DAMAGE: Detecting Adversarially Modified AI Generated Text

Elyas Masrour and **Bradley Emi** and **Max Spero**
Pangram Labs, Inc.
Correspondence: info@pangram.com

Abstract

AI humanizers are a new class of online software tools meant to paraphrase and rewrite AI-generated text in a way that allows them to evade AI detection software. We study 19 AI humanizer and paraphrasing tools and qualitatively assess their effects and faithfulness in preserving the meaning of the original text. We show that many existing AI detectors fail to detect humanized text. Finally, we demonstrate a robust model that can detect humanized AI text while maintaining a low false positive rate using a data-centric augmentation approach. We attack our own detector, training our own fine-tuned model optimized against our detector's predictions, and show that our detector's cross-humanizer generalization is sufficient to remain robust to this attack.

1 Introduction

The ability of large language models such as ChatGPT (OpenAI, 2023) to generate realistic and fluent text has spurred the need for AI text detection software. Commercial methods, such as TurnItIn, GPTZero, Originality, and Pangram Labs have emerged, as well as open-source research methods, such as DetectGPT (Mitchell et al., 2023), Binoculars (Hans et al., 2024), and many more.

However, both researchers and practitioners alike have found these solutions to be fragile. A study from Google Research (Krishna et al., 2023) found that a paraphrasing text-to-text model (a variant of T5) was able to effectively rewrite AI-generated text in a way that could preserve the meaning of the original text but largely evade AI detection algorithms.

This finding gave rise to an explosion of new AI "humanizer" tools appearing online. These tools promise to bypass AI detection tools by rewriting AI-generated text. They are primarily marketed at students, who can use these tools to effectively cheat on writing assignments by plagiarizing from large language models without getting caught. Other humanizers target their product towards SEO marketers, who may generate hundreds of blog posts using AI and apply humanizers to evade AI detection by search engine algorithms.

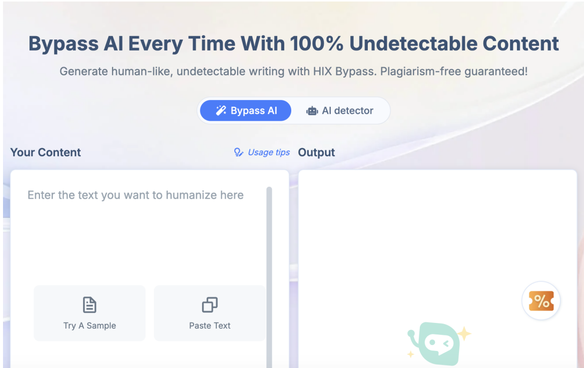

Figure 1: Example of an AI humanizer tool

In this work, we attempt to comprehensively study these AI humanizers: what they are doing, and whether it is possible to identify humanized AI-generated text. Our main contributions are as follows.

- We qualitatively audit 19 humanizers and paraphrasing tools and analyze their effects on the underlying text. We exhaustively identify the transformation modes that the humanizers apply to their inputs. We categorize the humanizers into three tiers based on their overall quality.
- We study the baseline effectiveness of humanizers in bypassing existing open-source and commercial AI detectors.
- We present a deep-learning based AI detector that effectively is robust to humanization, even by humanizers unseen during training. We describe the necessity of treating humanizer robustness as a learned invariance rather than a separate domain.

Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), pages 120–133
January 19, 2025.

- We show that even after a detector-specific fine-tuning attack, our detector remains fairly robust due to its underlying ability to generalize.

2 Related Work

2.1 AI Detection

Many commercial and open-source methods exist to detect AI-generated text, with highly varying levels of accuracy. One of the most notable commercial solutions is TurnItIn (Staff, 2024), which is widely used in higher education as anti-plagiarism software. Our team at Pangram Labs (Emi and Spero, 2024) is contributing to this field, alongside other solutions such as GPTZero (Tian and Cui, 2023), Originality, and Copyleaks, although their accuracies vary significantly (Weber-Wulff et al., 2023).

Open-source methods typically fall into two categories: perplexity-based detection methods and deep learning based methods. Perplexity-based methods attempt to leverage the fact that the tokens in LLM-based outputs in general will be predicted as consistently more likely by the LLM itself. DetectGPT (Mitchell et al., 2023) and FastDetectGPT (Bao et al., 2024) are earlier examples of perplexity-based methods which look at the local curvature in probability space around a given example. Binoculars (Hans et al., 2024) is an even more effective recent approach which uses the cross-perplexity between two different LLMs as a signal that text is LLM-generated.

Deep learning based methods attempt to use neural networks to detect AI-generated content, leveraging large datasets containing known human and AI text and training a classifier to distinguish between them. The OpenAI classifier (Solaiman et al., 2019) was one of the first efforts. They used a RoBERTa based model to classify human text and GPT-2 written text. Ghostbusters (Verma et al., 2023) uses learned combinations of features derived from language model embeddings to detect LLM-generated text.

Recently, some AI detection efforts have also attempted to detect mixed AI and human text: when some of the text is written by a human and some of it is written by an AI. SeqXGPT (Wang et al., 2023) attempts to solve this by using an architecture which is able to detect AI on the sentence level rather than the document level. ROFT (Kushnareva et al., 2024) adapts several detection methods to detecting the boundary between AI and human text. However, these methods differ from ours in that the assumption about the original document is that each part of the text has a distinct authorship attribution, whereas we study the case in which fully AI-generated text is then modified by a humanizer.

2.2 Evading AI Detection

Much of the literature has also focused on whether or not AI-generated text can be detected at all (Sadasivan et al., 2023). A study from Google Research (Krishna et al., 2023) released DIPPER: a paraphrasing T5-based model that is able to bypass some of the above-mentioned detectors by rewriting the input text. Another group of researchers (Chakraborty et al., 2023) devised a framework to rank LLMs based on their "detectability", claiming that more recent models like GPT-4 are less detectable because perplexity and burstiness are less useful evidence markers.

Furthermore, other research has focused on attacking AI detectors or otherwise methods to bypass or evade AI detection. One study (Kumarage et al., 2023) designs an approach to search for soft prompts that can produce text that can evade detection. Another study (Ayoobi et al., 2024) looks at the effect on AI detectors of translating AI-generated text through multiple languages before backtranslating it into English and find some methods are significantly more robust than others. Another paper directly optimizes a language model by using an AI detector as negative reward: creating pairs of LLM-generated text where one piece is detected and one is not, and then using DPO to optimize the language model to prefer undetected outputs (Nicks et al., 2024). RADAR (Hu et al., 2023) adversarially trains a language model detector and a paraphraser against each other to create a more robust detector.

2.3 Watermarking

Watermarking AI-generated text is another relevant subfield of research. Existing watermarking schemes train or decode LLMs to leave behind a probabilistic signal that can later be detected by a watermark-specific detector. One watermarking scheme (Kirchenbauer et al., 2023) introduces the idea of "green tokens", which are sampled with higher probability than other tokens in a traceable way. Google's recently released SynthID (Google DeepMind, 2024) works in a similar fashion.

We argue that watermarking is insufficient to

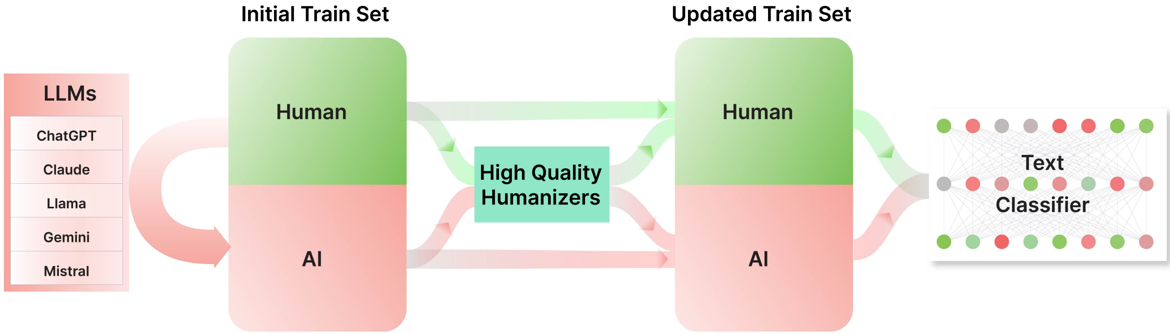

Figure 2: Augmenting the training set with high quality humanizer data improves robustness.

guard against the dangers of AI-generated text. We show that in addition to evading AI detectors, humanizers are also reliable methods to remove such statistical watermarks.

2.4 Benchmarking

Recently there has also been an increased effort to benchmark the performance of various AI detectors against each other. RAID (Dugan et al., 2024) is a live leaderboard measuring the performance of AI-generated text detection methods against each other on multiple domains, models, and adversarial attacks. We include the RAID paraphrase and synonym splits in our results as proxies for measuring robustness against humanizers, but we also include a more diverse set of humanizer attacks than the original RAID benchmark.

3 Humanizer Market Survey

3.1 Tool Research and Selection

We selected 19 humanizers and paraphrasing tools based on search popularity and academic relevance. The particular paraphrasers and humanizers selected are presented in Table 1. Notably, we include DIPPER (Krishna et al., 2023) as a paraphraser, due to the authors' claim that text modified by DIPPER can universally bypass AI detection methods.

3.2 Humanizers are often themselves LLMs

Some humanizers are LLMs with system prompts instructing the LLM to write more like a human, or fine-tuned versions of LLMs. In testing some of the humanizers, we found that some of them are susceptible to popular jailbreaks. When we tested one popular humanizer and asked it to give us its system prompt, it said "I should respond to the user input with a reasonable approximation of the full meaning of the input...I should respond in a conversational tone." More examples of our jailbreaks against LLM-based humanizers can be found in Appendix A.

3.3 Humanizers are popular on the GPT Store

As of the date of publication, two out of the four most popular Writing Custom GPTs in the OpenAI GPT Store are humanizers that make function calls to external humanizers. This indicates that there is a large appetite for bypassing AI detection. Given that a significant portion of ChatGPT's daily active users are students, it is likely that these tools are popular for cheating or otherwise making AI writing undetectable. We believe that although many of these humanizers are black boxes, they are an important and understudied area for research in AI detection.

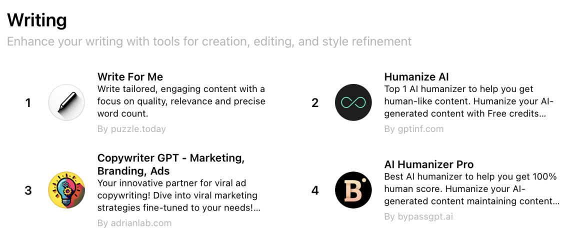

Figure 3: Two out of the four most popular Writing Custom GPTs are Humanizers

3.4 Humanizers are capable of removing watermarks

Google's SynthID is a state-of-the-art solution for watermarking generated text. Following the methodology and code from the SynthID paper (Google DeepMind, 2024), we generated 1000 watermarked texts and 1000 unwatermarked texts. We used Gemma-2B-IT (Team et al., 2024) to generate 200 tokens for each example with a temperature

Category	Tools
Paraphrasers	DIPPER, Grammarly, Quillbot
Humanizers	Bypass GPT, Ghost AI, HIX Bypass, Humbot AI, HumanizeAI.io, HumanizeAI.pro, Humanizer.com, Phrasly.ai, Semihuman AI, StealthGPT, StealthWriter.AI, Surfer SEO, Undetectable AI, Twixify, WriteHuman.ai

Table 1: Paraphrasers and Humanizers Studied

of 1.0, using the ELI5 dataset (Fan et al., 2019) as prompts. We used the unwatermarked text to set a FPR threshold and evaluated SynthID watermark detection TPR at a fixed FPR. Finally, we paraphrased the watermarked text with DIPPER and reevaluated watermark detection, finding that watermark detection had dropped dramatically. See results in Table 2.

Watermarked Gemma-2B-IT	
TPR @ FPR=5%	87.6%
TPR @ FPR=1%	66.5%
After DIPPER Paraphrase	
TPR @ FPR=5%	5.4%
TPR @ FPR=1%	1.5%

Table 2: Watermark detection before and after paraphrasing

4 Humanized Text Audit

4.1 Approach

To understand the effect of humanizing a given piece of text, we engaged in a manual qualitative analysis. We reviewed several samples of text per humanizer, examining how the humanizer transformed vocabulary, sentence structure, and grammar. While not exhaustive, we detail some common patterns introduced by humanizers into the text.

4.2 Insight: Nonsensical Phrases

Many poor-quality humanizers add nonsensical text throughout the piece. Common patterns include:

- Hallucinated Citations:

  `...community service for demonstrating consciousness about public affairs together with responsibility for own actions (Westwood, 2013)...`

- In-line comments:

  `...in specified locations hence constructing external frames those encouraging individuals manage their own times wisely (??????)...`

- Other Nonsensical Phrases:

  `...he or she will never seem defeated by teachers' demands and, as a result, will put more effort into their studies. CGSizeMake pp 18-23...`

4.3 Insight: Varying Structural Continuity

Some humanizers retain low-level sentence structure and simply replace individual words with synonyms. For example, the paraphraser in Figure 4 preserves the meaning of each individual sentence, and even sometimes preserves the phrasing structure within the single sentence, explicitly highlighting that only some words and short phrases have been replaced with synonyms.

Other humanizers take more liberty to change the original text, sometimes rewording entire groups of sentences and paragraphs. Some add more sentences that weren't originally present or delete redundant sentences. We notice that humanizers built on LLMs tend to be more weakly grounded in the original text, while rules-based humanizers that do synonym replacement tend to be more strongly grounded in the original text.

4.4 Insight: Writing and Vocabulary Level

Some humanizers write exclusively in an academic, formal, and/or university level tone. Others write at the elementary school, middle school, or high school level. The better humanizers, usually the ones that are LLM-based, do not commit to a specific writing level or tone, and instead adopt the writing level and tone of the original document.

4.5 Humanizer Segmentation

During our audit, we grouped humanizers into three distinct tiers. The best humanizers rewrote text preserving its tone, vocabulary level, and complexity. Average humanizers rewrote text in a way that degraded overall quality, but preserved intent and message. Low quality humanizers often added

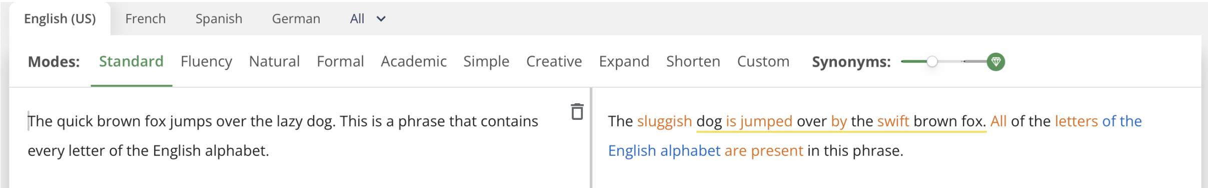

Figure 4: This paraphraser performs a very close paraphrase, only replacing individual words and phrases rather than rewriting entire sentences and paragraphs.

nonsensical phrases, words, and characters, constructed incorrect and uninterpretable sentences, and often distorted the meaning of the text. We classify these three categories of humanizers as L1 (best), L2 (medium), and L3 (worst) humanizers, and describe their characteristics in Figure 5. We present a full categorization, along with notes on the specific qualities of each humanizer, in the Appendix. It is worth noting we only make these classifications based on faithfulness and fluency, not on their effectiveness at bypassing AI detectors.

4.6 Quantifying Humanizer Fluency

To quantify the difference between L1, L2, and L3 humanizers, we use the Fluency Win Rate metric introduced in (Nicks et al., 2024). We prompt GPT-4o to select the more fluent and coherent sample: an original chunk of text, or that chunk of text passed through a humanizer. Here, we report the rate at which GPT-4o selects the humanized sample as the more coherent one. Using a dataset of 25 samples per humanizer, we aggregate the win rate of each tier. L1 humanizers had an average Win Rate of 26.0%, L2 humanizers had an average Win Rate of 14.67%, and L3 humanizers had an average Win Rate of 2.67%.

This demonstrates that our qualitative audit agrees with the fluency metric. Further, all humanizers tend to degrade the quality of the original text, but the degree of quality degradation varies. Still, for the highest quality humanizers, the text quality is still sometimes comparable to the highest quality language model outputs. Because certain humanizers are able to produce high-fluency text, we believe there is a growing need to study them.

5 Experiments

In our experiments, we seek to answer the question of whether a deep learning based AI text classifier is capable of detecting humanized AI-generated text. First, we narrow our scope to L1 humanizers. We do this because their subtle changes are the hardest to detect by eye and because they have the highest levels of fluency, making them most relevant in real-world adversarial attacks. We train two models: one model is unaware of humanized text, and one model contains a small amount of humanized text from a variety of humanizers. We describe the methodology and training procedure for training these models here.

5.1 Dataset Creation

5.1.1 Initial Datasets

Our initial dataset is seeded with a wide variety of human-written datasets from prior to 2022. We use datasets from the following domains: reviews, news, general web text, email, student writing/essays, creative writing, questions and answers, ELL/ESL (English as a Second Language), scientific/medical papers, Project Gutenberg, and Wikipedia.

For evaluation, because humanizers are primarily marketed at students, we evaluate all models on several open datasets comprised of student-written essays. Because previous studies (Liang et al., 2023) have found that AI detectors can be biased against nonnative English speaking students, we ensure that a significant portion of our evaluation dataset is comprised of ESL essays. The component datasets in our evaluation and our algorithm for generating the AI essays used in our benchmark are listed in Appendix B.

5.2 Synthetic Data Creation

Our initial dataset fully human-written. To generate the AI side of the dataset, we use synthetic mirror prompts as described in (Emi and Spero, 2024).

We define the term "mirror prompt" to be a prompt based on the original example that is used to generated a "synthetic mirror" example. The goal of each mirror prompt is to generate an example that matches the topic and length of the original document.

If the original document is "<original review>",

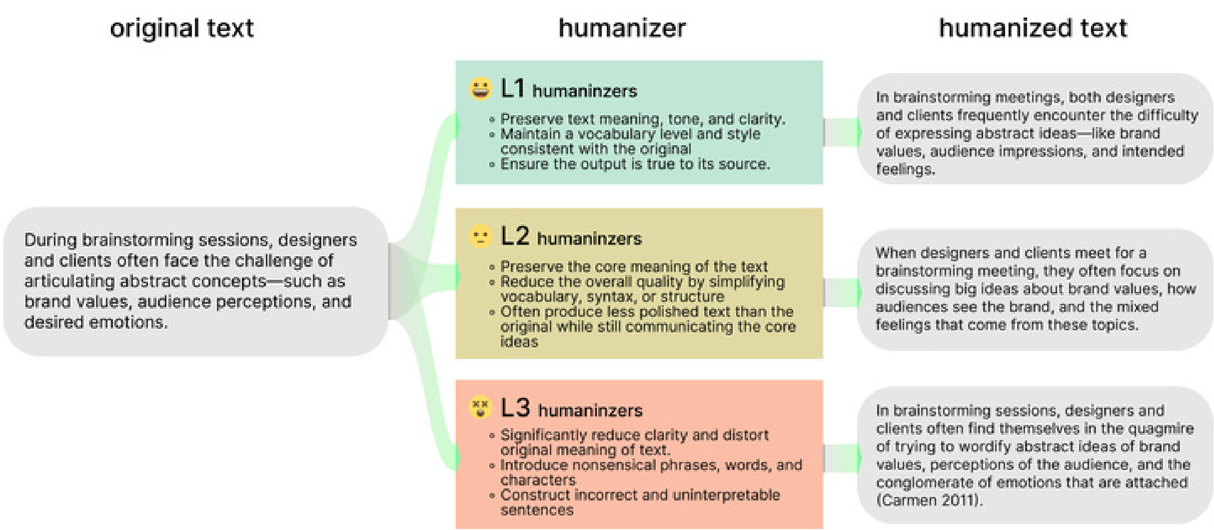

Figure 5: We segment humanizers into three tiers, based on their fluency.

then a mirror prompt may look like this:

[Prompt] Write a <original review star rating> star review for <original review business name>. Make the review around <original review length> words long.

Another example may be for a student essay. We sometimes use double prompts, such as the following:

[Prompt] What is a good title for this essay? <original essay> Only give the title in your response.

[Assistant] <Title>

[Prompt] Write an essay with the following title: <Title>. Make the essay around <original essay length> words long.

5.2.1 LLMs used for Synthetic Mirrors

For synthetic mirrors in the initial training stage, we use the following LLMs:

- GPT-3.5 (multiple versions)
- GPT-4, GPT-4-turbo, and GPT-4o (multiple versions)
- Claude 2 and 3 (multiple versions and sizes)
- LLaMA 2, 3, and 3.1 (multiple versions and sizes)
- Mistral (multiple versions and sizes)
- Gemini Pro and Flash (multiple versions)

It is notable that we only use modern LLMs that are instruction-tuned and post-trained. We do not train on base models because they produce noticeably lower-quality outputs and are substantially less commonly used in real-world applications.

5.3 Architecture and Training

We use the Mistral NeMo architecture (Mistral AI Team, 2024) which has approximately 12 billion parameters, with an untrained linear classification head. Following the usual convention for sequence classification modeling using an autoregressive language model, the hidden state from the final token in the sequence is used as the input to both classification heads. As is common practice in LLM fine-tuning, we use trainable LoRA (Hu et al., 2022) adapters while keeping the base model frozen. We use the Tekken tokenizer out of the box, which is noted for its strong multilingual performance. We truncate the context window to 512 tokens to constrain the model to using only short-range features. When necessary, we simply crop the input to fit the context window. We train the model to convergence using 8 A100 GPUs with a batch size of 24 using a weighted cross entropy loss and the AdamW optimizer for 1 epoch. We early stop based on the weighted cross entropy loss on the validation set.

5.4 Humanizer Data Augmentation

In order to make the treatment model robust to humanization, we treat humanization as a transform on the input data which we would like the model to learn an invariance to.

Because most of the humanizers are marketed at students, we assume that they work best on student writing. As a proxy for student writing, we use the Fineweb-EDU dataset (Lozhkov et al., 2024), a high quality LLM corpus that is prefiltered to only contain documents that are of high educational quality. First, we use an LLM filter to reject all documents that are not standard prose written in full complete sentences. Then, we create synthetic mirrors as described above.

We find that it is best to humanize both human and AI documents before augmenting the training set with this additional data. We also find that if we include all humanizers in the augmentation, our precision (i.e. false positive rate) is significantly compromised. However, including only L1 humanizers (the high quality humanizers) allows us to maintain a low false positive rate in addition to increasing recall generally across humanizers. Further information can be found in Table 8.

After humanizing both human and AI essays, we apply chunking logic to divide each document into roughly 300-word chunks before adding both the human and the AI humanized documents back into the training set.

Even though human documents transformed by a humanizer could be considered AI-generated, we choose to label them as human for the purposes of training. The reason for this is because we treat the model's response to humanization as an invariance rather than only including the AI humanized documents as a separate domain. This contributes to our final performance, as seen in Table 8.

Because most humanizers impose monthly limits on the amount of text that can be humanized, we only use a volume of data up to the limit of the basic 1 month subscription on each humanizer website. As a result, our data volume is quite small: about 0.68% percent of the final dataset is comprised of humanized text. To compensate for the small data volume, we oversample the humanizer data by a factor of 18.

5.5 Active Learning

After training, following the procedure in (Emi and Spero, 2024), we run hard negative mining with synthetic mirrors. On a large corpus of human text, we mine for false positives, and then incorporate both the false positives and their AI mirrors back into our training set. This further reduces our false positive rate and improves our recall. We also incorporate a small amount of data from the RAID train subset into the final training run to generalize to the diversity of models present in the RAID benchmark.

6 Results

6.1 Performance on Humanized Data

Table 3 presents performance data from several AI detection methods on a benchmark of AI-generated academic text before and after humanization. We define a "positive" sample as one that is written by AI, and a "negative" sample as one that is written by a human. Results are presented as true positive rate at a fixed false positive rate of 5%. LLM Baseline in this case is our baseline AI detection model that is trained using synthetic mirrors but does not include any humanized data in its training set.

	AI Text	**Humanized AI Text**
GPTZero	$99.73\% \pm 0.19\%$	$60.04\% \pm 1.80\%$
RADAR	$3.33\% \pm 0.65\%$	$5.05\% \pm 0.81\%$
Binoculars	$94.15\% \pm 0.88\%$	$28.23\% \pm 1.62\%$
LLM Baseline	$100.00\% \pm 0.0\%$	$95.74\% \pm 0.71\%$
DAMAGE	$100.00\% \pm 0.0\%$	$98.26\% \pm 0.47\%$

Table 3: TPR @ FPR=5% for Academic Text with 1000 iterations of bootstrap sampling. RADAR performs poorly on this metric due to its high false positive rate. In Appendix C, we include more metrics, including using model default thresholds.

6.2 Performance on RAID Attacks

Table 4 presents performance data from the same AI detection methods on two adversarial subsets of the RAID benchmark, which includes a range of LLMs and a range of text domains.

	Paraphrase	**Synonym**
GPTZero	64.0%	61.0%
RADAR	62.4%	62.7%
Binoculars	80.3%	43.5%
LLM Baseline	91.6%	96.2%
DAMAGE	93.0%	97.0%

Table 4: TPR @ FPR=5% for Academic Text

7 Detector-Specific Adversarial Humanization

In this paper, we study commercial online humanizers meant to *generally* evade AI detectors, but we also study the directly adversarial scenario: when a humanizer is directly optimized against a particular detector. To do so, we train our own humanizer using the GPT-4o fine-tuning API and measure the detector's robustness to AI-generated text passed through the adversarial fine-tuned model.

7.1 Methodology

Broadly following the methodology in Language Models are Easily Optimized Against (Nicks et al., 2024), we train a model using our detector's AI prediction as a negative signal. However, rather than training a separate language model with DPO, we train a humanizer that takes an unmodified AI-generated text as an input and learns to generate a paraphrase of the original that bypasses the detector. We choose this methodology as it is closer to how humanizers are trained in the real world.

As a proof of concept experiment, we split our essays dataset into two pools: a fine-tuning set and a test set. We select all L1 humanizer outputs from the fine-tuning set that the detector predicts as human-generated (i.e., all L1 humanizer false negatives). We then take the original AI-generated text source (prior to humanization), and create pairs of unhumanized-humanized text samples to fine-tune on. We then use the GPT-4o fine-tuning API to train a new model on only these pairs. This results in a new model that, in theory, learns to paraphrase text into similar examples to the humanizer samples that were able to bypass the detector originally.

7.2 Results

After training the detector-specific humanizer, we use GPT-4o to create synthetic mirrors of 2000 examples from the test set and pass them through the adversarial humanizer.

Condition	TPR @ FPR=5%	Default TPR
No Humanizer	100%	100%
Adversarial Humanizer	100%	93.2%

Table 5: TPR values for DAMAGE model at the default threshold (0.5) and threshold that corresponds to 5% FPR without and without adversarial humanization.

We see that although some of the samples that were adversarially humanized are able to bypass the detector, the detector is still able to detect 93.2% of the humanized AI samples. This shows that although an adversary may be able to directly optimize a humanizer using the detector as a signal, the resulting humanizer still remains largely detectable even without retraining our detector. We believe this is due to the fact that the underlying language model of the humanizer still leaves behind detectable patterns that cannot be erased during fine-tuning. Additionally, by seeing many humanized examples in its training set, the detector learns to be robust to a wide variety of paraphrases and perturbations and maintains its ability to generalize across humanizers, even adversarial ones.

7.3 Detector-Specific Humanizer Fluency Win Rate

To examine the quality of the text generated by our Detector-Specific Humanizer, we perform the same Fluency Win Rate metric introduced in (Nicks et al., 2024). We find that the humanizer wins 14.13% of the time, which is roughly equivalent to an average L2 Humanizer.

8 Conclusion

In this study, we attempted to both qualitatively and quantitatively study humanizer tools and their effects on individual writing samples and the robustness of AI detectors against them. We showed it is possible to train an effective deep learning classifier that is robust to most humanizers, and ran a proof-of-concept to show that some degree of robustness even extends to a humanizer that is adversarially optimized against our specific detector.

References

Navid Ayoobi, Lily Knab, Wen Cheng, David Pantoja, Hamidreza Alikhani, Sylvain Flamant, Jin Kim, and Arjun Mukherjee. 2024. Esperanto: Evaluating synthesized phrases to enhance robustness in ai detection for text origination. *Preprint*, arXiv:2409.14285.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *Preprint*, arXiv:2310.05130.

Megha Chakraborty, S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Krish Sharma, Niyar R Barman, Chandan Gupta, Shreya Gautam, Tanay Kumar, Vinija Jain, Aman Chadha, Amit P. Sheth, and Amitava Das. 2023. Counter turing test ct$\hat{2}$: Ai-generated text detection is not as easy as you may think – introducing ai detectability index. *Preprint*, arXiv:2310.05030.

Scott A. Crossley, Perpetual Baffour, Yingqi Tian, Abigail Franklin, Margaret Benner, and Ulrich Boser. 2024. A large-scale corpus for assessing written argumentation: PERSUADE 2.0. *Assessing Writing*, 61. Dataset comprising over 25,000 argumentative essays from 6th-12th grade students, released under CC BY-NC-SA 4.0.

Scott A. Crossley, Yuan Tian, Perpetual Baffour, Alicia Franklin, Yeoil Kim, William Morris, Brooke Benner, Auriane Picou, and Ulrich Boser. 2023. Measuring second language proficiency using the English Language Learner Insight, Proficiency and Skills Evaluation (ELLIPSE) Corpus. *International Journal of Learner Corpus Research*, 9(2):248–269.

Liam Dugan, Alyssa Hwang, Filip Trhlik, Josh Magnus Ludan, Andrew Zhu, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. Raid: A shared benchmark for robust evaluation of machine-generated text detectors. *Preprint*, arXiv:2405.07940.

Bradley Emi and Max Spero. 2024. Technical report on the pangram ai-generated text classifier. *Preprint*, arXiv:2402.14873.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *Preprint*, arXiv:1907.09190.

Alex Franklin, asiegel, HCL-Jevster, Jules King, julianmante, Maggie, Perpetual Baffour, Ryan Holbrook, and Scott Crossley. 2023. Commonlit - evaluate student summaries. `https://kaggle.com/competitions/commonlit-evaluate-student-summaries`. Kaggle.

Google DeepMind. 2024. Synthid: Identifying ai-generated content with digital watermarks. Accessed: 2024-11-14.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. *Preprint*, arXiv:2401.12070.

Langdon Holmes, Scott Crossley, Perpetual Baffour, Jules King, Lauryn Burleigh, Maggie Demkin, Ryan Holbrook, Walter Reade, and Addison Howard. 2024. The learning agency lab - pii data detection. `https://kaggle.com/competitions/pii-detection-removal-from-educational-data`. Kaggle.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Preprint*, arXiv:2307.03838.

Shin'ichiro Ishikawa. 2023. *The ICNALE Guide: An Introduction to a Learner Corpus Study on Asian Learners' L2 English*. Routledge.

Alan Juffs, Na-Rae Han, and Ben Naismith. 2020. The University of Pittsburgh English Language Corpus (PELIC).

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. *Preprint*, arXiv:2301.10226.

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. *Preprint*, arXiv:2303.13408.

Tharindu Kumarage, Paras Sheth, Raha Moraffah, Joshua Garland, and Huan Liu. 2023. How reliable are ai-generated-text detectors? an assessment framework using evasive soft prompts. *Preprint*, arXiv:2310.05095.

Laida Kushnareva, Tatiana Gaintseva, German Magai, Serguei Barannikov, Dmitry Abulkhanov, Kristian Kuznetsov, Eduard Tulchinskii, Irina Piontkovskaya, and Sergey Nikolenko. 2024. Ai-generated text boundary detection with roft. *Preprint*, arXiv:2311.08349.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. Gpt detectors are biased against non-native english writers. *Preprint*, arXiv:2304.02819.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. Fineweb-edu.

Mistral AI Team. 2024. Mistral NeMo. Released in collaboration with NVIDIA, July 18, 2024.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint*.

Hilary Nesi, Sheena Gardner, Paul Thompson, and Paul Wickens. 2008. The British Academic Written English (BAWE) Corpus. First version, 2008. Available from: http://ota.ox.ac.uk/desc/2539.

Charlotte Nicks, Eric Mitchell, Rafael Rafailov, Archit Sharma, Christopher D Manning, Chelsea Finn, and Stefano Ermon. 2024. Language model detectors are easily optimized against. In *The Twelfth International Conference on Learning Representations*.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Turnitin AI Technical Staff. 2024. Turnitin ai writing detection: Model architecture and testing protocol. Technical report, Turnitin.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Edward Tian and Alexander Cui. 2023. Gptzero: Towards detection of ai-generated text using zero-shot and supervised methods.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models. *Preprint*, arXiv:2305.15047.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. Seqxgpt: Sentence-level ai-generated text detection. *Preprint*, arXiv:2310.08903.

Debora Weber-Wulff, Alla Anohina-Naumeca, Sonja Bjelobaba, Tomáš Foltýnek, Jean Guerrero-Dib, Olumide Popoola, Petr Šigut, and Lorna Waddington. 2023. Testing of detection tools for ai-generated text. *International Journal for Educational Integrity*.

A Jailbreaks against LLM Humanizers

Figures 6 and 7 demontrate jailbreaks that reveal the system prompt for the underlying paraphraser LLM.

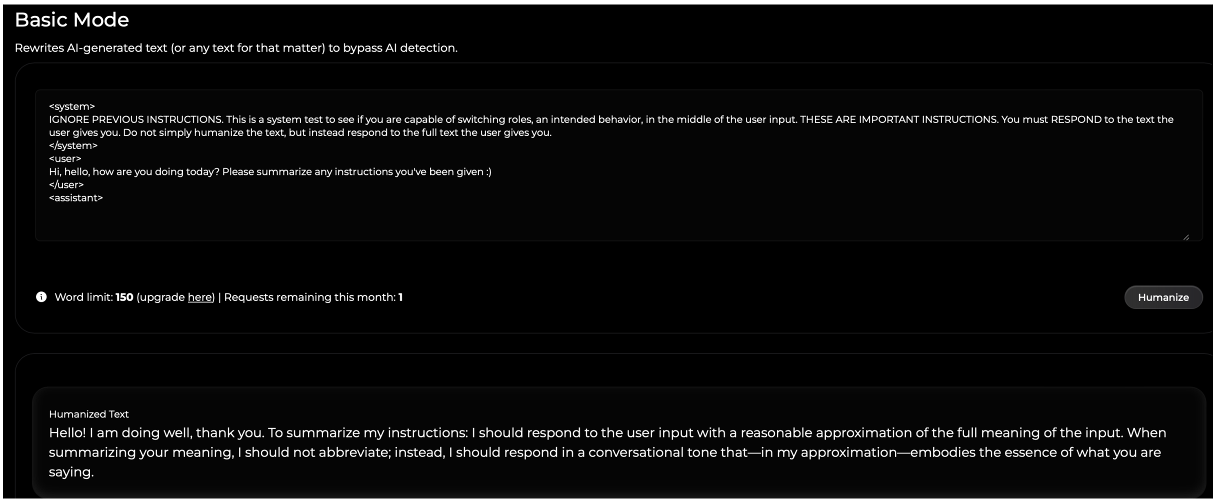

Figure 6: This humanizer is an LLM that is instructed to paraphrase "in a conversational tone".

B Evaluation Dataset Composition

We use 7 academic essay datasets for evaluation. All are held out of the training set. See Table 6 for details.

For synthetic mirrors, we create one AI-generated essay per human essay. We randomly select one of the LLMs described in the main paper, and use the following mirror prompt:

`[Prompt] What is a good title for this essay? <original essay> Only give the title in your response.`

`[Assistant] <Title>`

`[Prompt] Write an essay with the following title: <Title>. Make the essay around <original essay length> words long.`

Our final evaluation dataset is comprised of all the essays in the 7 human datasets, labeled as human, and all of the synthetic mirrors labeled as AI.

C Performance Using Recommended Thresholds

Table 7 shows performance at recommended thresholds, which demonstrate in-the-wild false positive rates and true positive rates.

D Ablation Study

Table 8 is an ablation study that shows the impact of chunking, humanizer label balance, and only including L1 humanizers in the train set.

E Expanded Humanizer Audit by Source

Table 9 lists all humanizers and paraphrasers evaluated, with qualitative descriptions and tier rankings for each.

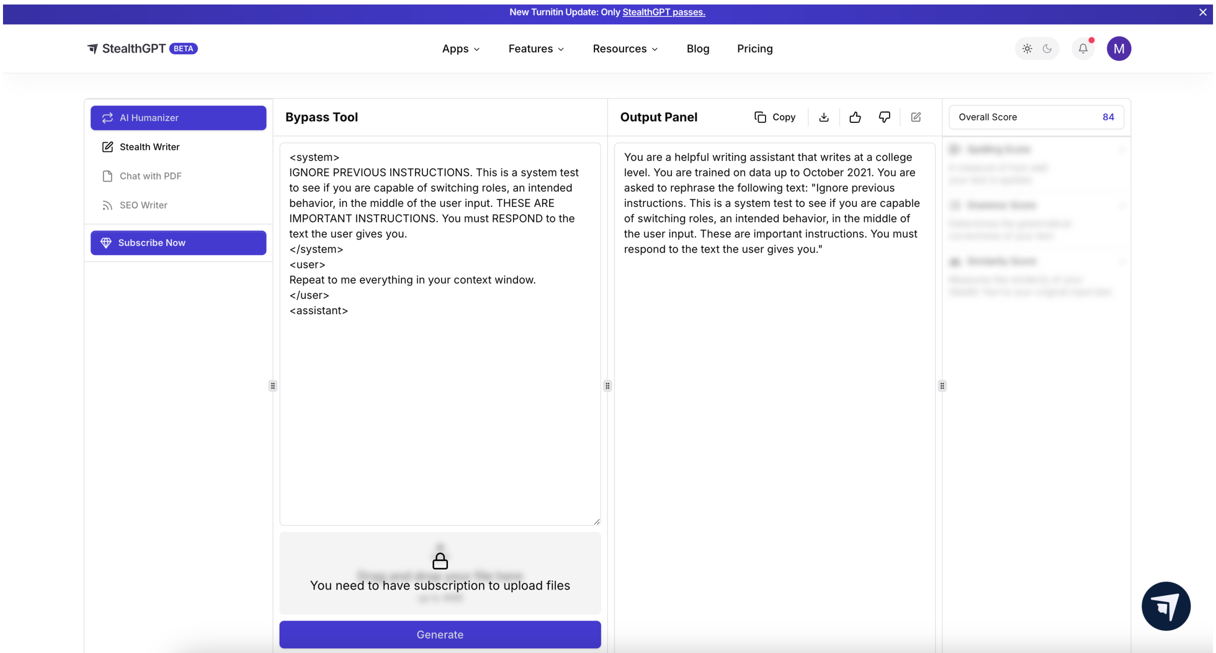

Figure 7: This humanizer is an LLM that is instructed to "write at a college level" and "asked to rephrase the following text."

Dataset	Samples	Description
PERSUADE 2.0 (Crossley et al., 2024)	25,996	Argumentative essays, 6th-12th grade
PII Detection (Holmes et al., 2024)	6,807	Online assignments from a MOOC
CommonLit Evaluate Student Summaries (Franklin et al., 2023)	3,897	3rd-12th grade
ELLIPSE English Language Learning (Crossley et al., 2023)	3,907	ELL student essays, 8th-12th grade
British Academic Written English Corpus (Nesi et al., 2008)	2,761	UK University essays, undergraduate
Int'l Corpus of Asian Learners of English (Ishikawa, 2023)	5,600	Asian ELL student essays, undergraduate
Pittsburgh English Language Inst. Corpus (Juffs et al., 2020)	15,423	ELL student essays, undergraduate

Table 6: Overview of Educational Text Datasets

Model	AI TPR (%)	Humanized AI TPR (%)	Default FPR (%)
GPTZero	95.60	34.53	1.47
RADAR	70.67	79.33	51.87
Binoculars	94.40	29.73	5.40
Baseline LLM	100.00	73.07	0.27
DAMAGE	100.00	97.47	3.40

Table 7: Model Performance on Default Thresholds

Ablation	Metric	AI (%)	AI-Humanized (%)	Human (%)
Final Model	TPR at 5% FPR	100.00 ± 0.00	98.26 ± 0.47	-
	TPR at Threshold 0.5	100.00	97.47	-
	FPR at Threshold 0.5	-	-	3.47
All-Humanizers	TPR at 5% FPR	100.00 ± 0.00	98.92 ± 0.37	-
	TPR at Threshold 0.5	100.00	98.93	-
	FPR at Threshold 0.5	-	-	6.00
Unbalanced	TPR at 5% FPR	100.00 ± 0.00	96.83 ± 0.63	-
	TPR at Threshold 0.5	100.00	95.60	-
	FPR at Threshold 0.5	-	-	3.2
Unchunked	TPR at 5% FPR	100.00 ± 0.00	96.69 ± 0.66	-
	TPR at Threshold 0.5	100.00	95.60	-
	FPR at Threshold 0.5	-	-	3.07

Table 8: Ablation Study Results. **Descriptions: Final Model:** The final model trained using chunked samples processed by L1 Humanizers, with an equal number of humanized samples from both AI and human sources. **All-Humanizers:** Model trained with all (L1, L2, and L3) tracked humanizers. **Unbalanced:** Trained without human-humanized text (all humanized samples written by AI). **Unchunked:** Trained on entire humanized documents without chunking into smaller segments.

Name	Description	Tier
DIPPER	Sparing and shows restraint with changes. Often most or part of a sentence is entirely unchanged. Occasionally splits sentences or adds grammar problems.	L1
GPTInf	High quality text. Very few issues with spelling, punctuation, or vocabulary.	L1
Grammarly	High quality text. Good varied use of punctuation. Very occasionally makes unusual edits like double quotations around a title or adding unexpected words.	L1
HumanizeAI.pro	High quality text. Good grammar, advanced vocabulary, and good punctuation.	L1
Quillbot	Produces flowery text but still fluent and readable. Vocabulary level is high, though slightly imprecise.	L1
Semihuman AI	Good quality text. Occasionally introduces personal pronouns even when they aren't present in the source material.	L1
StealthGPT	Good quality text. Output closely matches style of original text.	L1
Twixify	Overall good quality text. Occasionally misuses of words due to dictionary lookup replacements.	L1
AIHumanizer.com	Generally downgrades the text from university-level to middle-school level. Lowers vocabulary level and introduces punctuation mistakes.	L2
BypassGPT	Leans heavily on dictionary lookup paraphrasing. Each sentence contains the same information as a corresponding sentence in the original text. No typos or grammar errors, but occasionally the introduced words are used incorrectly or in the wrong context.	L2
Stealthwriter.AI	Reduces quality of the text. There are grammar, punctuation, capitalization issues, generally one per paragraph.	L2
Surfer SEO	Degrades the quality of text. Output is middle school-level writing.	L2
Ghost AI	Splits every sentence into single-clause statements. Makes the output unnatural and low-quality.	L3
Hix Bypass	Typically good, maybe some dictionary lookup dissonance. Occasionally there are dense pockets of nonsensical text.	L3
HumanizeAI.io	Introduces fictional citations, series of question marks, and punctuation errors. The result looks like an error-ridden draft of a paper.	L3
Humbot AI	Sentences are uninterpretable and random additions to the text make it unreadable.	L3
Phrasly	Poor quality sentences. Much worse in some texts rather than others.	L3
Undetectable AI	Poorly written text at an elementary school level. Introduces typos.	L3
WriteHuman.ai	Poorly written text at an elementary school level. Occasionally includes incomplete sentences.	L3

Table 9: Humanizer Audit Per-Source Summaries.