Coarse-to-Fine AI Text Detection: Hierarchical Contrastive Learning in Dual Stages

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
`email@domain`

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
`email@domain`

Abstract

As the ability of Large Language Models (LLM) to imitate human writing becomes stronger and the diversity of machine texts disguises continuously increases, current AI text detectors have shown vulnerabilities. Among them, a considerable number of detectors are single-stage, relying heavily on the comparison results between final text score and the threshold value for judgment, which has significant vulnerabilities when dealing with strategically disguised machine texts. Given that the current disguise methods can be roughly divided into four levels: character, word, sentence, and paragraph, we targetedly propose a two-stage detector based on hierarchical contrastive learning. The "human" texts filtered out in the first-stage contains real human texts and disguised machine texts, and these samples will be fine-grainedly discriminated through the second-stage detector which applies a hierarchical unsupervised contrastive learning strategy. This multi-stage strategy makes up for the loophole that disguised machine texts can successfully escape the traditional detector after a single detection, and shows excellent robustness in the task of detecting disguised machine texts. In addition, the second-stage detector can be deployed separately on the existing detector as a backup when facing large-scale disguised machine texts attack. Experiments show that our two-stage detector has achieved advanced results, moreover, this detachable and composable approach shows strong flexibility and generalization. We hope that this multi-stage training strategy can inspire new sparks in the field of AI-generated text detection.Our code and dataset will soon be available.

1 Introduction

The explosive rise of LLMs[Claude AI, 2024, DeepSeek-AI, 2025, Gemini, 2024] makes it easy for people to get machine-generated texts. However, just as a coin has two sides, new problems emerge when the text generation function of LLMs facilitates people's work and life. Researchers have demonstrated various malicious applications of LLMs, including academic fraud[Perkins, 2023], spam generation, and false information dissemination[Hazell, 2023, Weidinger et al., 2022]. In order to prevent machine-generated texts from leaning into the wrong direction, machine-generated text detectors come into being to correct the development of LLMs. Existing detectors include those based on statistics and mathematics[Mitchell et al., 2023, Tian and Cui, 2023], watermarks[Gu et al., 2022, Kirchenbauer et al., 2023], classifiers[Guo et al., 2023, Wang et al., 2023], to name a few. Among them, a considerable number of detectors judge whether the text is generated by the machine based on the comparison of the final text score with a certain threshold. For example, when the test score of a text is greater than 0.5, it is considered to be written by a machine, otherwise it is considered to be written by a human. However, as large quantities of machine texts become easier to obtain and the disguise methods become more and more diverse[Zhou et al., 2024, Huang et al., 2024], it is

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

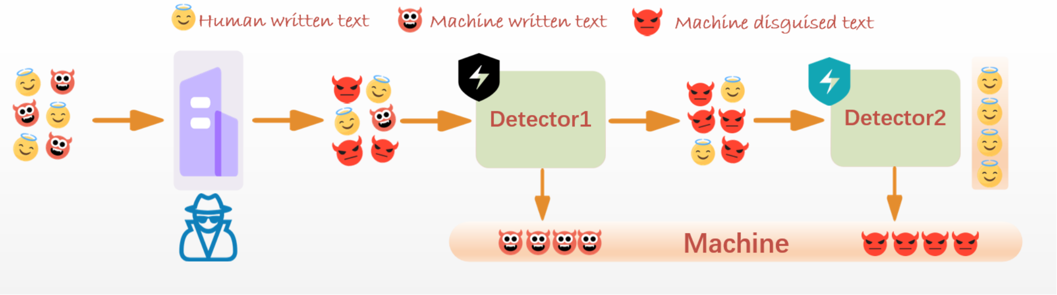

Figure 1: Overview of framework. (a) Machine text disguise. Machine-generated texts may be disguised by attackers to evade detection. We put the original machine texts into the factory for processing and get the disguised machine texts. (b) First-stage detect. The texts are filtered through the first-stage detector, and most of the original machine texts will be filtered out. (c) Second-stage detect. The filtered texts moves to the second-stage detector to get fine-grained recognition.

not enough for detectors to depend solely on the comparison of the final texts score with a certain threshold, and further improvement is needed.

There are many ways of machine disguise methods at present, such as simulating human spelling errors, word replacement, sentence back translation and so on, which can be roughly divided into four levels: character, word, sentence, and paragraph[Zhou et al., 2024]. The vulnerability of detectors in the face of these attacks has also been pointed out by many researchers[Dugan et al., 2024, Krishna et al., 2024]. Therefore, we believe that it is necessary to conduct a two-stage detection of texts, to detailedly speak, when the scores of texts are within the machine category range, the output is normally machine-generated, and when the texts' scores are within the human category range, the texts may not only be human written, but also disguised machine texts, and a second-stage detection is required. In other words, we add an extra gate to the detector, so that disguised machine texts not only need to pass the first-stage detection, but also have to undergo an extra fine-grained inspection by second-stage detector, which greatly reduces the possibility of machine texts evading detection, thereby improving the robustness of detector in the face of attacks.
Specifically, we propose a two-stage detector. For the first-stage detector, it is trained based on a general classifier framework, while for the second-stage detector, its training data comes from samples judged by the first-stage detector as human writing (including human writing samples and disguised machine samples). In order to further distinguish human samples and machine samples disguised at different levels, we prescribe the right medicine by using hierarchical contrastive learning to learn the similarities and differences between them. Concretely, for machine texts disguised at the same level, their distances in the sample space should be close to each other and far from machine samples disguised at other levels; for machine texts disguised at different levels along with original machine texts, their distances in the sample space should be close to each other and far from human texts; for human-written texts, their distances in the sample space should also be close to each other and far from machine samples. The final result can be obtained by integrating the results of the first-stage detection and the second-stage detection, which shows the vulnerability of the first-stage detector (as a representative of the current common detectors) in the face of diversified attacks, the necessity and effectiveness of the second-stage detection, and the excellent performance of our detector that ultimately exceeds the baseline.

We not only propose a joint detector with good performance, but also provide a new training paradigm for subsequent AI text detectors. The two-stage detector breaks through the previous "one" constraint of AI text detectors, allowing them to be disassembled and deployed separately: if the requirements are not high, the first-stage detector is sufficient; and for existing detectors, the second-stage detector can serve as a backup in the face of large-scale machine disguised texts detection, providing them with a possible patch.

In short, our work is multifaceted and can be summarized as follows:
(1) We use hierarchical contrastive learning to achieve fine-grained distinction between human texts, original machine texts and machine texts disguised at different levels, providing ideas for detector

designers to resist the current diverse attacks.
(2) We proposed a joint framework for training detectors, and the two-stage detector trained by the framework has achieved excellent robustness in detecting machine texts.
(3) The advantage of the two-stage detector is that it can be deployed separately according to needs, and the second-stage detector can be used as a patch to improve the robustness of existing detectors.

2 Related Works

2.1 Machine-generated text detectors.

In order to prevent machine-generated texts from being abused, numbers of detectors have been proposed by researchers, thus consolidating the defense line of text detection. We classify the existing detectors into the following four categories:
Statistical and mathematical based detectors: Using information entropy, cross perplexity, word frequency statistics and other features to perform zero-shot detection.Mitchell et al. [2023] quantify the difference between machines and humans in word selection via conditional probability curvature.Su et al. [2023] apply log-rank information to detect. Open source detecting platform Tian and Cui [2023], Gehrmann et al. [2019] are also included.
Watermark based detectors: Watermark detection algorithms in machine texts detection track the source of generated texts by embedding invisible identifiers. Representative works include:Gu et al. [2022], Liu et al. [2024a], Hou et al. [2024], Lu et al. [2024].Among them,Kirchenbauer et al. [2023] add a fixed weight to the logit value of the predefined "green word list" and determine whether the text is generated by the model by counting the proportion of green words in the texts.
Classifier based detectors:Chen et al. [2023], Miao et al. [2024], Mireshghallah et al. [2024], Wang et al. [2023], Liu et al. [2024c] typically employ RoBERTa[Liu et al., 2019] as the backbone architecture for training supervised binary classifiers. Notable developments are seen in OpenAI's official detection toolkit[Solaiman et al., 2019] and RADAR[Hu et al., 2023], which enhances adversarial robustness against perturbation attacks through paraphrase-based adversarial training.
Other methods based detecors:Soto et al. [2024b] use style representations,Huang et al. [2024] take advantage of siamese neural network,Krishna et al. [2024] achieve success through retrieval methods.Zhu et al. [2023] innovatively query LLM to detect LLM-generated texts.

2.2 Disguise methods of machine-generated texts

Many researchers have pointed out the vulnerability of current detectors, indicating even a small perturbation attack can cause the performance of the detector to drop sharply[Zhou et al., 2024, Dugan et al., 2024, Krishna et al., 2024, Liu et al., 2024b, Huang et al., 2024, Wang et al., 2024]. Specifically, Liu et al. [2024b] point out that DetectGPT relies on the threshold setting of the logit regression module (which coincides with our motivation), which is sensitive to the detection results, and the perturbations of deletion, duplication, insertion, and replacement imposed on the test data cause the performance of AI text detector to drop significantly; Dugan et al. [2024] design a variety of perturbations such as local vocabulary replacement, syntactic structure adjustment, semantic preservation rewriting, finding that with only 5% of the text content being modified, the detection accuracy drops by an average of 37.2%; Zhou et al. [2024] use twelve attacks from four levels of character, word, sentence, and paragraph on a variety of detectors, pointing out that current detectors need to be trained with adversarial texts. Their works show that lacerating the mask of disguised texts is an urgent affair that current detectors need to solve.

2.3 Contrastive Learing in Detectors

There has been work showing that contrastive learning has excellent performance in the field of natural language processing[Cheng et al., 2023]. MixCSE[Zhang et al., 2022b], SimCSE[Gao et al., 2021], VaSCL[Zhang et al., 2022a] use unsupervised contrastive learning framework to enhance the semantic discrimination ability of the model; CoCo[Liu et al., 2023] use supervised contrastive learning to make the model pay more attention to difficult negative samples in low-resource scenarios; Soto et al. [2024a], Guo et al. [2024] use contrastive learning to distinguish the style features of human and machine writing. By narrowing the distance between positive samples and increasing the distance between negative samples, contrastive learning has shown great potential in training AI content detectors.

3 Model and Methodology

In this section, we will introduce our proposed method. Section 3.1 will describe machine texts disguise methods and the composition of the detection framework, Section 3.2 is an illustration of the first-stage detector, and Section 3.3 is a detailed description of the two-stage detector design and an extended discussion of our motivation for using a two-stage detector along with the "blending the strengths of both" core training ideas.

3.1 Framework Overview

Zhou et al. [2024], Huang et al. [2024]'s work extensively explore the vulnerability of detectors to different attacks. Based on their work, we classify the existing common attack methods into the following four types,which consistent with Zhou et al. [2024]'s summary of present attacks:

1. **char level**:Attacks at this level include space deletion, space addition[Cai and Cui, 2023], capitalization typo simulation, punctuation deletion, and random word merging.
2. **word level**:Attacks at this level include keyboard spelling errors, which replace characters in similar keyboard positions; swaping adjacent characters, inserting irrelevant characters, and deleting specific characters, thereby simulating human negligence when typing; Word spelling errors, which simulate users' incorrect spelling of words through a predefined spelling error dictionary; Adverb perturbations, which randomly insert relevant adverbs before verbs in the original text; Word replacement, which uses the BERT model[Devlin et al., 2019] to replace words in the text with synonyms.
3. **sentence level**:Attacks at this level include adding irrelevant sentences; repeating parts of sentences; randomly selecting sentences for back-translation; and sentence-level replacement, which randomly masks 2 to 5 sentences in the original text and replaces them using the BART-large model[Lewis et al., 2020].
4. **paragraph level**:Attacks at this level include rewriting using the Dipper interpreter[Krishna et al., 2024]; back-translation using the Helsinki-NLP model[Tiedemann and Thottingal, 2020]; and rearrangement of paragraph structure.

Given a set of undisguised machine original texts $X = \{x_1, x_2, ..., x_k\}$ and human writing texts set $Y = \{y_1, y_2, ..., y_k\}$, an attacker with ulterior motives may disguise the original texts of machine at character, word, sentence, or paragraph level to try to evade detection, thus there will be sets of disguised machine texts:

$$\begin{aligned}
&X_{char} = \{x_{char1}, x_{char2}, \ldots, x_{chark}\} \\
&X_{word} = \{x_{word1}, x_{word2}, \ldots, x_{wordk}\} \\
&X_{sentence} = \{x_{sent1}, x_{sent2}, \ldots, x_{sentk}\} \\
&X_{paragraph} = \{x_{para1}, x_{para2}, \ldots, x_{parak}\}
\end{aligned}$$

Given a sample space S in real world detection,we have $S = X_{char} + X_{word} + X_{sentence} + X_{paragraph} + X + Y$. For a sample $s \in S$, our goal is to ultimately determine whether it is machine generated. Sample s will first pass through the first-stage detector, the final text's score is compared with the threshold and if it is in the machine range, then the conclusion that s is machine generated is directly output, which indicates that some machine samples are successfully identified by the first-stage detector,which is $s \in X$. If the text is judged to be human written, the text may be real human text or machine written text in disguise, which is $s \in X + Y + X_{char} + X_{word} + X_{sentence} + X_{paragraph}$. Sample s which is judged as human by the first stage will enter the second detector for detection.In the second detector, s will undergo multi-level contrastive learning for fine-grained differentiation, ultimately determining whether s is indeed a disguised machine text.

3.2 First Stage Detector

The first stage detector is a traditional classifier-based detector. Its purpose is to perform coarse-grained screening of input batch text. The first-stage detector can also be used as a representative of the current detector that has not been specially trained with text perturbations. A good first-stage

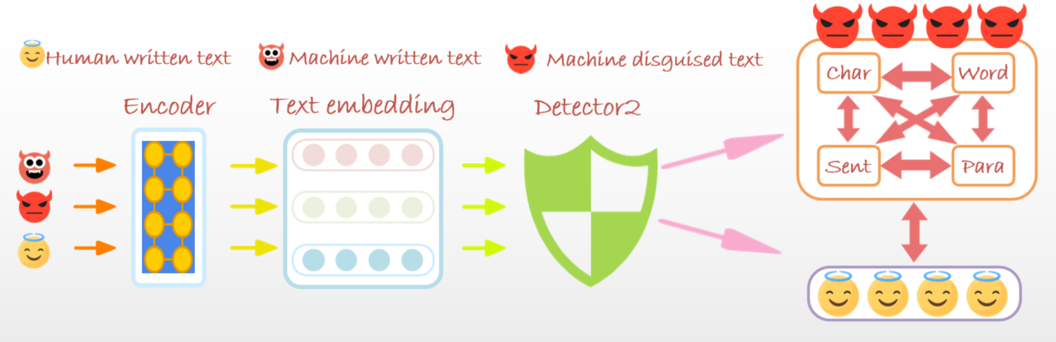

Figure 2: Overview of the second stage detector. After the text is encoded by the encoder, it's deeper features are learned through hierarchical contrastive learning strategy, which pulls in the same positive samples and pushes away the negative samples to distinguish human texts and machine texts that have undergone four levels of disguise.

detector should be able to filter out most of the original machine text that has not been disguised. Its loss function is as follows,in which l represents true label,p represents predicted label:

$$\mathcal{L}_{ce} = -\frac{1}{N}\sum_{i=1}^{N} l_i \cdot log(p_i) + (1-l_i) \cdot log(1-p_i), \tag{1}$$

For input $s \in S$, the text encoding $\Phi(s)$, the output is:

$$Prediction(s) = \begin{cases} human, score(\Phi(s)) > threshold \\ machine, scores(\Phi(s)) < threshold \end{cases} \tag{2}$$

3.3 Second Stage Detector

For the sample set S' entering the second detector, it may be human texts, the original machine texts previously missed by the first-stage detector, or the disguised machine texts that deceived the first-stage detector. Machine texts may use different levels of disguise, accordingly there are three levels of hierarchical contrastive learning in the second-stage detector: machine texts using the same disguise method and machine texts with different disguise methods, machine texts using the same level of disguise method and machine texts with different levels of disguise methods, human texts and machine texts. Given a text label triple (p, q, l), where p represents the disguise method used, q represents the level of the disguise method, and l represents the source of the sample (human or machine), we have the cosine similarity constraints:

$$\begin{cases} Sim(\Phi(s_1), \Phi(s_2)) < Sim(\Phi(s_1), \Phi(s_3)), p(s_1) = p(s_2), p(s_1) \neq p(s_3) \\ Sim(\Phi(s_4), \Phi(s_5)) < Sim(\Phi(s_4), \Phi(s_6)), q(s_4) = q(s_5), q(s_4) \neq q(s_6) \\ Sim(\Phi(s_7), \Phi(s_8)) < Sim(\Phi(s_7), \Phi(s_9)), l(s_7) = l(s_8), l(s_7) \neq l(s_9) \end{cases} \tag{3}$$

where $s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9 \in S'$.
For contrastive learning at a specific level, we use the contrastive loss based on the SimCLR framework[Chen et al., 2020], which takes the form of a negative logarithmic aggregation function, we have the loss expression eq.4, in which s represents targeted sample, S_{K^+} is a set of positive samples, S_{K^-} is a set of negative samples, τ is the temperature coefficient.

$$\mathcal{L}_p = -\log \frac{\exp\left(\sum_{k\in K^+} \frac{s(p,k)}{\tau} / S_{K^+}\right)}{\exp\left(\sum_{k\in K^+} \frac{s(p,k)}{\tau} / S_{K^+}\right) + \sum_{k\in K^-} \exp\left(\frac{s(p,k)}{\tau}\right)}. \tag{4}$$

We take contrastive loss in label p as example,label q and l's loss are consistent with the above equation.

The final contrastive learning loss should be the sum of the contrastive losses at different levels above, so we have:

$$\mathcal{L}_{contrastive-tot} = \sum_{i=1}^{K} l_i \cdot \mathcal{L}_l + (1 - l_i) \cdot (\mathcal{L}_p + \mathcal{L}_q). \tag{5}$$

where l_i represents the label l that whether the sample belongs to human or machine, K indicates the sum of all samples entering the second stage detector, and $\mathcal{L}_l$, $\mathcal{L}_p$, and $\mathcal{L}_q$ represents the loss of the second stage detector at above level.

Through hierarchical contrastive loss function propagation, the model can distinguish the differences between machine texts disguised at different levels and the similarities between machine texts disguised at the same level in a fine-grained manner. In order to determine whether the final text is machine-generated, we also introduce the cross entropy loss function eq.1 to drive the model to improve performance in the final binary classification task,we have the final loss as:

$$\mathcal{L}_{final-loss} = \mathcal{L}_{contrastive-tot} + \mathcal{L}_{ce}. \tag{6}$$

Blending the strengths of both——our two stage training idea: For the task of detecting human texts and original machine texts, the classification task is simple and direct. The classifier based on cross entropy loss has the advantages of stable optimization process, rapid convergence, and applicability to mutually exclusive scenarios[Dickson et al., 2022, Ma et al., 2022, Wood et al., 2022]. Therefore, we choose it as the main body in the first-stage detection, hoping to take advantage of its strengths and quickly filter out the original machine texts without disguise. For the more realistic task of detecting human texts and disguised machine texts, due to the prevalence of LLMs and the various ways of machine text disguise, cross entropy lacks sensitivity to intra-class differences[Liu et al., 2016, Sun et al., 2020]. Therefore, we introduce hierarchical contrastive learning as the second-stage detector, hoping it can further learn the feature representation of disguised machine texts and human texts.

In general, the first-stage detector quickly and roughly screens texts, solving the problem that the detector based on contrastive learning has high training computational overhead when directly facing large-scale text detection and has limited learned human text feature representation when facing data imbalance[Liu et al., 2016, Sohn, 2016]. The detector based on hierarchical contrastive learning, in turn, improves the lack of robustness to fine-grained features of the detector based on cross entropy loss. Therefore, our two-stage detector can achieve better performance in whether detecting human texts and original machine texts or detecting human texts and disguised machine texts. Our experimental results also confirm the correctness of this idea of "blending the strengths of both".

4 Experiments

In this section, we will introduce the experimental process. Section 4.1 will introduce the dataset we used, Section 4.2 will introduce the evaluation metrics we use to evaluate the model, and Section 4.3 will introduce the existing baseline detectors we compare with.

4.1 Datasets

We use three datasets that are widely used for detector training and detection. Detailed dataset information in Appendix C.

CheckGPT[Liu et al., 2024c]: This dataset contains 900,000 samples, generated by ChatGPT based on prompts, covering multiple fields such as news, reviews, and literatures.

HC3[Guo et al., 2023]: A high-quality dataset specifically for fine-tuning dialogue models, containing QA question-answer pairs in multiple fields, each question corresponds to at least one human answer and one machine-generated answer, focusing on multiple open-ended questions such as finance and medicine.

SeqXGPT-Bench[Wang et al., 2023]: A benchmark dataset designed specifically for sentence-level AI generated text detection tasks, containing text generated from multiple LLMs (such as GPT-2, GPT-Neo, GPT-J, LLaMa, and GPT-3). The dataset uses feature alignment design to align word-level log probabilities to a common vocabulary.

4.2 Evaluation metrics

In order to systematically and thoroughly evaluate our work and the work of others, we use Accuracy(ACC), F1-score(F1), and Recall (which is further divided into machine-recall and human-recall) as the standard. ACC: The proportion of correctly predicted samples to the total number of samples, which directly reflects the overall prediction accuracy of the model, but may be distorted when the categories are unbalanced; Recall: The proportion of correctly predicted samples among samples that are actually positive, which measures the model's coverage of positive samples, but may have a high false alarm rate; F1: The harmonic average of precision and recall, which balances Precision and Recall and avoids the one-sidedness of a single indicator. We use the three in combination in the hope of evaluating model performance more comprehensively.

4.3 Baseline Detectors

To verify the effectiveness of our method, we select the following five representative detectors as baselines and compare them with our two-stage detector. Considering that machine texts may be disguised in reality, we specially select three baseline detectors that have gone through adversarial training and text perturbation training.
SimpleAI[Guo et al., 2023]: Fine-tune the pre-trained RoBERTa model, filter the patterned words in the training data to improve generalization ability, and add sentence-level data to enable the model to capture local features.
Watermark[Kirchenbauer et al., 2023]: The watermark embeds the signal by biasing the "green token list" at generation time, and the detector counts the actual number of green tokens in the texts. It is completely independent of the generation model, avoiding the overhead of traditional model training while ensuring the robustness and interpretability of the detection.
CoCo[Liu et al., 2023]:By constructing a coherence graph to capture the entity interaction structure of the text and introducing a supervised contrastive learning framework, the model's understanding of language patterns is enhanced.
RADAR[Hu et al., 2023]: Using the adversarial learning framework of generator and discriminater along with some instruction tuning, model shows excellent robustness and transferability.
PECOLA[Liu et al., 2024b]: The noise introduced by random perturbations is reduced through selective perturbation strategies, the key features of the text are retained, and the contrastive learning strategy is further used to enhance the robustness.

5 Results and Analysis

In this section we present our experimental results and analyze them. Specifically, we will first show the refinement of our detector in the task of detecting human texts and original machine text, then we will show the robustness of our detector in the more realistic task of detecting disguised machine text, and finally we will explore the "patch effect" of our two-stage strategy on existing detectors.

In order to better illustrate the effectiveness of our method, we retrain the existing baseline detectors on the three datasets of CheckGPT, HC3, and SeqXGPT according to the method described in their papers and compare them with our detector to verify the effectiveness and compatibility of our method. Among them, RADAR does not provide source code, so we use the API they provide. In addition, the watermark-based detector does not need to be trained, and its processing work is to add watermarks to the input dataset. We first conduct extensive tests on undisguised machine texts and human texts. The results are shown in Table 1. Our detector and the existing baseline detectors achieve good results on the three datasets. For the HC3 and SeqXGPT datasets, our method outperforms all baseline detectors in five evaluating metrics. For the CheckGPT dataset, we achieve the best in machine-recall, F1, and ACC, and also achieve the second best in overall recall. Further, using the comprehensive evaluating metric of F1 to illustrate, our detector is 4.1% higher than the second place on the CheckGPT dataset and 2.27% higher than the second place on the SeqXGPT dataset. For the HC3 dataset, although it is released relatively early and the recognition difficulty may be relatively low, all baseline detectors perform well, our method still achieves a certain breakthrough, 1.43% higher than the second place. The above experimental results show that the cross-data adaptability of our two-stage framework is commendable, and acquires the most advanced performance in the early and simple task of detecting original machine texts and human texts.

Dataset	Detectors	Metrics				
		Human-recall	Machine-recall	Recall	F1	ACC
CheckGPT	SimpleAI	90.21	87.49	88.85	88.79	88.80
	Watermark	96.65	97.48	**97.06**	72.26	75.69
	CoCo	**97.42**	72.38	84.90	85.97	84.55
	PECOLA	94.69	75.23	84.96	84.51	84.58
	RADAR	68.41	58.11	63.26	62.05	63.05
	Two-stage	87.72	**98.94**	93.33	**92.89**	**93.55**
HC3	SimpleAI	95.66	89.98	94.32	94.31	94.32
	Watermark	90.71	98.78	94.75	95.13	94.88
	CoCo	99.58	99.05	99.31	98.30	98.42
	PECOLA	97.58	99.14	98.36	98.35	98.36
	RADAR	84.86	94.28	89.57	90.39	89.57
	Two-stage	**99.74**	**99.82**	**99.78**	**99.78**	**99.80**
SeqXGPT	SimpleAI	93.52	95.25	94.38	94.37	94.37
	Watermark	none	none	none	none	none
	CoCo	91.74	72.97	82.36	79.54	80.67
	PECOLA	90.35	77.72	84.04	84.04	84.13
	RADAR	75.99	46.31	61.15	54.16	61.37
	Two-stage	**94.35**	**99.04**	**96.70**	**96.64**	**96.66**

Table 1: Performance results of different models across datasets without disguise.The best number is highlighted in **bold**, while the second best one is underlined, other tables are the same.

To further illustrate that our two-stage training paradigm can train a more robust detector, we perturb the machine texts in the CheckGPT, HC3, and SeqXGPT datasets at different levels, and retrain the baseline detector based on the perturbed dataset to compare with our detector. The experimental results are shown in Table 2. The results show that our detector has achieved State-Of-The-Art(SOTA) performance on the three perturbed datasets. In terms of recall, F1, and ACC, our detector is 17.6%, 0.93%, and 3.08% higher than the second place on the HC3 dataset, and 3.58%, 0.96%, and 1.05% higher than the second place on the SeqXGPT dataset. On the CheckGPT dataset, recall and ACC are both the first place, and F1 also reaches the runner-up performance. Furthermore, for the detector CoCo, which also has outstanding performance, although its performance on the CheckGPT dataset is comparable to ours, it relies on the extraction of entities in the texts and the construction of a coherence graph. If the machine texts are relatively concise and short, its performance will suddenly drop because the model cannot build coherence graphs. For example, in our experiments on the HC3 dataset, we find that 10,083 out of 24,000 data do not have corresponding coherence graphs, which leads to a sudden collapse of CoCo's machine-recall metric (23.99%). Our detector performes well on all three datasets, and demonstrates excellent generalization and robustness in the more realistic task of detecting disguised machine texts and human texts.

We further explore the effect of our framework in "patching" existing detectors, given that the first-stage detector plays an important role in the overall detection effect, its initial screening of data directly affects the training and detection of the second-stage detector. We replace the first-stage detector based on traditional classification loss with a detector based on supervised contrastive learning, the framework is consistent with Chen et al. [2022]'s work,we have:

$$\mathcal{L}_{con} = -\sum_{s_t \in \mathcal{S}} \frac{1}{c} \log \left(\frac{\sum\limits_{l_r = l_t} \exp(z_t \cdot z_r / \tau)}{\sum\limits_{l_r = l_t} \exp(s_t \cdot s_r / \tau) + \sum\limits_{l_{r'} \neq l_t} \exp(s_t \cdot s_{r'} / \tau)} \right) \tag{7}$$

where s represents sample, l represents label, c represents the number of samples entering the first stage detector, τ is the temperature coefficient. The final loss function is:

$$\mathcal{L}_{final} = \mathcal{L}_{con} + \mathcal{L}_{ce} \tag{8}$$

Dataset	Detectors	Metrics				
		Human-recall	Machine-recall	Recall	F1	ACC
CheckGPT	SimpleAI	88.55	90.59	89.57	70.19	90.50
	Watermark	53.32	99.82	76.54	69.56	56.22
	CoCo	**97.58**	73.68	85.63	**98.21**	96.60
	PECOLA	93.45	83.55	88.5	62.64	83.98
	RADAR	84.27	60.89	72.71	75.35	61.94
	Two-stage	92.19	**99.98**	**96.08**	95.71	**99.63**
HC3	SimpleAI	43.34	99.82	71.58	79.25	96.44
	Watermark	52.74	99.32	76.03	69.05	55.16
	CoCo	92.36	23.99	58.18	95.09	94.44
	PECOLA	24.80	96.63	60.72	65.09	94.70
	RADAR	76.27	82.12	79.20	89.40	81.75
	Two-stage	**92.53**	**99.98**	**96.26**	**95.99**	**99.52**
SeqXGPT	SimpleAI	64.13	99.34	81.74	86.18	97.04
	Watermark	none	none	none	none	none
	CoCo	88.63	98.68	93.65	93.06	98.16
	PECOLA	24.67	98.70	61.69	65.60	93.87
	RADAR	76.27	57.27	66.77	72.08	58.51
	Two-stage	**94.94**	**99.52**	**97.23**	**94.02**	**99.21**

Table 2: Performance results of different models across datasets under disguise.

Type	Detectors	Metrics				
		Human-recall	Machine-recall	Recall	F1	ACC
Original	Single	**89.48**	92.25	90.86	90.45	91.44
	Joint	88.13	**98.99**	**93.56**	**93.15**	**93.77**
Attacked	Single	**86.90**	94.13	90.52	55.78	93.81
	Joint	85.91	**99.80**	**92.86**	**90.39**	**99.18**

Table 3: Performance results of "patching" results.

Results are shown in Table 3. For the existing detectors based on contrastive learning, the F1 reaches 90.45% and the ACC reaches 91.44% in the task of detecting human texts and original machine texts, which shows that a relatively ideal effect has been achieved in the first screening and classification of texts. On this basis, we further use our second-stage detector to play the role of the second gate in the hope of achieving a more refined effect. The results are consistent with our expectations. The patched detector further improves the F1 to 93.15% and the ACC to 93.77%. Considering that many machine texts in real life often use disguised methods to try to evade the detection of detectors, our patch further improves the recall and ACC for the task of detecting human texts and disguised machine texts. The recall and ACC of machine texts reached nearly perfect 99.80% and 99.18%, indicating that the detector can almost completely identify disguised machine texts. For the F1 score, the detector without the patch only has 55.78%, which shows that it is very sensitive to disguised texts. A large number of machine texts are misjudged as human, resulting in performance degradation. However, using our patch, a second checkpoint is set up in time to capture these disguised texts that bypass detection, and the F1 score is increased by a huge span of 34.61% to 90.39%. This further proves our original intention of using two-stage detection: To give the existing detector robustness when facing large-scale machine disguised texts, and further improve its recognition ability of machine texts so that it can effectively capture both original and disguised texts. Overall, by applying our patch, we achieve a significant improvement in most evaluation metrics, with only a minor sacrifice of 1% in human texts recall, resulting in a substantial overall performance optimization.

6 Ablation Studies

In this section, we will introduce ablation experiments and systematically evaluate the effects of each component. We first split the two-stage detector and apply it separately to the dataset to explore the impact of the composition of each component on the overall effect. Moreover, we train the two detectors directly on the dataset, aiming to verify the effect of "1+1>2".

Type	Detectors	Metrics				
		Human-recall	Machine-recall	Recall	F1	ACC
Original	First	88.75	92.30	90.52	90.07	90.60
	Second	**97.29**	77.00	87.15	63.88	47.18
	Combined	87.72	**98.94**	**93.33**	**92.89**	**93.55**
Attacked	First	86.90	94.13	90.52	55.78	93.81
	Second	**96.82**	82.52	89.67	8.50	5.15
	Combined	92.19	**99.98**	**96.08**	**95.71**	**99.63**

Table 4: Results of ablation. Original means datasets are unprocessed, while Attacked means datasets are disguised. First represents first-stage detector, Second represents second-stage detector, Combined represents the two-stage detector.

Taking the CheckGPT dataset as an example. First, we directly apply the first-stage detector, the second-stage detector, and the joint detector to the two tasks of detecting human texts and original machine texts (origin), detecting human texts and disguised machine texts (second). The results are shown in Table 4. For the first task, the first-stage detector performes well overall, which confirms its applicability and practicality in the simple binary classification task of detecting human texts and original machine texts. The second-stage detector performes relatively averagely, with an F1 of 63.88% and an ACC of only 47.18%. This phenomenon is well explained: the second-stage detector is trained with samples screened by the first-stage detector as data set. If it is directly applied to the original texts detection, the effect may not be satisfactory. This is also the reason why the F1 and ACC scores of the second-stage detector plummets in Task 2. For task 2, although the first-stage detector bears a good recall rate, combined with the imbalance of the dataset under this task (the disguise methods of machine texts are varied) and its poor performance in F1 score (55.78%), indicating that it will misjudge a large number of disguised machine texts as human, which is consistent with our expectation that "the robustness of the first-stage detector is fragile when facing large-scale disguised texts", and is one of the reasons why we introduce the two-stage "patch".
Overall, whether it is detecting original machine text or disguised machine text, deleting any component will lead to a decrease in the overall performance of the detector. For the two-stage detector, it is undoubtedly worthwhile to sacrifice the recall rate of human texts slightly in exchange for a significant improvement in the overall performance.

To further verify the effectiveness of the two-stage joint training method, we directly trains the first-stage and two-stage detectors on the CheckGPT's human texts and original machine texts, and human texts and disguised machine texts, respectively. The results are shown in Table 5. For the first-stage detector, even after training with disguised texts, it still has the phenomenon of "wrongly accusing good guys" when detecting disguised machine texts, which reveals the limitations of a single detector in adversarial scenarios — it can only ensure the stability of some evaluating metrics, but cannot be globally reliable, so it is necessary to fuse multi-dimensional features through a joint detector. For the two-stage detector, in the task of detecting original machine texts, its hierarchical contrastive learning will degenerate into single-level contrastive learning, while in the task of detecting disguised machine texts, its ability to discriminate human texts is slightly weak due to the fact that in the face of a variety of large-scale machine disguised texts in reality, contrastive learning learns limited features of human texts.

Type	Detectors	Metrics				
		Human-recall	Machine-recall	Recall	F1	ACC
Original	First	**88.75**	92.30	90.52	90.07	90.60
	Second	81.10	95.23	88.16	87.09	88.40
	Combined	87.72	**98.94**	**93.33**	**92.89**	**93.55**
Attacked	First	68.54	98.32	83.43	74.95	96.94
	Second	77.00	99.85	88.42	85.51	97.83
	Combined	**92.19**	**99.98**	**96.08**	**95.71**	**99.63**

Table 5: Results of ablation, examining the effect of different parts in detail.

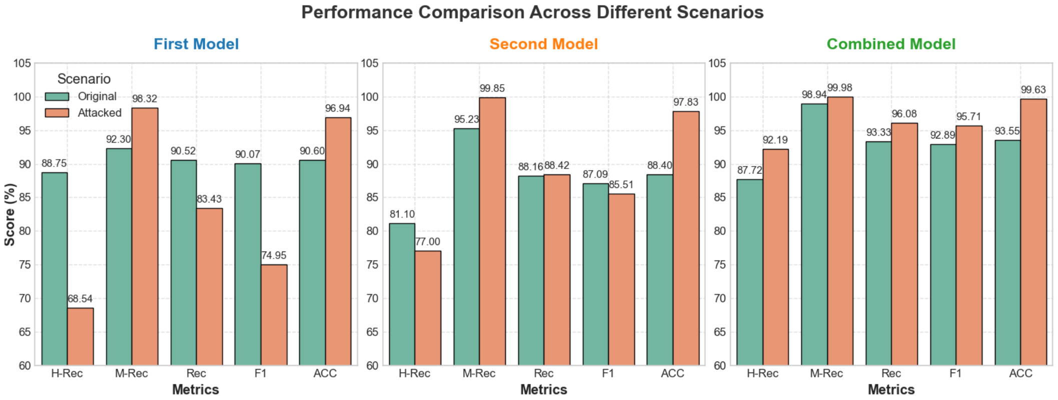

Figure 3: Results of ablation, examining the effect of different parts in detail.

Overall,the two-stage detector is the best of the above two detectors,and has a significant increase in evaluating metrics such as F1 and Recall, which shows that our training idea of "blending the strengths of both" is correct and appliable.

7 Conclusion

In this paper, we propose a coarse-to-fine AI generated text detector model and a novel two-stage detector training paradigm. The first-stage detector quickly screens the original machine texts, and the second-stage detector uses hierarchical contrastive learning to carefully distinguish different levels of disguised machine texts. Our detector has achieved SOTA performance in both the task of detecting human texts from original machine texts and the more realistic task of detecting human texts from disguised machine texts, proving the effectiveness of each component and the correctness of the "blending the strengths of both" idea in ablation studies. We hope that this robust detector can better assist AI text detection in real life, and that this two-stage training framework can bring new maps and ideas to researchers in this field.

8 Acknowledgements

We sincerely thank all the researchers who have worked in this field and laid the foundation for our work, and we are grateful to all the reviewers, area chairs, and all the scholars who read the paper and provided suggestions.

References

Shuyang Cai and Wanyun Cui. Evade chatgpt detectors via a single space, 2023. URL https://arxiv.org/abs/2307.02599.

Junfan Chen, Richong Zhang, Yongyi Mao, and Jie Xu. Contrastnet: A contrastive learning framework for few-shot text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36 (10):10492–10500, Jun. 2022. doi: 10.1609/aaai.v36i10.21292. URL https://ojs.aaai.org/index.php/AAAI/article/view/21292.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. Token prediction as implicit classification to identify llm-generated text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 13112–13120. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.810. URL http://dx.doi.org/10.18653/v1/2023.emnlp-main.810.

Xuxin Cheng, Bowen Cao, Qichen Ye, Zhihong Zhu, Hongxiang Li, and Yuexian Zou. ML-LMCL: Mutual learning and large-margin contrastive learning for improving ASR robustness in spoken language understanding. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6492–6505, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.406. URL https://aclanthology.org/2023.findings-acl.406/.

Claude AI, 2024. URL https://www.ibm.com/think/topics/claude-ai.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Matthew C. Dickson, Anna S. Bosman, and Katherine M. Malan. *Hybridised Loss Functions for Improved Neural Network Generalisation*, page 169–181. Springer International Publishing, 2022. ISBN 9783030933142. doi: 10.1007/978-3-030-93314-2_11. URL http://dx.doi.org/10.1007/978-3-030-93314-2_11.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.674. URL https://aclanthology.org/2024.acl-long.674/.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.552. URL https://aclanthology.org/2021.emnlp-main.552/.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. Gltr: Statistical detection and visualization of generated text, 2019. URL https://arxiv.org/abs/1906.04043.

Gemini. Gemini: A family of highly capable multimodal models, 2024. URL https://arxiv.org/abs/2312.11805.

Chenxi Gu, Chengsong Huang, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. Watermarking pre-trained language models with backdooring. *arXiv preprint arXiv:2210.07543*, 2022.

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

Xun Guo, Shan Zhang, Yongxin He, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. Detective: Detecting ai-generated text via multi-level contrastive learning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 88320–88347. Curran Associates, Inc., 2024.

Julian Hazell. Spear phishing with large language models, 2023. URL https://arxiv.org/abs/2305.06972.

Abe Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. k-SemStamp: A clustering-based semantic watermark for detection of machine-generated text. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1706–1715, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.98. URL https://aclanthology.org/2024.findings-acl.98/.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Radar: Robust ai-text detection via adversarial learning, 2023. URL https://arxiv.org/abs/2307.03838.

Guanhua Huang, Yuchen Zhang, Zhe Li, Yongjian You, Mingze Wang, and Zhouwang Yang. Are AI-generated text detectors robust to adversarial perturbations? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6005–6024, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.327. URL https://aclanthology.org/2024.acl-long.327/.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36, 2024.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703/.

Aiwei Liu, Leyi Pan, Xuming Hu, Shu'ang Li, Lijie Wen, Irwin King, and Philip S. Yu. An unforgeable publicly verifiable watermark for large language models, 2024a. URL https://arxiv.org/abs/2307.16230.

Shengchao Liu, Xiaoming Liu, Yichen Wang, Zehua Cheng, Chengzhengxu Li, Zhaohan Zhang, Yu Lan, and Chao Shen. Does DetectGPT fully utilize perturbation? bridging selective perturbation to fine-tuned contrastive learning detector would be better. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1874–1889, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.103. URL https://aclanthology.org/2024.acl-long.103/.

Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 507–516, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/liud16.html.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. Coco: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning, 2023. URL https://doi.org/10.48550/arXiv.2212.10341.

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

Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. On the detectability of chatgpt content: Benchmarking, methodology, and evaluation through the lens of academic writing, 2024c. URL https://arxiv.org/abs/2306.05524.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. An entropy-based text watermarking detection method. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11724–11735, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.630. URL https://aclanthology.org/2024.acl-long.630/.

Chao Ma, Daniel Kunin, Lei Wu, and Lexing Ying. Beyond the quadratic approximation: the multiscale structure of neural network loss landscapes, 2022. URL https://arxiv.org/abs/2204.11326.

Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. Efficient detection of llm-generated texts with a bayesian surrogate model, 2024. URL https://arxiv.org/abs/2305.16617.

Niloofar Mireshghallah, Justus Mattern, Sicun Gao, Reza Shokri, and Taylor Berg-Kirkpatrick. Smaller language models are better black-box machine-generated text detectors, 2024. URL https://arxiv.org/abs/2305.09859.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/mitchell23a.html.

Roe Perkins, Mike. 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), October 2023. ISSN 1572-8544. doi: 10.1007/s10805-023-09492-6. URL http://dx.doi.org/10.1007/s10805-023-09492-6.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. Release strategies and the social impacts of language models, 2019. URL https://arxiv.org/abs/1908.09203.

Rafael Alberto Rivera Soto, Kailin Koch, Aleem Khan, Barry Y. Chen, Marcus Bishop, and Nicholas Andrews. Few-shot detection of machine-generated text using style representations. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=cWiEN1plhJ.

Rafael Rivera Soto, Kailin Koch, Aleem Khan, Barry Chen, Marcus Bishop, and Nicholas Andrews. Few-shot detection of machine-generated text using style representations, 2024b. URL https://arxiv.org/abs/2401.06712.

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

Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Edward Tian and Alexander Cui. Gptzero: Towards detection of ai-generated text using zero-shot and supervised methods, 2023. URL https://example.com.

Jörg Tiedemann and Santhosh Thottingal. OPUS-MT – building open translation services for the world. In André Martins, Helena Moniz, Sara Fumega, Bruno Martins, Fernando Batista, Luisa Coheur, Carla Parra, Isabel Trancoso, Marco Turchi, Arianna Bisazza, Joss Moorkens, Ana Guerberof, Mary Nurminen, Lena Marg, and Mikel L. Forcada, editors, *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal, November 2020. European Association for Machine Translation. URL https://aclanthology.org/2020.eamt-1.61/.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. SeqXGPT: Sentence-level AI-generated text detection. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1144–1156, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.emnlp-main.73/.

Yichen Wang, Shangbin Feng, Abe Hou, Xiao Pu, Chao Shen, Xiaoming Liu, Yulia Tsvetkov, and Tianxing He. Stumbling blocks: Stress testing the robustness of machine-generated text detectors under attacks. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2894–2925, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.160. URL https://aclanthology.org/2024.acl-long.160/.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, Julia Haas, Sean Legassick, Geoffrey Irving, and Iason Gabriel. Taxonomy of risks posed by language models. FAccT '22, page 214–229, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393522. doi: 10.1145/3531146.3533088. URL https://doi.org/10.1145/3531146.3533088.

Danny Wood, Tingting Mu, and Gavin Brown. Bias-variance decompositions for margin losses. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 1975–2001. PMLR, 28–30 Mar 2022. URL https://proceedings.mlr.press/v151/wood22a.html.

Dejiao Zhang, Wei Xiao, Henghui Zhu, Xiaofei Ma, and Andrew Arnold. Virtual augmentation supported contrastive learning of sentence representations. In Smaranda Muresan, Preslav Nakov, and

Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 864–876, Dublin, Ireland, May 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.70. URL https://aclanthology.org/2022.findings-acl.70/.

Yanzhao Zhang, Richong Zhang, Samuel Mensah, Xudong Liu, and Yongyi Mao. Unsupervised sentence representation via contrastive learning with mixing negatives. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11730–11738, Jun. 2022b. doi: 10.1609/aaai.v36i10.21428. URL https://ojs.aaai.org/index.php/AAAI/article/view/21428.

Ying Zhou, Ben He, and Le Sun. Navigating the shadows: Unveiling effective disturbances for Modern AI content detectors. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10847–10861, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.584. URL https://aclanthology.org/2024.acl-long.584/.

Biru Zhu, Lifan Yuan, Ganqu Cui, Yangyi Chen, Chong Fu, Bingxiang He, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. Beat LLMs at their own game: Zero-shot LLM-generated text detection via querying ChatGPT. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7470–7483, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.463. URL https://aclanthology.org/2023.emnlp-main.463/.

A Broader Impacts

The rapid development of LLMs has enabled a large amount of machine-generated texts to be obtained quickly and at low cost. Given that it may lead to academic fraud, phishing emails, the spread of false information and other problems, detecting and monitoring AI-generated texts is undoubtedly a top priority. However, due to the fragility of current AI content detectors and the diversity of text disguise methods, machine texts can easily bypass detection through disguise. Therefore, the development of robust AI content detectors is urgent. Our paper introduces a new robust AI content detector training paradigm, which demonstrates SOTA performance in multiple benchmarks. These advances will bring the green development and use of LLMs with new power. In addition, our second-stage detector can be used as a "patch" to further improve the performance of current detectors when facing large-scale disguised machine texts, which shows that our method has broad prospects for practical application and rich significance.

B Limitations and Future Work

In this paper, we take into consideration that machine texts may use different disguise methods to evade the detector and thus use hierarchical contrastive learning to strengthen the detector in a targeted manner. However, the original machine texts generated by different models often has certain differences, which may affect the performance of the detector. In addition, we did not introduce some latest disguise strategies (such as adding emoticons to machine texts) and did not train on a larger corpus. In the future, we will continue to work in this direction and further improve the performance of the model.

C Detailed Construction of Dataset

Dataset	Train	Test	Valid
CheckGPT	(2000,2000)	(1921,2078)	(2500,2500)
HC3	(5000,5000)	(5000,5000)	(2000,2000)
SeqXGPT	(2467,2533)	(1928,1872)	(1005,995)

Table 6: Detailed composition of the dataset for detecting human texts and original machine texts.

Watermark	Human	Machine
CheckGPT	566	570
HC3	438	538
SeqXGPT	600	520

Table 7: Detailed composition of the dataset for watermarks.

Dataset	Train	Test	Valid
CheckGPT	(500,8500)	(1101,23887)	(500,8490)
HC3	(500,7500)	(1500,22500)	(500,7500)
SeqXGPT	(500,7500)	(1500,21004)	(500,6494)

Table 8: Detailed composition of the dataset for detecting human texts and attacked machine texts.

Watermark	Human	Machine
CheckGPT	566	8550
HC3	438	8098
SeqXGPT	600	520

Table 9: Detailed composition of the attacked dataset for watermarks.

For the simple binary classification task of detecting human texts and original machine texts, the distribution of our samples is shown in Table 6. We ensure the distribution of human texts and machine texts is approximately 1:1. The two-tuple (human, machine) in the table represents the number of human texts and the number of machine texts. For the watermark detector, since it focuses on the hidden singal embedded in the data and does not require training, it only needs to build a test set. While it takes a long time to process the watermark on the dataset, we did not generate a large test set. The data are shown in Table 7. For the more realistic task of detecting human texts and disguised machine texts, the distribution of our samples is shown in Table 8. We select 500 human texts and 500 original machine texts from the dataset respectively, and perform 16 disguises methods on the machine texts at the character, word, sentence, and paragraph levels, thereby constructing a

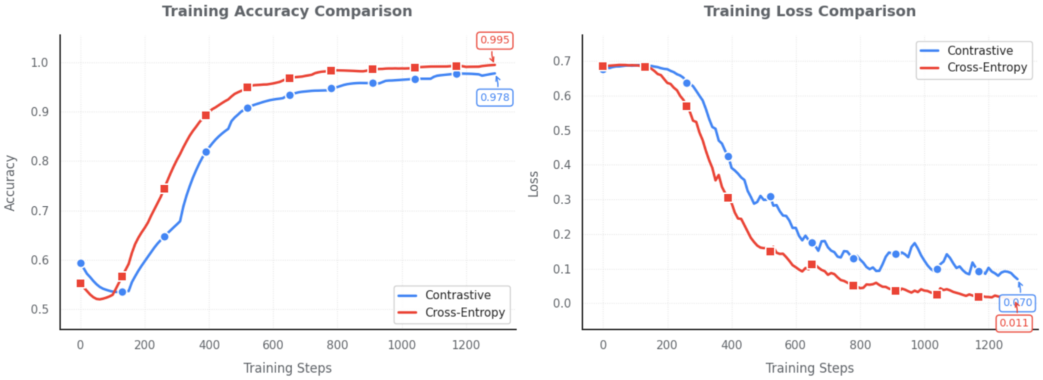

Figure 4: Comparison of contrastive learning and cross entropy on training accuracy and loss.

perturbation dataset containing human texts, original machine texts, and disguised machine texts. Therefore, the perturbed datasets are mostly composed of machine texts, which are consistent with the current trend of a variety of machine text disguise methods and a flood of generation sources. For the watermark detector, similarly, after the machine texts are injected with the watermark, we disguise them in sixteen different ways and explore whether these disguise strategies will cause the watermark to be covered and invalid. The data distribution is shown in Table 9.

D Comparison of Cross Entropy Loss and Contrastive Learning Loss

We take the SeqXGPT dataset as an example, recording the training accuracy and loss of the two models when using cross entropy loss and contrastive learning loss to train the model respectively, experimental results are shown in Figure 4. The results show that the model loss based on cross entropy loss converges faster and is more stable, and for the task of detecting human texts and original machine texts, its training accuracy is higher than that of the model based on contrastive learning, which is one of the reasons why we choose the cross entropy-based classifier model as the first-stage detector.