,3*]
[1]University of Chinese Academy of Sciences, Beijing, China
[2]Chinese Information Processing Laboratory [3]State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China
zhouying20@mails.ucas.ac.cn, benhe@ucas.ac.cn, sunle@iscas.ac.cn

Abstract

With the development of large language models (LLMs), detecting whether text is generated by a machine becomes increasingly challenging in the face of malicious use cases like the spread of false information, protection of intellectual property, and prevention of academic plagiarism. While well-trained text detectors have demonstrated promising performance on unseen test data, recent research suggests that these detectors have vulnerabilities when dealing with adversarial attacks such as paraphrasing. In this paper, we propose a framework for a broader class of adversarial attacks, designed to perform minor perturbations in machine-generated content to evade detection. We consider two attack settings: white-box and black-box, and employ adversarial learning in dynamic scenarios to assess the potential enhancement of the current detection model's robustness against such attacks. The empirical results reveal that the current detection models can be compromised in as little as 10 seconds, leading to the misclassification of machine-generated text as human-written content. Furthermore, we explore the prospect of improving the model's robustness over iterative adversarial learning. Although some improvements in model robustness are observed, practical applications still face significant challenges. These findings shed light on the future development of AI-text detectors, emphasizing the need for more accurate and robust detection methods.

Keywords: LLM Generation, AI-text Detection, Black-box Attack, Adversarial Learning

1. Introduction

Large language models (LLMs) (OpenAI, 2023; Anil et al., 2023; Touvron et al., 2023) have rapidly emerged as a dominant force within the field of natural language processing (NLP). These models acquire extensive internal knowledge through pre-training on large-scale self-supervised data, endowing them the capacity to tackle various tasks, from answering factual questions to generating fluent text, and even performing complex reasoning, which has significantly impacted diverse NLP application domains. However, these advancements have also raised ethical concerns on their inherent risks (McKenna et al., 2023; Bian et al., 2023; Ferrara, 2023), including the spread of misinformation, the hallucinations in generated content, and even potential discrimination against specific groups. The growing recognition of these issues has led to the development of AI-text detection research. Nevertheless, AI-text detector may inherit vulnerabilities from neural network models (Szegedy et al., 2014), spurring related research (Sadasivan et al., 2023; Krishna et al., 2023) aimed at conducting paraphrasing attacks on AI detectors to mislead their predictions. We believe that the exploration of potential adversarial attacks on text detectors is of paramount importance, as it allows for the identification of vulnerabilities in AI detectors before their deployment in real-world applications, such as student essay plagiarism detection, while also facilitating the development of appropriate countermeasures.

Current detection methods are typically categorized into three groups: those relying on statistical measures (Mitchell et al., 2023) like entropy, perplexity, and log-likelihood; those training neural classifiers (Guo et al., 2023) from supervised data with human/AI-generated labels; and those utilizing watermarking (Kirchenbauer et al., 2023) to inject imperceptible pattern to the AI-generated text. Unfortunately, limited research has explored the adversarial perturbations targeting AI-text detectors. Notably, Sadasivan et al. (2023); Krishna et al. (2023) explored the use of paraphraser to rewrite machine-generated content for adversarial attacks. Simultaneously, Shi et al. (2023) utilized LLMs to generate adversarial word candidates, and created adversarial results using a search-based method. While these prior studies have revealed the vulnerabilities of AI detectors to adversarial perturbations, the influence of adversarial attacks on the detector in real-world dynamic scenarios remains largely unexplored.

In this paper, we propose a broader task: Adversarial Detection Attack on AI-Text (ADAT). The objective of ADAT is to perturb AI-generated text in a semantically preserving manner, thereby influencing the detector's predictions and enabling machine-generated text to evade detection. Figure 1 outlines the general process of ADAT, along

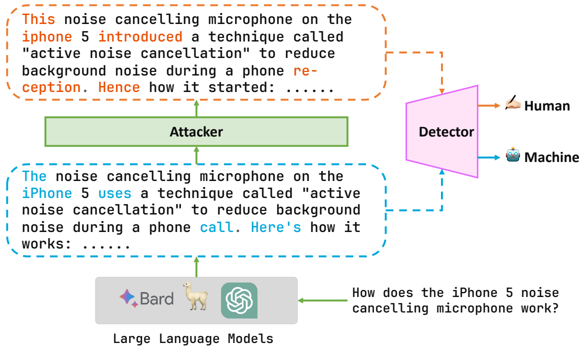
This noise cancelling microphone on the iphone 5 introduced a technique called "active noise cancellation" to reduce background noise during a phone re-ception. Hence how it started: ......
Attacker
Detector
Human
Machine
The noise cancelling microphone on the iPhone 5 uses a technique called "active noise cancellation" to reduce background noise during a phone call. Here's how it works: ......
Bard
How does the iPhone 5 noise cancelling microphone work?
Large Language Models

Figure 1: Overview of the Adversarial Detection Attack on AI-Text (ADAT) task.

with an example of attacks[1]. We consider both black-box attack setting, where the attacker can access only the target detector predictions without any internal information of the detection model, and white-box attack setting. To bridge the gap between these settings, we employ an intermediate model to unify the attack methods. Additionally, to address the challenge of constructing a more robust detection model, we introduce an adversarial learning approach in a dynamic scenario for the ADAT task, whereas the detector iteratively updates its parameters using adversarial samples. Our experiments validate the enhanced robustness achieved through this approach and highlight potential challenges. Building on the above considerations, we introduce a novel and comprehensive framework called the Humanizing Machine-Generated Content (HMGC) for ADAT tasks, which is designed to facilitate the interaction process between the attacker and the detector.

Our main contributions could be summarized as follows:

- To the best of our knowledge, ADAT is the first rigorously defined task in the field of adversarial attacks on AI-text detection. It encompasses both white-box and black-box attack settings, serving as a foundational reference for future research in this domain.
- We introduce the HMGC framework, which offers a general attack paradigm for ADAT tasks. Extensive experiments reveal the efficacy of the HMGC framework. In particular, we provide empirical evidence highlighting the significance of perplexity in AI-text detection.
- Our experimental results demonstrate that the proposed adversarial learning approach in dynamic scenarios effectively enhances the robustness of detection models, suggesting the potential to train a universal AI-text detector through dynamic adversarial learning. The data and related resources are available online[2].

2. Related Works

Large Language Model. The powerful capabilities of large language models, exemplified by GPT (OpenAI, 2023), PaLM (Anil et al., 2023), and LLaMA (Touvron et al., 2023) have revolutionized the application landscape within the field of natural language processing. These models can generate coherent and fluent text enriched with external knowledge, effectively tackling complex issues across various domains, from physics (West, 2023) and medicine (DiGiorgio and Ehrenfeld, 2023) to mathematics (Li et al., 2023) and linguistics (Liu et al., 2023b). Nonetheless, current language models still grapple with issues like hallucination (McKenna et al., 2023), the inadvertent spread of misinformation (Bian et al., 2023), and the potential for value discrimination (Ferrara, 2023) in practical applications. Consequently, the regulation of large language models to mitigate the risk of significant social problems has become increasingly vital. We focus on studying how AI-generated text can circumvent existing detection mechanisms, aiming to provide valuable insights and perspectives for the development of robust detection models.

AI-text Detection. Current text detection methods typically fall into three categories: 1) Statistical methods (Gehrmann et al., 2019; Lavergne et al., 2008; Solaiman et al., 2019; Mitchell et al., 2023; Su et al., 2023) employ statistical tools to make zero-shot distinctions between human and machine-generated text using measures such as information entropy, perplexity, and n-gram frequency. Recent work in this category includes DetectGPT (Mitchell et al., 2023), which observed that text generated by language models often resides in the negative curvature region of the log probability function, and it proposed the defining curvature-based criteria to distinguish AI-text. 2) Classifier-based methods involve training text detection models based on supervised data (Uchendu et al., 2020; Deng et al., 2023; Mireshghallah et al., 2023), whereas recent research (Guo et al., 2023; Liu et al., 2023c) often utilizes RoBERTa to train a binary classifier for text detection. However, this method requires a substantial amount of training data and faces challenges related to weak generalization (Bakhtin et al., 2019) and limited robustness against attacks (He et al., 2023; Qi et al., 2021). 3) Watermarking methods are emerging with the rise of decoder-only large language models, which imprint specific patterns on generated text. For instance, Kirchenbauer et al.

[1] Sample is from real text attacks, available for testing at https://huggingface.co/Hello-SimpleAI/chatgpt-detector-roberta

[2] https://github.com/zhouying20/HMGC

(2023) propose randomly partitioning the vocabulary into a green list and a red list during text generation, with the division based on the hash of previously generated tokens. Meanwhile, Liu et al. (2023a) introduces a watermarking method akin to the RSA asymmetric cryptography. Challenges for this category include that the generated text may lack smoothness, and the watermark pattern is susceptible to leakage (Liu et al., 2023a).

Adversarial Attack. A few recent works have made attempts to attack text detection models. For example, Sadasivan et al. (2023); Krishna et al. (2023) proposed to use paraphrasers to rewrite the generated text of LLMs, successfully evading detection by the 3 categories of models mentioned above. Notably, our work shares common views on the usage of adversarial learning with Hu et al. (2023), but differs in that Hu et al. (2023) introduced a paraphraser to enhance the robustness of detector, whereas our work explores whether the detector can continue to learn from multiple rounds of attacks in dynamic scenarios to resist adversarial attacks. Shi et al. (2023) demonstrated the utility of word substitution attacks against AI-text detectors. When compared to our approach, it's worth noting that their method relies on LLMs to generate candidate words, which might encounter efficiency challenges when computing resources are limited. Building upon these observations, our paper presents a comprehensive detector adversarial attack framework called HMGC. The key distinction between our work and these studies lies in our introduction of the ADAT task, providing a formal paradigm for future research. Moreover, we also emphasize our work as universal adversarial perturbations, which can be applied to any input for any target detector model.

3. Preliminary

In this section, we introduce the key definitions of the Adversarial Detection Attack on AI-Text (ADAT) task.

3.1. Problem Statement

In the detection of machine-generated content, when presented with a set of user requirements $\mathcal{U}$ and their corresponding response articles $\mathcal{T} = \mathcal{T}_{human} \cup \mathcal{T}_{machine}$, the objective of the detection model $\mathcal{D}$ is to assign a score $\mathcal{D}(u, t)$ to each article to help users discern whether the article is generated by machines, specifically for large language models (LLMs) like ChatGPT or Bard. For instance, given a threshold of 0.5, the detector generates detection probabilities for all articles, resulting in an array $\mathcal{P} = [D_1, D_2, ..., D_n]$, in which articles with a detection probability exceeding 0.5 are identified as machine-generated content, defined as $\overline{\mathcal{T}}_{machine} = \{t \mid D(u_t, t) > 0.5\}$.

	CheckGPT	HC3
Training size	720,000*	58,508
Testing size	90,000*	25,049
Avg #words	136.68	145.89
Domains	News, Essay, Research	QA

Table 1: Data statistics, where * denotes the data are randomly split with seed 42, and #words denotes the number of words in one sample.

Adversarial Attack on Detection. We propose the task of Adversarial Detection Attack on AI-Text (ADAT). The objective is to introduce subtle modifications to machine-generated articles, aiming to fool the detector into classifying them as human-authored. Formally defined, given user requirements u_t and a machine-generated article t, the attack's objective is to construct an effective adversarial sample t_{adv} based on t, ensuring its detection probability falls below the classification threshold. Concretely, we define a successful attack as: $D(u_t, t_{adv}) < threshold$ when $D(u_t, t) \geq threshold$, with the condition that $Dis(t, t_{adv}) \leq \epsilon$. Here, $D(u_t, t)$ and $D(u_t, t_{adv})$ are the detection probabilities of the text before and after the attack, respectively. Dis is the similarity distance evaluation function, and ϵ is a small value ensuring minimal deviation of the text distribution in the adversarial sample from the original text.

White-box and Black-box Attack. In terms of attack methodologies against the detector, we establish two realistic settings: In the **white-box attack** scenario, the attacker accesses comprehensive information about the victim detector, including parameters, gradients, training data, and more. Contrastingly, in the **black-box attack** scenario, which aligns more closely with practical applications, the attacker can only access the output results of the victim model. This means only $\mathcal{D}(u, t_{adv})$ or even the binary predictions are provided. Typically, this is further categorized into score-based black-box attacking and decision-based black-box attacking (Wu et al., 2023). In this work, we both consider *white-box attacks* and *decision-based black-box attacks*.

Dynamic Adversarial Attack Another crucial aspect of the detector is its ability to undergo continuous updates using augmentation data from users or other models. As mentioned above, Hu et al. (2023) proposed leveraging a paraphraser as a data generation method to fortify the detector against rewriting attacks during training, resulting in sufficiently robust and transferable detection results. However, this approach remains confined to the model training phase and doesn't explore the robustness and adaptability of the detector in a more dynamic scenario. In this work, we introduce the concept of **dynamic attacks** to iteratively optimize the interplay between attacker and detector across multiple rounds. Detailed processes are shown in Section 4.

3.2. Benchmarking

To validate the performance of adversarial attacks in different scenarios, we selected two datasets: 1) *CheckGPT* (Liu et al., 2023c): Due to the unavailability of the training data division or other details of its detector except for the detector model checkpoint and the full dataset, we consider this scenario as a black box attack, and we partition the entire dataset randomly into 80% training set, 10% validation set, and 10% test set. A surrogate model is trained to act as a proxy for the attacker, and subsequently, we assess the effect of the black box attack on the released original model. 2) *HC3* (Guo et al., 2023): The data partitioning and model parameters are publicly available, making it suitable for a white-box attack. We utilize the public test set as the attack samples and employ the released classifier as the victim model to evaluate the effectiveness of the white-box attack. Further details about both datasets are presented in Table 1.

4. Methodology

In this section, we start by providing an overview and mathematical definitions of our proposed attack methodologies. Following this, we outline the detailed process of establishing a unified framework that bridges the gap between black-box and white-box attack scenarios in Section 4.1 through a surrogate victim model. Furthermore, we delve into the core of our adversarial attack method in Sections 4.4, 4.2, and 4.3 by elucidating the constraints, word importance, and word replacement strategy. Finally, in Section 4.5, we introduce our innovative evaluation paradigm focused on dynamic adversarial attacks.

Algorithm Overview. The HMGC framework we introduce can be conceptualized as an ongoing interaction between the attacker and the detector. When presented with machine-generated text, the attacker iteratively modifies the text in an attempt to fool the detector. This process continues until the detector finally classifies the adversarial text as human-generated. We illustrate the general process in pseudo Algorithm 1. More specifically, the attacker in our HMGC framework comprises four key concepts: the surrogate detection model $\mathcal{D}_\theta$, the word importance ranker $\mathcal{R}$, the encoder-based word swapper $\mathcal{M}$, and a set of constraints $\mathbb{C} = \{c_1, c_2, ..., c_k\}$. The final objective of the ADAT task could be formulated as:

$$t_{adv} = \underset{t_{adv}}{\operatorname{argmin}} \mathcal{D}_\theta(t_{adv}),$$
$$s.t. \quad t_{adv} \in \{t_{tgt}\} \cup \mathcal{M}(t_{tgt}, R(w_i)),$$
$$\sum_{c_i \in \mathbb{C}} \mathbb{1}\left(c_i(t_{adv})\right) = |\mathbb{C}|. \tag{1}$$

Algorithm 1 HMGC

Input: the original detection model $\mathcal{D}_{ori}$, pre-collected training dataset $\mathbf{T}_C$, a target text t, and an encoder model $\mathcal{M}_{mlm}$, a set of attack constraints $\mathbb{C}$

Parameter: τ threshold for human-written detection, k maximum words can be replaced in one attack

Output: adversarial text t_{adv}

```
Initialize $t_{curr}$ to represent the current text to be attacked: $t_{curr} \leftarrow t$
procedure 1. Train Surrogate Model
for $t_c$ in $\mathbf{T}_C$ do
    $P_D \leftarrow$ predict all $t_c$ using $\mathcal{D}_{ori}$
end for
Train surrogate model $\mathcal{D}_\theta$ on $\mathbf{T}_C$ using $P_D$ as the target label in terms of Eq.2
procedure 2. Pre-attack Assessment
Predict whether the current sample is human-written: $p_h \leftarrow \mathcal{D}_\theta(t_{curr})$
procedure 3. Word Importance Ranking
$W_{curr} = \{w_1, w_2, ..., w_n\} \leftarrow$ Split current sample to words
Calculate word importance $I_{w_i}$ based on gradient and perplexity for each word with Eq.5
$W[:k] \leftarrow$ Sort $W_{curr}$ based on importance
procedure 4. Mask Word Substitution
for $w_i$ in $W[:k]$ do
    Replace $w_i$ in $t_{curr}$ with MASK token
    Obtain $M$ candidates $\{p_m\}_{m=1}^{M}$ with Eq.6
    $p_m^* \leftarrow argmax_{m=1}^{M}(\mathcal{D}_\theta(t_{curr} + p_m))$
    $t_{opt} \leftarrow$ Fill in $p_m^*$ for $t_{curr}$
    if $\mathcal{D}_\theta(t_{opt}) < \mathcal{D}_\theta(t_{curr})$ then
        $t_{curr} \leftarrow t_{opt}$
    end if
    Post-checking for $t_{curr}$ with Eq.7 on $\mathbb{C}$
end for
procedure 5. Post-attack Checking
if $\mathcal{D}_\theta(t_{curr}) < \tau$ or reach attack limits then
    return $t_{adv} = t_{curr}$    ▷ Algorithm terminates
else
    goto procedure 3.    ▷ Repeat the process
end if
```

4.1. Surrogate Victim Model

Under the black-box attack setting, obtaining the internal information of the detection model directly is not feasible. To compute the importance of words, it is necessary to train a surrogate model that emulates the behavior of the original detection model and can provide gradients for adversarial attacks as a proxy. Following (Guo et al., 2023), we train the surrogate model based on RoBERTa (Liu et al., 2019) with the binary classification task. The training supervision signal is distilled directly from the predictions of the original model. Formally, leveraging a pre-collected training dataset $\mathbf{T}$, we employ the original detection model $\mathcal{D}_{ori}$ to predict each sample, obtaining a set of prediction results, P_D. Subsequently, we initialize the surrogate detection model $\mathcal{D}_\theta$ using the original RoBERTa for training, whereas the training objective is as follows:

$$\mathcal{L}_\theta = -\big(p_i log(\hat{y}_i)\big) + (1 - p_i)\log(1 - \hat{y}_i)), \tag{2}$$

where $p_i \in P_D$, and $\hat{y}_i = \mathcal{D}_\theta(t_i), t_i \in \mathbf{T}$.

4.2. Word Importance Ranking

In our preliminary experiments, we observed that the detector exhibits greater sensitivity to individual words within the text, particularly those that occurred in user requirements. So perturbing important words within the text tends to be more effective in carrying out adversarial attacks. To address this, we have proposed a dual-aspect word importance ranking algorithm that combines model gradients and perplexity derived from large language models. Firstly, it is intuitive that attacks are more effective on tokens with higher gradients on the victim model, whereas higher gradients indicate greater impacts on the final result. Consequently, we consider the gradient norm value corresponding to the *i*-th token as the first aspect of word importance:

$$I^g_{w_i} = \left\| \frac{\partial \mathcal{L}_\theta}{\partial \mathbf{e}_{w_i}} \right\|_1, \tag{3}$$

where $\mathcal{L}_\theta$ is the loss of the objective function; $\mathbf{e}_{w_i}$ is the embedding vector for the *i*-th token in the surrogate model.

Furthermore, existing research (Guo et al., 2023; Liang et al., 2023) on AI-text detection has emphasized the importance of language perplexity as a key indicator for distinguishing between human and machine-generated text. Typically, machine-generated content exhibits lower perplexity. To enhance the effectiveness of our attacks, we introduce additional constraints aimed at increasing the perplexity of the adversarial results, whereas we propose the use of LLM perplexity as a measure for our word importance ranking. More specifically, for each input token, we calculate the perplexity importance as the difference in language perplexity before and after the *i*-th token is removed:

$$I^p_{w_i} = ppl(t_{\setminus w_i}) - ppl(t), \tag{4}$$

where $t_{\setminus w_i}$ represents the text after removing the *i*-th token.

Subsequently, by introducing α as the weighting factor, we obtain the final word importance score:

$$I_{w_i} = (1 - \alpha)I^g_{w_i} + \alpha I^p_{w_i}. \tag{5}$$

4.3. Mask Word Replacements

Here, for an adversarial attack, we sequentially obtain synonymous candidates for each word based on its word importance score in descending order and replace them back into the original text. For instance, for the i-th important token in the text t, we: 1) Replace the token with [MASK]. 2) Utilize the encoder-based model $\mathcal{M}_{mlm}$ to predict the logits for the masked position and perform softmax. 3) Select the top k words with the highest scores as candidates. 4) Replace the candidate words back into the original text one by one, and get the final result following a greedy search strategy. It's worth noting that the source for synonym generation can be any suitable algorithm, such as word embedding spaces (Mrksic et al., 2016) or querying WordNet[3]. However, our early experiments have shown that using an encoder-based model is the most effective way for a random replacement. In summary, we generate candidate synonyms as:

$$p^i_m = \underset{m=1,2,\ldots,k}{\mathrm{softmax}} \left(\mathcal{M}_{mlm}(t_{\setminus w_i} + [MASK]_i)\right). \tag{6}$$

4.4. Attack Quality Constraints

Following the word replacement process, it can be challenging to ensure that the semantics of the original text remain relatively unchanged. For instance, a sentence like "I like that guy" might be transformed into "I hate that guy" after perturbation, resulting in a complete reversal of sentiment. To maintain both syntactic correctness and semantic consistency, we introduce three additional constraints to control the extent of deviation: 1) POS Constraint enforces that the candidate word must align with the part of speech of the word it's replacing, e.g., adjectives cannot be used to substitute nouns. 2) Maximum Perturbed Ratio Constraint limits the proportion of replacement words in the original text within a certain threshold. 3) USE Constraint utilizes the Universal Sentence Encoder (USE) (Cer et al., 2018) as a sentence similarity scorer to measure the distance between the context window of the replacement word and the original text to address the possible semantic shift problem. If the difference is too substantial, the attack is abandoned. In formal terms, for each constraint c_i and the current adversarial text t_{adv}:

$$c_i(t_{adv}) = \begin{cases} true & \text{if } t_{adv} \text{ satisfies } c_i \\ false & \text{else} \end{cases} \tag{7}$$

[3] https://wordnet.princeton.edu/

4.5. Dynamic Detector Finetuning

As mentioned in Section 3.1, in the dynamic attack setting, following each round of attacks that gather a substantial collection of adversarial samples, we proceed to continue training the surrogate model in terms of Eq. 2. This process is designed to strengthen the detector's defense capabilities against one specific form of attack, thereby enabling us to simulate a real-world application scenario where the detector accumulates user queries and continually enhances its capabilities.

5. Experiments

In this section, we first introduce our experimental setup. Next, we compare the performance between HMGC and the baselines in both black-box and white-box attack settings. We then move to the dynamic attack setting, conducting 10 rounds of attack-then-detect iterations to assess the impact of adversarial learning on attack efficacy. Finally, we conduct ablation experiments to analyze the significance of different modules within HMGC.

5.1. Experimental Setup

5.1.1. Evaluation Metrics

Attack performance measures. In line with previous research (Mitchell et al., 2023), we employ the AUC-ROC and the confusion matrix to evaluate the attack performance: 1) **AUC-ROC** is a performance measure that assesses the area under the receiver operating characteristic curve, whereas a higher AUC-ROC score indicates better detection performance. 2) **Confusion matrix** provides a detailed breakdown of the model's performance, with 'positive' denoting human-written content. We report the following three metrics: Positive predictive value (PPV) $\frac{TP}{TP+FP}$, i.e., the proportion of human-written cases among all predicted cases classified as human-written. True negative rate (TNR) $\frac{TN}{TN+FP}$, i.e., the accuracy in classifying machine-generated text. We also denote the decrease of TNR as ΔAcc, which quantifies the reduction in the accuracy of machine-generated sample detection after the attack. It is calculated as $\frac{\text{TNR before Attack} - \text{TNR after Attack}}{\text{TNR before Attack}}$.

Text quality measures. Here, we use the following metrics to evaluate the text quality after the adversarial attack. **Flesch reading ease**: Higher Flesch scores indicate that the material is easier to read. To assess the impact of text's readability, we use the difference ratio of the Flesch score, denoted as ΔFlesch$\%$. **Perplexity from LLMs**: It measures the level of uncertainty of a given document. We calculate the change in perplexity before and after the attack with Pythia-3B (Biderman et al., 2023) to measure the overall quality of an adversarial text, denoted as Δppl.

5.1.2. Baselines

Referring to recent research (Sadasivan et al., 2023; Krishna et al., 2023; Shi et al., 2023), we investigate three primary categories of baseline algorithms for the ADAT task: **Word-level perturbation** treat tokens in the input as the smallest attack units. It disrupts the detection model by substituting specific words in the original text, typically with words that have similar meanings. We consider *WordNet* and *BERT MLM predictions* as the sources for synonyms. **Sentence-level perturbation** commonly employs a seq-to-seq model to rephrase or rewrite sentences from the original text, thereby perturbing the distribution of the original content. In our study, we examined two strategies serving as baselines: introducing *irrelevant sentences* and utilizing *BART* to replace random sentences from the original text. **Full-text rewriting perturbation** involves using a rewriter to directly substitute the original text, effectively evading detection. We considered three methods: *back translation* which translates the original English text into German and then translates it back to English, and crafting the prompt to instruct a *LLaMA-2* to rewrite the article, aiming to maximize the divergence from the original text. Moreover, we also employ the SoTA paraphraser *DIPPER* (Krishna et al., 2023) with the lex=40, order=40, the most effective setting in their paper.

5.1.3. Model Ablations

As discussed in Section 4, four variations of the HMGC model are implemented by modifying the constraint module and the word importance calculation method: 1) HMGC.$_{-POS}$ does not enforce the replacement and original words to belong to the same part of speech. 2) HMGC.$_{-USE}$ eliminates the constraints related to semantic space consistency using USE (Cer et al., 2018). 3) HMGC.$_{-MPR}$, where there is no restriction on the proportion of words that can be replaced. 4) HMGC.$_{-PPL}$ removes the constraint of semantic perplexity in the word importance method, relying solely on the gradient information of the victim model.

5.1.4. Implementation Details

For the black-box attack using CheckGPT datasets, we adopt RoBERTa as the surrogate model, where we distilled the original detection performance over two epochs on the 720k training data. In detail, the maximum sequence length is set to 512, and the learning rate is set to 5e-6. As for both white-box and black-box attacks, we select 10k samples

Model	White-box Attack on HC3				Black-box Attack on CheckGPT				Duration
	AUC ↓	PPV ↓	TNR ↓	ΔAcc ↑	AUC ↓	PPV ↓	TNR ↓	ΔAcc ↑	Sec/Sample ↓
WordNet Syn	98.36	98.73	97.30	2.55	91.39	85.56	83.07	16.81	≈ 0
MLM Syn	97.79	98.19	96.15	3.70	87.68	80.46	75.65	24.24	0.1
Irr Sent	98.66	99.00	97.89	1.96	95.88	92.66	92.06	7.81	≈ 0
MLM Sent	95.80	96.40	92.17	7.69	94.26	89.97	88.83	11.05	5.27
Back Trans	99.20	99.51	98.97	0.87	94.89	90.99	90.07	9.80	3.26
LLaMA-2-7B	95.94	96.52	92.45	7.41	96.13	93.09	92.56	7.31	9.51
LLaMA-2-13B	97.97	98.37	96.52	3.33	96.23	93.26	92.76	7.11	10.61
DIPPER	98.62	98.78	97.82	2.02	88.77	81.90	77.84	22.05	14.43
HMGC	**51.06**	**68.29**	**2.70**	**97.29**	**76.64**	**68.35**	**53.57**	**46.35**	9.25

Table 2: Attack performance of white-box and black-box setting on HC3 and CheckGPT.

Round	Automatic Metrics				Duration
	AUC	PPV	TNR	ΔAcc	Sec
1	49.05	49.18	0.44	99.56	9.25
2	46.64	49.06	1.69	98.30	15.11
3	48.47	45.85	4.79	95.21	20.99
4	62.87	56.61	32.61	67.35	26.40
5	75.89	68.61	58.91	40.83	30.40
6	83.16	80.62	78.27	21.29	32.30
7	88.58	87.12	87.09	12.53	33.10
8	90.00	90.17	90.23	9.47	33.40
9	90.80	87.30	87.25	12.19	33.17
10	87.25	85.96	85.06	14.85	32.00

Table 3: Attack performance in a dynamic environment on CheckGPT. The first-round model represents the original surrogate model, evaluated on test data generated by its adversarial attacks. The second-round model signifies the model that has undergone an adversarial learning process and is evaluated on the test data generated by its adversarial attack. This process continues iteratively for subsequent rounds.

from their test set for attacking. In the dynamic attack, we randomly divide the 90k test data into 10 equal parts. After each attack, 80% of the attack results are incorporated as new training data, while 20% of the attack results (equivalent to 1.8k) are utilized for evaluating the model. For the attacker, the perplexity weighting factor α is set at 0.2, the window size for the USE in the fluency constraint is 50, the minimum tolerance threshold γ is 0.75, and the maximum proportion of replaceable words does not exceed 40% of the original text. All experiments were conducted on a machine equipped with six 3090 GPUs.

5.2. Experimental Results

Detectors are vulnerable to adversarial attack. Table 2 shows the results of both the baseline models and our proposed HMGC in white-box and black-box attack settings for the two datasets. Notably, white-box attacks naturally provide precise insights into the internal information of the detection model, making them less robust against disturbance. The AUC, for instance, demonstrates a significant drop in detection performance, plummeting from 99.63% pre-attack to a mere 51.06%, which is akin to that of a random binary classifier. It should be noted that the HC3 dataset comprises more human-generated articles (67.82%), leading to a PPV much higher than 50%. Meanwhile, for a more intuitive measure ΔAcc, after the attack, the model's misclassification rate for machine-generated content surged by 97.29%. Furthermore, in the more challenging black-box attack setting, our proposed HMGC demonstrated considerable effectiveness. It successfully perturbed approximately 46% of machine-generated articles, marking a 22% improvement compared to the optimal baseline model. In general, regardless of whether in white-box or black-box attack settings, our proposed HMGC method consistently outperforms all the baseline models.

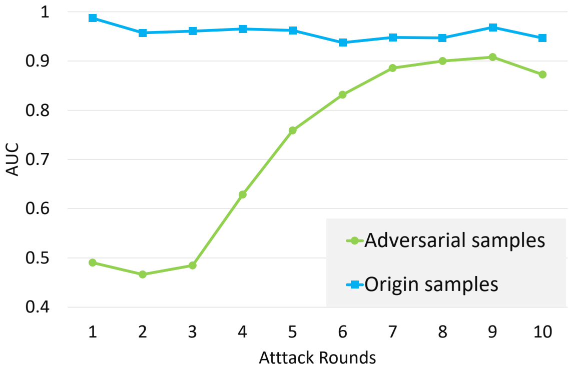

Figure 2: Detection performance across different attack rounds. As the attack rounds intensify, the detector builds resilience against this form of attack.

Training method for the detector significantly influences its robustness. Analyzing the baseline performance from Table 2, it is evident that models produced through different training methods exhibit varying degrees of robustness against general perturbations. The original CheckGPT model,

Model	White-box Attack on HC3				Black-box Attack on CheckGPT				Text Quality	
	AUC ↓	PPV ↓	TNR ↓	ΔAcc ↑	AUC ↓	PPV ↓	TNR ↓	ΔAcc ↑	Δflesch%	Δppl
HMGC	51.06	68.29	2.70	97.29	76.64	68.35	53.57	46.35	-7.74	6.17
HMGC.$_{-USE}$	49.96	67.80	0.50	99.50	75.99	67.76	52.29	47.64	-7.76	6.36
HMGC.$_{-MPR}$	50.82	68.18	2.21	97.79	76.65	68.36	53.59	46.33	-7.75	6.17
HMGC.$_{-POS}$	50.04	67.84	0.65	99.35	77.95	69.59	56.20	43.72	-6.36	5.86
HMGC.$_{-PPL}$	50.46	68.02	1.49	98.51	80.64	72.29	61.57	38.34	-6.08	5.66

Table 4: Attack performance and text quality comparisons between HMGC and its ablations.

which only trained the top-level LSTM with frozen RoBERTa, showed a limited capacity to withstand minor perturbations. Substituting some words with BERT yielded a substantial improvement, increasing ΔAcc by 24%. On the other hand, the HC3 with full parameter fine-tuning appeared more robust. Notably, considering the relative trends, it becomes apparent that using a language model to generate candidate words and sentences demonstrates greater adversarial performance compared to heuristic replacement methods. Moreover, employing a prompt to guide LLMs in rewriting the text proves to be more versatile. Even without prompt engineering, it can still achieve notable adversarial effects on all detection models.

Adversarial learning can effectively enhance detector performance. As depicted in Table 3, the increase in the number of rounds of dynamic adversarial learning, which represents the continued training for the detector with adversarial samples, positively correlates with the detector's robustness. Meanwhile, this enhancement comes at the cost of increased time required to perform an adversarial attack. Specifically, in the initial 3 rounds of attacks, the detector remains relatively vulnerable, with its AUC index dropping to approximately 0.5 whereas the time required for each attack approximately doubles. Subsequently, equilibrium is reached after roughly 7 rounds of attacks, corresponding to iterative learning from 50,000 adversarial examples. At this point, the attacker can produce an attack every 30 seconds, albeit with only about a 10% success rate. This trend is visually illustrated in Figure 2. Despite the improved robustness, the detector remains impractical for real-world applications, as it still yields an error rate of over 10% when classifying adversarial content as human-written. In summary, these experimental results demonstrate the positive impact of iterative adversarial learning in dynamic scenarios on enhancing detector robustness. Since the current detector does not incorporate adversarial attack considerations in its model design, our research contributes valuable insights for the development of future detection models.

Trade-off: evasion of detection or preservation of original semantics. We conducted an ablation analysis to examine the key modules in the HMGC model design. From the results in Table 4, we observe the following: 1) In the white-box attack setting, which is relatively straightforward, the ablation analysis of the model causes only minor fluctuations in attack accuracy, typically within 1 or 2 percentage points. These variations may stem from the random factors involved in the attack experiments. 2) The black-box attack setting effectively demonstrates the significance of the perplexity word importance we proposed. When the module is removed as HMGC.$_{-PPL}$, the attack success rate decreases by 8%, indicating the effectiveness of word perplexity in the ADAT task. 3) Ablation results for other modules show that the attack success rate is directly proportional to the language perplexity of the adversarial text. For instance, when the USE constraint is removed as HMGC.$_{-USE}$, the attack success rate increases by approximately 1%, but the corresponding language perplexity also rises by 0.2. From these observations, we can deduce that an effective strategy to evade AI-text detection is introducing external noise to increase text perplexity. However, this approach may face the challenge of semantic shifts between the original text and the adversarial text. Balancing these factors should be a crucial consideration in future research on AI-text detection.

6. Conclusion

In this paper, we introduce the Adversarial Detection Attack on AI-Text (ADAT) task, which includes two attack settings: white-box and black-box attacks. Furthermore, we propose a novel approach involving adversarial learning in dynamic scenarios to enhance the resistance of detection models against adversarial attacks. Our algorithm demonstrations and experimental results prove the vulnerability of the current design in detection models, revealing their susceptibility to even minor perturbations that can effectively disrupt the final prediction results. To perform effective adversarial attacks, we present the Humanizing Machine-Generated Content (HMGC) framework, which emulates the interactive attack process between an attacker and a detector, continuously refining the attack strategy based on the rewards provided by the detector. Our proposed approach, supported by extensive experimental results, not only highlights the vulnerabilities in existing AI-text detection methods but also sheds light on the risks and directions for future research in the AI-detection domain.

In future work, we plan to expand our framework to support sentence-level and document-level substitutions to produce more fluent adversarial texts. Concurrently, we will refine the adversarial learning approach in dynamic scenarios to train a more robust, stable, and versatile AI-text detector. Moreover, beyond the ADAT tasks, exploring more general content correction technology for the AI-text also appears to be a promising direction for further research.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (62272439), and the Fundamental Research Funds for the Central Universities.

Bibliographical References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, and et al. 2023. Palm 2 technical report. *CoRR*, abs/2305.10403.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *CoRR*, abs/1906.03351.

Ning Bian, Peilin Liu, Xianpei Han, Hongyu Lin, Yaojie Lu, Ben He, and Le Sun. 2023. A drop of ink makes a million think: The spread of false information in large language models. *CoRR*, abs/2305.04812.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 169–174. Association for Computational Linguistics.

Zhijie Deng, Hongcheng Gao, Yibo Miao, and Hao Zhang. 2023. Efficient detection of llm-generated texts with a bayesian surrogate model. *CoRR*, abs/2305.16617.

Anthony M. DiGiorgio and Jesse M. Ehrenfeld. 2023. Artificial intelligence in medicine & chatgpt: De-tether the physician. *J. Medical Syst.*, 47(1):32.

Emilio Ferrara. 2023. Should chatgpt be biased? challenges and risks of bias in large language models. *CoRR*, abs/2304.03738.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: statistical detection and visualization of generated text. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pages 111–116. Association for Computational Linguistics.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *CoRR*, abs/2301.07597.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. Mgtbench: Benchmarking machine-generated text detection. *CoRR*, abs/2303.14822.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: robust ai-text detection via adversarial learning. *CoRR*, abs/2307.03838.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *CoRR*, abs/2303.13408.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. In *Proceedings of the ECAI'08 Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse, Patras, Greece, July 22, 2008*, volume 377 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Pin-Hui Li, Hsin-Yu Lee, Yu-Ping Cheng, Andreja Istenic Starcic, and Yueh-Min Huang. 2023. Solving the self-regulated learning problem: Exploring the performance of chatgpt in mathematics. In *Innovative Technologies and Learning - 6th International Conference, ICITL 2023, Porto, Portugal, August 28-30, 2023, Proceedings*, volume 14099 of *Lecture Notes in Computer Science*, pages 77–86. Springer.

Weixin Liang, Mert Yüksekgönül, Yining Mao, Eric Wu, and James Zou. 2023. GPT detectors are biased against non-native english writers. *Patterns*, 4(7):100779.

Aiwei Liu, Leyi Pan, Xuming Hu, Shuang Li, Lijie Wen, Irwin King, and Philip S. Yu. 2023a. A private watermark for large language models. *CoRR*, abs/2307.16230.

Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, and et al. 2023b. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. *CoRR*, abs/2304.01852.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. 2023c. Check me if you can: Detecting chatgpt-generated academic writing using checkgpt. *CoRR*, abs/2306.05524.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Javad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. *CoRR*, abs/2305.14552.

Fatemehsadat Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. 2023. Smaller language models are better black-box machine-generated text detectors. *CoRR*, abs/2305.09859.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

Nikola Mrksic, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gasic, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve J. Young. 2016. Counter-fitting word vectors to linguistic constraints. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 142–148. The Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 443–453. Association for Computational Linguistics.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *CoRR*, abs/2303.11156.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2023. Red teaming language model detectors with language models. *CoRR*, abs/2305.19713.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *CoRR*, abs/2306.05540.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, and et al. 2023. Llama

2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8384–8395. Association for Computational Linguistics.

Colin G. West. 2023. AI and the FCI: can chatgpt project an understanding of introductory physics? *CoRR*, abs/2303.01067.

Chen Wu, Ruqing Zhang, Jiafeng Guo, Maarten de Rijke, Yixing Fan, and Xueqi Cheng. 2023. PRADA: practical black-box adversarial attacks against neural ranking models. *ACM Trans. Inf. Syst.*, 41(4):89:1–89:27.