SENTRA: Selected-Next-Token Transformer for LLM Text Detection

Anonymous ACL submission

Abstract

LLMs are becoming increasingly capable and widespread. Consequently, the potential and reality of their misuse is also growing. In this work, we address the problem of detecting LLM-generated text that is not explicitly declared as such. We present a novel, general-purpose, and supervised LLM text detector, *SElected-Next-Token tRAnsformer (SENTRA)*. SENTRA is a Transformer-based encoder leveraging selected-next-token-probability sequences and utilizing contrastive pre-training on large amounts of unlabeled data. Our experiments on three popular public datasets across 24 domains of text demonstrate SENTRA is a general-purpose classifier that significantly outperforms popular baselines in the out-of-domain setting.

1 Introduction

The problem of determining whether a text has been generated by an LLM or written by a human has been widely studied in both academia (Tang et al., 2024) and industry. Several commercial-level automated text detection systems have been developed, including GPTZero (Tian and Cui, 2023), Originality (Originality.AI, 2025), Sapling (Sapling AI, 2025), and Reality Defender (Reality Defender, 2025). Although significant progress has been made in detecting LLM-generated text over the past several years, these systems remain far from perfect and are often unreliable. A major limitation is their brittleness: they can perform well on certain types of LLM-generated text but fail catastrophically in other cases (Dugan et al., 2024). This issue is particularly pronounced when operating in a real world scenario, where models must handle out-of-domain (OOD) data, different LLM generators, or various LLM "attacks" (Dugan et al., 2024; Zhou et al., 2024). Therefore, it is crucial to develop more generalizable methods that deliver reliable performance across these settings.

The probability of a document under and LLM's model can be measured by auto-regressively feeding the document's tokens into the LLM and observing the predicted probabilities for each token. This process produces a sequence of values called selected-next-token-probabilities (SNTP) that has been extensively used in prior work on LLM-generated text detection (Guo et al., 2023; Hans et al., 2024; Verma et al., 2024). These prior works primarily rely on either heuristics (handcrafted functions) applied to SNTP sequences or linear models trained on expert-derived features (Hans et al., 2024; Verma et al., 2024). In contrast, our proposed approach encodes SNTP sequences using a Transformer model pre-trained on unlabeled data, leveraging the expressivity of Transformers to directly learn a representation of the probability that a single or multiple LLMs assign to tokens in a document. More specifically, we propose *SElected-Next-Token tRAnsformer (SENTRA)*, a method for detecting LLM-generated text that directly learns a detection function in a supervised manner from SNTP sequences. This method utilizes a novel Transformer-based architecture with a contrastive pre-training mechanism. The learned representation can be fine-tuned on labeled data to create a supervised model that distinguishes LLM-generated texts from human-written texts.

For the LLM-text-detection task, supervised detectors have been shown to generalize poorly outside the training distribution (Dugan et al., 2024). Prior supervised methods typically leverage raw tokens as input and tend to overfit to token selections in a document. Heuristic or linear models on SNTP input have been shown to generalize well, but these simple models lack the expressivity to fully exploit the information in the SNTP sequences. Our SENTRA network addresses this issue by learning generalizable functions on SNTP. We show empirically that the supervised method presented in this paper generalizes to unseen domains better than both supervised and unsupervised baselines by leveraging our proposed Transformer-based architecture, thus demonstrating greater generalization to distribution shifts.

In this paper, we demonstrate the following:

- Detectors utilizing SENTRA as their encoder *generalize* well to domains outside of the training distribution(s).
- Contrastive pre-training of SENTRA leads to *improved generalization* results on new domains.
- SENTRA outperforms all studied baselines in out-of-domain evaluations on three widely used benchmark datasets.

Because of the number of possible domains, improving out-of-domain generalization is the most important task to achieve LLM generated text detection in the wild.

2 Related Work

With the rise of LLMs, significant research has been conducted on accurately detecting text generated by these models (Tang et al., 2024). At a high level, these detectors can be categorized into three approaches: watermarking, unsupervised (or zero-shot) detection, and supervised detection. Watermarking generally relies on the LLM deliberately embedding identifiable traces in its output (Liu et al., 2025). In this work, we focus on the general detection problem, including cases involving non-cooperative LLMs; therefore, we do not consider watermarking as a point of comparison. Unsupervised methods typically leverage metrics computed by an LLM on the target document. These methods can be further divided into white-box detection, where the candidate LLM is known (Mitchell et al., 2023), and black-box detection, where the candidate LLM is unknown (Tang et al., 2024). Given our focus on the general detection problem, we prioritize black-box detection methods. Supervised methods, on the other hand, involve collecting a corpus of human-written and LLM-generated text samples, which are then used to train the detection models (Verma et al., 2024; Soto et al., 2024).

Selected-next-token-probabilities (SNTP) have been widely used for LLM detection in both white and black box settings (Guo et al., 2023; Hans et al., 2024; Verma et al., 2024). Perplexity (Jelinek et al., 1977) is a commonly used metric to evaluate an LLM's ability to model a given dataset. In the context of AI detection, a lower perplexity score on a document indicates an LLM "fits" a document and this indicates a higher likelihood the document was LLM-generated. Conversely, a higher perplexity score suggests the LLM's probability model does not fit or accurately represent the candidate text, implying a lower likelihood that the text was generated by the LLM (Guo et al., 2023).

Some detectors use multiple sequences of SNTP for the detection task (Verma et al., 2024; Hans et al., 2024). Verma et al. (2024) leveraged SNTPs from two Markov models, along with an LLM's SNTP, extracted features, and a forward feature selection scheme as inputs to a linear classifier. In contrast to Guo et al. (2023), Hans et al. (2024) argued that relying solely on the perplexity score for LLM-generated content detection can be misleading. Although human-authored text generally results in higher perplexity, prompts can significantly influence perplexity values. The authors highlighted the "capybara problem", where the absence of a prompt can cause an LLM-generated response to have higher perplexity, leading to false detections. They addressed this issue by introducing *cross-perplexity* as a normalizing factor to calibrate for prompts that yield high perplexity.

DetectGPT is an unsupervised method based on the idea that texts generated by LLMs tend to "occupy negative curvature regions of the model's log probability function" (Mitchell et al., 2023). The method generates perturbations of the sample text using a smaller model and compares the log probability of the sample text to that of the perturbations. Fast-DetectGPT replaces the perturbations in DetectGPT with a more efficient sampling step (Bao et al., 2024). Nguyen-Son et al. (2024) observed that the similarity between a sample and its counterpart generation is notably higher than the similarity between the counterpart and another independent regeneration. They demonstrated that this difference in similarity is useful for detection.

The most common supervised baseline for LLM-generated text detection is a RoBERTa classifier (Liu et al., 2019) trained on a corpus of labeled text, where each document is marked as either human-written or LLM-generated. Several studies have expanded on this approach to supervised text-based classification. Yu et al. (2024) trained a feed-forward classifier with two hidden layers using intrinsic features derived from Transformer hidden states, determined via KL-divergence. Tian et al. (2024) address the challenge of detecting short texts by treating short samples in the training

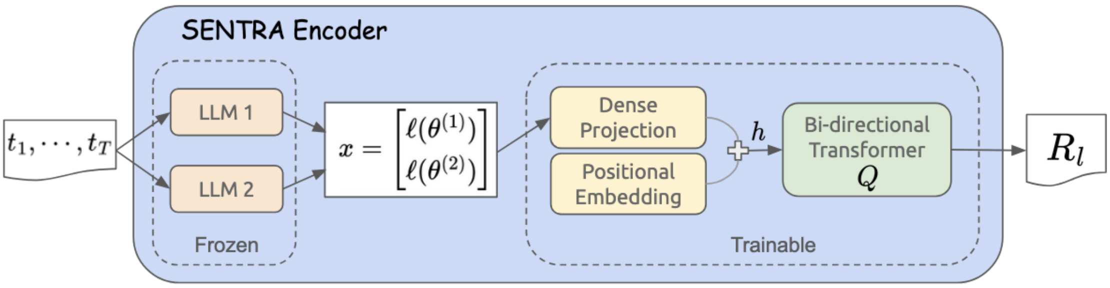

Figure 1: SENTRA leverages the selected-next-token-probabilities from two frozen LLMs. These two sequences of logits are concatenated into a vector. Each of these vectors are projected to the dimension of the bi-directional Transformer.

corpus as partially "unlabeled". Hu et al. (2023) employed adversarial learning to improve the robustness of their RoBERTa-based classifier against paraphrase attacks.

Several publications have explored contrastive training for the LLM detection task (Bhattacharjee et al., 2023, 2024; Soto et al., 2024; Guo et al., 2024). These studies use contrastive pre-training for a text Transformer, which is chosen to be RoBERTa (Liu et al., 2019) in many cases, to guide the network toward a representation more useful for LLM-generated text detection. Furthermore, many prior contrastive training strategies focus on identifying stylometric features (Soto et al., 2024; Guo et al., 2024), while other studies extract stylometric features directly and train classifiers using those features (Kumarage et al., 2023a). Rather than focusing on text representations, our method is mainly designed to produce useful SNTP representations and, thus, proposes a different contrastive pre-training scheme that compares textual representations with those of the SNTP Transformer.

However, SNTP and supervised methods have been shown, both intuitively and empirically, to struggle with generalization to unseen domains (Li et al., 2024; Roussinov et al., 2025). For instance, Lai et al. (2024) applied adaptive ensemble algorithms to enhance model performance in OOD scenario. Meanwhile, Guo et al. (2024) and Soto et al. (2024), recognizing the limited number of widely adopted general-purpose AI assistants, proposed to train an embedding model to learn the writing style of LLMs, and thereby improving the detection accuracy.

Prior work has shown SNTP to be an effective input for identifying LLM generated text (Guo et al., 2023; Hans et al., 2024; Verma et al., 2024), but they rely on relatively simple metrics or heuristics. In this paper, we propose a Transformer-based SENTRA model that learns a representation of SNTP sequences used for more effective training of detection models that better generalize to unseen domains.

3 Methodology

3.1 Overview of the SENTRA Method

Consider a document t consisting of an input sequence of T tokens $t = (t_1, t_2, \cdots, t_T)$. Assuming an LLM has parameters θ, the probability of document t given this LLM can be specified as

$$P(t_1, t_2, \cdots, t_T|\theta) = \prod_{t=1}^{T} q_i(\theta) \tag{1}$$

where

$$q_i(\theta) = P(t_i \mid t_1, t_2, \cdots, t_{i-1}; \theta) \tag{2}$$

is the probability of token t_i given the preceding tokens $(t_1, t_2, \cdots, t_{i-1})$. We denote the observed sequence of selected-next-token-probabilities (SNTP) as

$$q(\theta) = (q_1(\theta), q_2(\theta), \cdots, q_T(\theta))\,. \tag{3}$$

It is common, and done in this work, to use the log representation of these sequences

$$\ell_i(\theta) = -\log q_i(\theta) \tag{4}$$

where ℓ is the log of the SNTP sequences.

Prior work, reviewed in Section 2, has proposed various heuristic functions on these sequences that are useful in detecting LLM-generated text (Guo et al., 2023; Hans et al., 2024). SENTRA replaces these heuristic functions on SNTP sequence(s) with a neural network, as shown in Figure 1 illustrating

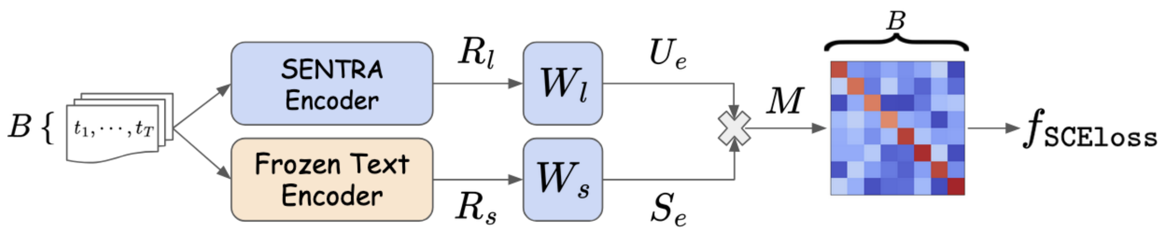

Figure 2: Pre-training: the outputs of SENTRA and a frozen text encoder go through linear layers, (W_s and W_l) respectively, and normalization before a matrix multiplication (`matmul`) operation to produce the similarity matrix M. Blue and orange blocks indicate trainable and frozen components respectively.

our proposed method. In particular, we leverage k LLMs, each with parameters $\theta^{(k)}$ to produce SNTP sequences $\ell^{(k)}$ and for a candidate document with T tokens using process in Equation 2. The k sequences are concatenated to form input sequence x. Note that in Figure 1, $k = 2$.

Instead of token embeddings often seen in Transformer architectures (Devlin et al., 2019), each token-indexed representation $x_t \in x$ is independently projected using a fully connected layer.

$$h_t = f(Wx_t + b) + Z_t \tag{5}$$

where h is the dense embedding representation, f is the ReLU activation function, W is the weight matrix, b is the bias, and Z_t are $Z \in \mathbb{R}^{T\times D}$ learned positional embeddings. This transformation results in a representation of size $T \times D$ for a single document. Note a learned [CLS] representation $h_{[CLS]} \in \mathbb{R}^D$ is pre-pended to the sequence before the positional embeddings are applied. This representation h_t is passed through a bi-directional Transformer (Devlin et al., 2019) Q, as shown in Figure 1.

The output of SENTRA is a learned representation over SNTP, capturing the probability assigned by two LLMs to the tokens in a document. For classification, we use the representation at the `[CLS]` token and append a classification head. This Transformer produces our SENTRA representation R_l over SNTP sequences.

$$R_l = Q(h) \tag{6}$$

where R_l is a D dimensional representation of the document over the token length T.

In summary, SENTRA is the first Transformer-based encoder to systematically learn a useful representation of SNTP sequences. Similar to many Transformer-based approaches (Devlin et al., 2019; Radford et al., 2021), that have traditionally used different modalities of input information, we demonstrate in Section 3.2 that our method can leverage large quantities of unlabeled data to enhance this learned representation.

3.2 SENTRA Contrastive Pre-Training

We further explore the pre-training of SENTRA using unlabeled text data and demonstrate in Section 4.4 that it significantly improves SENTRA's performance. Notably, this pre-training scheme is reminiscent of CLIP (Radford et al., 2021). Figure 2 illustrates our concept for pre-training SENTRA. We leverage off-the-shelf, pre-trained text representations to help SENTRA learn a useful representation of SNTP sequences. A document is encoded using both a pre-trained text encoder (Devlin et al., 2019; Liu et al., 2019) and our SENTRA network, producing representations R_l and R_s. These representations are projected to a joint embedding space, U_e and S_e, using fully connected layers C_l and C_s for the text and SNTP representations respectively.

$$\begin{aligned} U_e &= C_l(R_l) \\ S_e &= C_s(R_s) \end{aligned} \tag{7}$$

After applying L2 normalization to U_e and S_e to control for scaling, we then compute a comparison matrix M

$$M = (U_e S_e^T)e^r \tag{8}$$

where r is learned temperature scalar.

The two encoders learn to match representations of the same document within a batch B. Employing the contrastive learning objective, $\mathcal{L} = \frac{\mathcal{L}_s+\mathcal{L}_l}{2}$

$$\mathcal{L}_l = -\frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{\exp(M_{ii})}{\sum_{j=1}^{n}\exp(M_{ij})}\right) \tag{9}$$

$$\mathcal{L}_s = -\frac{1}{n}\sum_{j=1}^{n}\log\left(\frac{\exp(M_{jj})}{\sum_{i=1}^{n}\exp(M_{ij})}\right) \tag{10}$$

we then minimize the cross-entropy loss over the columns (text-to-SNTP), and rows (SNTP-to-text) of the comparison matrix M, using the ground truth text-SNTP pairings in the batch, $y = 0, 1, ...B - 1$.

The pre-training scheme effectively enables SENTRA to produce representations that align with those generated by the frozen text encoder, thereby yielding more useful representations of the $\ell^{k=1}$ and $\ell^{k=2}$ sequences.

In (Radford et al., 2021)'s work, the authors jointly trained text and image encoders from scratch. Unlike CLIP, which deals with text and images, we focus solely on text and on pre-training only the SENTRA SNTP encoder. To do this, we freeze a pre-trained text encoder and train only SENTRA and the contrastive embedding projections.

3.3 Implementation

We implement our SENTRA model with eight attention heads, eight layers, and a hidden dimension of 768 for a total of 57M parameters. The Transformer architecture and positional embeddings follow the same definitions as in BERT (Devlin et al., 2019). Before pre-training, the SENTRA parameters are randomly initialized. The frozen text encoder used for contrastive pre-training is initialized from RoBERTa (Liu et al., 2019). SENTRA is pre-trained on a 600K sample of Common Crawl data from RedPajama (Weber et al., 2024). Pre-training is conducted for 20 epochs with a batch size of 256 and a maximum token length of 64. We then continue contrastive training for 10 epochs with a batch size of 128 and a maximum token length of 512 to pre-train the later position embeddings. The peak learning rate was set to $1\mathrm{e}-4$ for both phases. We use the AdamW (Loshchilov and Hutter, 2019) optimizer with a weight decay of $1\mathrm{e}-2$ and set the contrastive learning temperature to 0.007 (Chen et al., 2020). During fine-tuning, we initialize SENTRA from the pre-trained model, use a learning rate of $1\mathrm{e}-4$, a weight decay of $1\mathrm{e}-2$, and apply early stopping with a patience of two epochs on a validation dataset.

As shown in Figure 1, we implemented SENTRA with two SNTP sequences and therefore $k = 2$. Following Binoculars (Hans et al., 2024), we use Falcon-7B and Falcon-7B-Instruct (Almazrouei et al., 2023) to produce these sequences. We used a sequence of two SNTP because Binoculars showed it is useful for the detector to compare both SNTP, and we used the Falcon models specifically because Binoculars showed they worked well (Hans et al., 2024). During SENTRA training, the SNTP sequences are precomputed and cached. At inference, the computational complexity is dominated by the Falcon models. Because we use the same LLMs as Binoculars (Hans et al., 2024) and our SENTRA encoder is small, our method has the same order of complexity as Binoculars. See Appendix B for additional details.

We will release our SENTRA implementation and pre-trained model.

4 Experiments

4.1 Datasets

If we define text similar to the training data distribution as in-domain and text that is dissimilar as out-of-domain, it is well established supervised LLM detection methods perform significantly better in-domain than out-of-domain (Dugan et al., 2024). However, a model designed for LLM-generated text detection in real world scenarios will inevitably encounter out-of-domain texts. For this reason, this work focuses on *out-of-domain experiments*, where key subsets of data are withheld from the training dataset.

To evaluate the effectiveness of our proposed method, we used three publicly available datasets: RAID (Dugan et al., 2024), M4GT (Wang et al., 2024a) and MAGE (Li et al., 2024), focusing exclusively on English-language data.

RAID: The full RAID dataset contains over 6 million human- and LLM-generated texts spanning 8 domains, 11 LLM models, multiple decoding strategies, penalties, and 11 adversarial attack types. We down-sampled it to 500K instances before performing out-of-domain split sampling. With the included attacks, the RAID dataset also assesses the effectiveness of different supervised baseline methods against adversarial attacks under the in-attack setup.

M4GT: An extension of M4 (Wang et al., 2024b), the M4GT dataset is a multi-domain and multi-LLM-generator corpus comprising data from 6 domains, 9 LLMs, and 3 different detection tasks.

MAGE: The MAGE dataset covers 10 content domains, with data generated by 27 LLMs using 3 different prompting strategies. It is specifically designed to assess out-of-distribution generalization capability. We use the "Unseen Domains" evaluation from (Li et al., 2024).

Each dataset is further split into training, validation and test sets. For MAGE, we used the published split. To mitigate the label imbalance problem, the train and validation splits are balance-sampled to ensure an equal number of human- and LLM-generated texts. This was achieved by down-sampling the majority class to match the size of the minority class within split. Addressing this imbalance is crucial for two reasons: 1) the percentage of LLM-generated text is over 97% in the RAID dataset by design; 2) across the three datasets, the proportion of LLM-generated text varies significantly. The average train and validation set sizes show how much data went into the training of the supervised methods while ensuring class balance, providing a clear comparison to the total dataset size. The MAGE dataset has significantly shorter texts and this adds difficulties to the detection task (Tian et al., 2024; Fraser et al., 2024).

Beyond out-of-domain evalution, we further assessed our method in an out-of-LLM (OOLLM) setup using MAGE's out-of-LLM testbed which contains 7 LLM splits. Table 4 contains detailed statistics on the evaluation datasets. For fair comparison across methods, we use the first 512 tokens from each document in all datasets.

4.2 Baseline Methods

We evaluated and compared the performance of our approach against multiple existing methods, including zero-shot, embedding-based, and supervised detectors. For zero-shot, we selected **perplexity** (Guo et al., 2023), **Fast-DetectGPT** (Bao et al., 2024), and **Binoculars** (Hans et al., 2024) detectors. For embedding-based detectors, we selected **UAR** (Soto et al., 2024) and evaluated both its Multi-LLM and Multi-domain models. For supervised detectors, we chose **RoBERTa** (Liu et al., 2019) with direct fine-tuning, **Ghostbuster** (Verma et al., 2024) which trains a logistic regression classifier on forward-selected crafted log-probability features, and **Text Fluoroscopy** (Yu et al., 2024) which utilizes intrinsic features. For RoBERTa, we used the same settings as the fine-tuning of SENTRA: a learning rate of $1\mathrm{e}-4$, a weight decay of $1\mathrm{e}-2$, and a patience of two epochs.

We used Falcon-7B and Falcon-7B-Instruct across all baseline methods that required LLMs, except for Fast-DetectGPT where we followed its black-box setting. Appendix C provides a detailed description of the setup, assumptions and modifications made for each baseline method.

We compared the baseline methods mentioned above with our proposed methods. We present results from two SENTRA encoder variations, R-SENTRA and SENTRA. R-SENTRA has all non-LLM weights in SENTRA encoder initialized at random (without pre-training), whereas the full SENTRA model has those weights pre-trained as described in Section 3.3.

Interestingly, prompting an LLM to do the LLM-text detection task is not well studied and does not appear in standard benchmarking work (Dugan et al., 2024; Wang et al., 2024b; Li et al., 2024). In the appendix D, we performed a small case study to evaluate how a SOTA LLM, GPT4-o (OpenAI et al., 2024a), and a reasoning model, o1 (OpenAI et al., 2024b), could perform on a sample of the OOD datasets. We were unable, due to the high cost of these APIs, to run the full evaluation datasets through these models and therefore chose to randomly sample from the full datasets and perform a fair comparison on the smaller test sets. The evaluation results for the GPT4-o and o1 LLMs and their comparison with SENTRA performance are reported in Appendix D. As Table 10 shows, GPT-4o shows little detection skill. o1 was able to detect reasonably well on some datasets, but not as well as SENTRA. This case study shows a full and robust evaluation of LLM performance is needed for the task of LLM-text detection, including using full instead of sampled datasets, exploring alternative prompting strategies, and other comprehensive experimental settings. Since this analysis is beyond the scope and budget of this work, we defer it as the topic of future research.

4.3 Ablation Study

Table 1 shows the effect of pre-training SENTRA on all datasets. r-SENTRA is the "raw" SENTRA showing the architecture's performance without pre-training on the M4GT dataset. Across the four datasets, the average and worst-case performance over the domains was increased after pre-training. This shows the contrastive pre-training method presented in Figure 2 is an effective method for improving SENTRA as an encoder for the LLM text detection.

Table 2 presents an ablation study on SENTRA components. Rows 2 and 3 of Table 2 show the AUROC performance metric after removing each of the two LLMs used to create SENTRA's SNTP input (see Figure 1). Rows 4 and 5 of the table show the results when the Falcon-7b models (Almazrouei et al., 2023) are replaced by different

	RAID-OOD		M4GT-OOD		MAGE-OOD		MAGE-OOLLM	
	Avg	W	Avg	W	Avg	W	Avg	W
r-SENTRA	90.9	85.5	92.8	83.9	93.8	84.6	93.5	**89.9**
SENTRA	**92.5**	**87.0**	**93.0**	**87.1**	**94.2**	**86.0**	**93.6**	88.0

Table 1: Effect of Pre-training on SENTRA performance. Results are the average (Avg) and worst (W) AUROC across the domains in the evaluation.

	Avg	W
r-SENTRA	**92.8**	**83.9**
– Base LLM	89.4	81.8
– Instruct LLM	88.1	74.1
– Falcon + Qwen-2.5-3b	89.3	75.0
– Falcon + Gemma-3-1b	91.2	82.7

Table 2: Ablation Study. Results show the average (Avg) and worst (W) domain AUROC on the M4GT dataset. The top section, r-SENTRA, is our method without pre-training. The second section shows the effect of dropping each of the two frozen LLMs. The last section shows the effect of swapping the Falcon-7b models for different pairs of LLMs.

pairs of LLMs: Qwen-2.5-3b (Qwen et al., 2025) and Gemma3-1b (Team et al., 2025). From the results, we can see that Gemma3-1b (Team et al., 2025) is competitive with Falcon-7b, and could be an alternative for more compute constrained environments. These choices in LLMs are by no means an exhaustive search, and this ablation shows SENTRA can work with other LLM pairs while echoing Binocular's result showing Falcon-7b is particularly effective (Hans et al., 2024).

4.4 Results

We measure performance of all the methods described in Section 4.2 on three out-of-domain and one out-of-LLM evaluation, and the average and worst-case AUROC results are presented in Table 3. For the supervised methods, these evaluations assess how well the LLM text detectors perform in real world scenarios, where data distributions differ from the training distribution. Detectors that remain more invariant across these evaluations are considered more robust to changes and variations in data, thus showing better generalization to unseen domains and generators.

Methods that are not zero-shot or linear models are inherently more stochastic; therefore, the UAR, RoBERTa, and SENTRA methods were ran over three random seeds. The main results in Table 3 show the mean over these seeds. Mean and standard deviation over the seeds across all domains and evaluations are shown in Appendix A. On each evaluation, our performance metric is the mean or minimum over the domains. For each method, this requires training a separate model for each random seed, each domain, and each evaluation. Because of the combinations of methods, seeds, domains, and datasets, each additional run becomes very expensive, and therefore, we were limited to three runs on each evaluation.

Table 3 presents performance of different baselines measured by AUROC across different OOD test data for the RAID, M4GT and MAGE datasets (columns RAID-OOD, M4GT-OOD and MAGE-OOD in Table 3 respectively) and for the OOLLM test data for the MAGE dataset (column MAGE-OOLLM in the table). The top section of Table 3 shows the performance of label-dependent methods while the second section shows the performance of heuristic methods.

Table 3 shows that SENTRA outperformed all the baselines on average and in the worst case across the three OOD and one OOLLM evaluations. SENTRA achieved average AUROC performance improvements of 1.8%, 5.4% and 6.7% for RAID (Dugan et al., 2024), M4GT (Wang et al., 2024a) and MAGE (Li et al., 2024) out-of-domain datasets respectively, as compared to the second-best performing baseline. For the OOLLM evalution, SENTRA showed a 7.5% increase over the next best baseline. These results show SENTRA serves as a generalizable encoder for LLM detection models when one considers likely OOD or OOLLM distribution shifts. These results show, in the likely event your detector encounters a domain outside the training distribution, we expect SENTRA to have the best expected performance and best worst-case performance on those unseen domains.

Since LLMs became increasingly available and their usage has surged, interest in detection tools, such as those presented in this paper, has grown

	RAID-OOD		M4GT-OOD		MAGE-OOD		MAGE-OOLLM	
	Avg	W	Avg	W	Avg	W	Avg	W
RoBERTa [20]	90.9	84.4	88.2	82.8	88.3	74.4	87.1	69.9
Text-Fluoroscopy [42]	76.4	70.6	83.2	78.1	63.9	47.8	41.5	28.3
UAR-D [32]	81.7	71.4	75.3	63.9	63.4	40.5	71.7	65.8
UAR-L [32]	87.3	76.3	84.7	71.0	76.4	61.2	80.4	70.7
Ghostbuster [37]	84.7	74.1	87.8	73.3	79.2	65.0	68.5	34.3
PPL [9]	72.9	69.4	87.0	81.7	57.2	45.7	59.0	25.4
Binoculars [11]	82.0	79.4	89.1	79.0	61.7	52.9	61.8	14.7
Fast-DetectGPT [2]	78.6	75.6	87.4	79.1	63.0	54.9	37.9	2.8
SENTRA	**92.5**	**87.0**	**93.0**	**87.1**	**94.2**	**86.0**	**93.6**	**88.0**

Table 3: Average (Avg) and worst (W) out-of-domain AUROC across the domains or LLMs. Methods in the top section are supervised while the methods in the second section are unsupervised. SENTRA is our method with pre-training. Results for non-deterministic methods are averaged over three random seeds.

(Wu et al., 2023). At the same time, countermeasures have emerged to attack these LLM text detectors, typically by altering LLM-generated text to elicit false negatives (Koike et al., 2024). Dugan et al. (2024) demonstrated many attacks can significantly degrade detector performance. In that study, the best open-source tool, Binoculars (Hans et al., 2024), exhibited much stronger performance on non-attacked data than on attacked data. For the unsupervised methods, (Guo et al., 2023; Hans et al., 2024; Bao et al., 2024), it is not immediately clear how to adapt the approach to a known attack. In contrast, for the supervised methods, the adaptation strategy is straightforward: train on attacked data. A model that is robust to a known attack, like the common paraphrase attack, should be able to detect LLM generated text even if that attack appears in a new domain. The RAID-OOD (Dugan et al., 2024) dataset demonstrates this situation where 11 attacks appear in the training and test sets. The results in Table 3 show SENTRA outperformed other methods when training and evaluating in the out-of-domain scenario where known attacks are included.

5 Conclusions

In this paper, we proposed a novel general purpose supervised LLM text detector method SENTRA that is a Transformer-based encoder leveraging SNTP sequences and utilizing contrastive pre-training on large amounts of unlabeled data. We show this supervised method acting on SNTP input outperforms previously considered heuristic functions and other methods that rely on text input. Since supervised detectors tend to perform better on data that is similar to their training distributions (Dugan et al., 2024), it is essential to include a wide variety of domains when testing such general-purpose detectors. Therefore, we tested the performance of SENTRA on three public datasets RAID, M4GT and MAGE containing a broad range of different domains (24 in total) across various experimental settings and compared its performance with eight popular baselines. We also evaluated SENTRA and the baselines on a out-of-LLM evaluation.

We empirically demonstrated that SENTRA significantly outperformed all baselines in our studied experimental settings. On our three evaluation datasets, SENTRA outperformed all eight popular baselines for the average and the worst-case OOD scenarios.

These results show that SENTRA is a strong method for training LLM text detectors that can generalize well to unseen domains and LLM generators. Our ablation study showed performance of SENTRA increases when two frozen LLMs are used instead of one frozen LLM. We also demonstrated our contrastive pre-training strategy increased the performance of SENTRA on all out-of-domain evaluations. Because SENTRA is better able to handle these critical out-of-domain and out-of-LLM settings, these results demonstrate SENTRA is a general-purpose encoder that can serve as a foundation for the LLM text detector models.

6 Limitations

In this work, we studied the effects of domain shifts on detection models. While these have significant impacts on detector performance, other factors can also influence results. Notably, prompt variation can have a large effect on detectors (Kumarage et al., 2023b). Many LLM detection benchmark datasets use prompt templates (Dugan et al., 2024) to generate their samples. However, these templates exhibit significantly less prompt variety than what a real-world detector is likely to encounter. Benchmark datasets with a broader range of prompting strategies are needed to further assess the robustness of detection methods.

We pre-trained our model on a relatively small sample of Common Crawl data. The volume of data and the amount of compute used for pre-training were small relative to what is typically used for foundation models (Liu et al., 2019; Radford et al., 2021). It is very likely SENTRA could be significantly improved with additional pre-training on larger datasets.

7 Ethical Considerations

In this study, we did not observe any detector achieving perfect performance on any slice of data. Therefore, any detector will inherently make trade-offs between false positives and false negatives when deployed in real-world scenarios, such as plagiarism detection. Users of LLM detection technology should be aware that these detectors are not perfect.

LLM Acknowledgement: We used ChatGPT for generating first iterations of some software snippets. We also consulted ChatGPT on the phrasing of some points in the paper and for catching some grammatical errors.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The Falcon Series of Open Language Models. *arXiv preprint*. ArXiv:2311.16867 [cs].

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. In *The Twelfth International Conference on Learning Representations*.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive domain adaptation for AI-generated text detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 598–610, Nusa Dua, Bali. Association for Computational Linguistics.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. Eagle: A domain generalization framework for ai-generated text detection. *arXiv preprint arXiv:2403.15690*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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. Association for Computational Linguistics.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

Kathleen C. Fraser, Hillary Dawkins, and Svetlana Kiritchenko. 2024. Detecting ai-generated text: Factors influencing detectability with current methods. *Preprint*, arXiv:2406.15583.

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

Xun Guo, Yongxin He, Shan Zhang, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. 2024. Detective: Detecting AI-generated text via multi-level contrastive learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in Neural Information Processing Systems*, 36:15077–15095.

Fred Jelinek, Robert L Mercer, Lalit R Bahl, and James K Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21258–21266.

Tharindu Kumarage, Joshua Garland, Amrita Bhattacharjee, Kirill Trapeznikov, Scott Ruston, and Huan Liu. 2023a. Stylometric Detection of AI-Generated Text in Twitter Timelines. *arXiv preprint*. ArXiv:2303.03697 [cs].

Tharindu Kumarage, Paras Sheth, Raha Moraffah, Joshua Garland, and Huan Liu. 2023b. How reliable are AI-generated-text detectors? an assessment framework using evasive soft prompts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1337–1349, Singapore. Association for Computational Linguistics.

Zhixin Lai, Xuesheng Zhang, and Suiyao Chen. 2024. Adaptive Ensembles of Fine-Tuned Transformers for LLM-Generated Text Detection. *arXiv preprint*. ArXiv:2403.13335 [cs].

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip Yu. 2025. A Survey of Text Watermarking in the Era of Large Language Models. *ACM Computing Surveys*, 57(2):1–36.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint*. ArXiv:1907.11692 [cs].

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Hoang-Quoc Nguyen-Son, Minh-Son Dao, and Koji Zettsu. 2024. SimLLM: Detecting Sentences Generated by Large Language Models Using Similarity between the Generation and its Re-generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22340–22352, Miami, Florida, USA. Association for Computational Linguistics.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, A. J. Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024a. GPT-4o System Card. *arXiv preprint*. ArXiv:2410.21276 [cs].

OpenAI, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil

Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. 2024b. OpenAI o1 System Card. *arXiv preprint*. ArXiv:2412.16720 [cs].

Originality.AI. 2025. Originality.ai - ai plagiarism and fact checker.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 Technical Report. *arXiv preprint*. ArXiv:2412.15115 [cs].

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Reality Defender. 2025. Reality defender.

Dmitri Roussinov, Serge Sharoff, and Nadezhda Puchnina. 2025. Controlling out-of-domain gaps in LLMs for genre classification and generated text detection. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3329–3344, Abu Dhabi, UAE. Association for Computational Linguistics.

Sapling AI. 2025. Ai detector.

Rafael Alberto Rivera Soto, Kailin Koch, Aleem Khan, Barry Y. Chen, Marcus Bishop, and Nicholas Andrews. 2024. Few-shot detection of machine-generated text using style representations. In *The Twelfth International Conference on Learning Representations*.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2024. The science of detecting llm-generated text. *Commun. ACM*, 67(4):50–59.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, C. J. Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju-yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. Gemma 3 Technical Report. *arXiv preprint*. ArXiv:2503.19786 [cs].

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

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, QINGHUA ZHANG, Ruifeng Li, Chao Xu, and Yunhe Wang. 2024. Multiscale positive-unlabeled detection of AI-generated texts. In *The Twelfth International Conference on Learning Representations*.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1702–1717, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. M4GT-bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3964–3992, Bangkok, Thailand. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024b. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. *NeurIPS Datasets and Benchmarks Track*.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F. Wong, and Lidia S. Chao. 2023. A Survey on LLM-generated Text Detection: Necessity, Methods, and Future Directions. *arXiv preprint*. ArXiv:2310.14724 [cs].

Xiao Yu, Kejiang Chen, Qi Yang, Weiming Zhang, and Nenghai Yu. 2024. Text fluoroscopy: Detecting LLM-generated text through intrinsic features. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15838–15846, Miami, Florida, USA. Association for Computational Linguistics.

Ying Zhou, Ben He, and Le Sun. 2024. Humanizing machine-generated content: Evading ai-text detection through adversarial attack. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation.*

A Additional Results and Experimental Notes

The datasets used in this work were used for research purposes. This aligns with their intended use and licenses. The details of the datasets are shown in Table 4.

Here we show the mean and standard deviation across three runs, (random seeds 42,43,44) for the methods that are not zero shot or logistic regression based. Note there were three M4GT and four RAID samples where Ghostbuster could not make an inference due to the low number of tokens in the document. For this documents, we infilled a low prediction score indicating human prediction. For the RAID dataset, we used the Binoculars for each document released by (Dugan et al., 2024).

B Computational Complexity

LLM generators are computationally expensive. Unfortunately, methods that rely on SNTP inputs depend on LLM inference, making it the most costly component of all detection methods studied in this work. However, SENTRA is a relatively small model with only eight Transformer layers, meaning that computational costs at inference are dominated by the production of SNTP inputs. During training, we cache the SNTP sequences so that the LLMs are run only once per sample. SENTRA uses the same LLMs as Binoculars (Hans et al.,

Dataset	Size	Domains	LLMs	Attks	A.Tokens	% LLM-Gen	A.Train	A.Val	A.Test
RAID-OOD	500,000	8	11	11	712	97.16%	22,398	2,488	62,500
M4GT-OOD	267,863	6	14	0	471	67.6%	97,584	10,893	33,482
MAGE-OOD	430,630	10	-	0	267	34.86%	167,972	50,387	5,682
MAGE-OOLLM	314,817	-	7	0	267	31.92%	186,636	47,988	8,022

Table 4: Overview of datasets used in the study. Attks is the number of attacks included in the dataset. A.Tokens is the average token length using the Falcon 1 tokenizer. A.Train, A.Val, and A.Test are the average train, validation, test set sizes across all domain splits. The train and validation datasets are class balanced. LLM stats for MAGE-OOD and domain stats for MAGE-OOLLM are not disclosed by the data authors.

	abstracts	books	news	poetry	recipes	reddit	reviews	wiki
RoBERTa	93.1±1.2	87.0±2.1	91.4±3.4*	95.2±1.3*	84.4±16.9	93.9±1.2*	90.2±2.3	91.8±2.8
Text-Fluor.	71.4±0.0	82.4±0.0	74.9±0.0	70.6±0.0	76.1±0.0	79.2±0.0	73.9±0.0	82.6±0.0
UAR-D	71.4±4.4	85.2±0.8	84.5±1.2	73.2±0.5	89.5±0.8*	82.4±0.3	84.9±1.1	82.3±0.2
UAR-L	89.6±2.0	91.1±0.2	89.8±0.4	76.3±2.6	85.3±1.2	88.8±0.7	88.1±0.4	89.3±0.5
PPL	69.7±0.0	76.8±0.0	69.4±0.0	73.9±0.0	69.6±0.0	76.6±0.0	75.8±0.0	71.3±0.0
Binoculars	83.2±0.0	84.3±0.0	80.2±0.0	83.5±0.0	79.4±0.0	83.2±0.0	82.1±0.0	80.2±0.0
Fast-DetectGPT	80.0±0.0	80.1±0.0	77.9±0.0	77.1±0.0	75.6±0.0	78.8±0.0	80.0±0.0	79.4±0.0
Ghostbuster	88.0±0.0	91.4±0.0	81.6±0.0	88.2±0.0	74.1±0.0	85.0±0.0	81.7±0.0	87.8±0.0
R-SENTRA	94.6±0.3	95.1±0.3*	88.4±0.5	92.5±2.2	85.5±0.9	91.7±0.1	87.8±0.5	91.8±0.3
SENTRA	95.1±0.1*	94.1±1.6	91.3±0.5	95.0±0.8	87.0±1.5	93.7±0.5	90.4±0.9*	93.2±0.7*

Table 5: Mean and standard deviation of the AUROC across random seeds on the RAID dataset.

2024), and because the cost of the SENTRA encoder is minimal compared to LLM inference, the overall computational complexity of SENTRA is roughly equivalent to that of the Binoculars method. Refer to Table 9 for detailed number of parameters.

Pre-training took approximately 36 hours on a GH200 GPU. We also fine-tuned RoBERTa and SENTRA models on GH200 instances. Fine-tuning for each data split too between .5 and 12 hours.

C Baseline Assumptions and Setups

This section details the assumptions and setups for all baseline methods if we have made modifications.

For UAR, the original paper compares the distance between the input query and the closest machine support query against the distance between the closest machine support query and the closest human support query. Mathematically speaking, given Q the input query, H the closest human support query, and $\mathbf{M}$ is the seeded machine support queries, the distance $d_{\mathrm{Q}} = \min_{m\in\mathbf{M}}[d(Q,m), d(H,m)]$ is used as the prediction. Though this allows d_Q to be directly usable for metric calculation, this is less trivial than a simple nearest neighbor classification where we calculate the percentage of machine support queries among k as the prediction. in our baseline, we employed the simple nearest neighbor approach with $k = 10$ and cosine similarity distance measure. For each domain, we randomly sampled 1,000 human and machine texts respectively to form the kNN seed corpus. We did not group texts into episodes and kept episode size of 1 due to the generally longer text lengths compared to twitter posts.

For Text Fluoroscopy, we switched the model from gte-Qwen1.5-7B-instruct to Falcon-7B-Instruct to better align with other baselines by eliminating the effect of model selection. With this change, we modified the input dimension to the feed forward network from 4096 to 4454 due to falcon models hidden state sizes. Despite the possibilities of under-training, we followed their implementation and sampled 160 data points for training, and 20 for validation (during training). The test set metric at the earliest highest validation accuracy was reported. We also optimized the feature selection script for more efficient batch processing.

For Ghostbuster, we included a minimum accuracy score improvement threshold of $1\mathrm{e}{-4}$ to avoid over-fitting and allow early stopping for MAGE dataset where we observed significantly more feature selection iterations compared to the other two datasets. In the case of least square convergence failure (`max_iter=1000`) in Logistic Regression fitting, the current feature list is taken as the best features for evaluation.

	arxiv	outfox	peerread	reddit	wikihow	wikipedia
RoBERTa	97.8±0.3*	84.9±2.2	82.8±18.6	89.6±3.9	85.5±2.3	88.5±3.9
Text-Fluor.	84.7±0.0	84.8±0.0	89.2±0.0	83.9±0.0	78.1±0.0	78.3±0.0
UAR-D	73.3±6.7	83.9±0.2	65.7±1.0	86.1±1.0	63.9±0.6	78.9±2.2
UAR-L	93.8±1.2	87.6±0.6	87.1±0.4	80.3±1.1	71.0±2.4	88.4±0.7
PPL	83.6±0.0	85.7±0.0	94.2±0.0	89.7±0.0	81.7±0.0	87.1±0.0
Binoculars	93.1±0.0	82.6±0.0	90.5±0.0	93.8±0.0	79.0±0.0	95.4±0.0
Fast-DetectGPT	91.9±0.0	80.3±0.0	88.2±0.0	91.0±0.0	79.1±0.0	93.7±0.0
Ghostbuster	94.3±0.0	87.3±0.0	81.9±0.0	95.4±0.0	73.3±0.0	94.5±0.0
R-SENTRA	94.6±0.5	88.4±0.4*	94.9±0.2	97.7±0.3*	83.9±1.3	97.4±0.3
SENTRA	92.3±1.0	88.0±0.1	95.0±0.3*	97.7±0.2	87.1±1.7*	97.7±0.3*

Table 6: Mean and standard deviation of the AUROC across random seeds on the M4GT dataset.

	cmv	eli5	hswag	roct	sci_gen	squad	tldr	wp	xsum	yelp
RoBERTa	94.8±1.0	92.9±0.7	87.4±4.2*	88.8±1.0*	84.3±6.5	93.3±1.0	85.7±5.1	90.3±1.5	74.4±3.4	91.3±1.6
Text-Fluoroscopy	62.1±0.0	61.9±0.0	69.5±0.0	71.6±0.0	79.1±0.0	53.3±0.0	73.2±0.0	56.5±0.0	47.8±0.0	64.3±0.0
UAR-D	80.2±1.8	74.4±1.7	63.5±2.3	61.5±2.5	56.5±4.7	59.6±3.4	60.1±1.7	67.8±3.3	40.5±0.9	70.3±0.4
UAR-L	90.1±0.7	81.9±0.7	61.2±2.4	73.5±1.0	80.6±1.7	76.1±0.8	66.3±2.8	88.2±0.9	69.0±1.9	77.5±1.3
PPL	57.9±0.0	61.4±0.0	73.8±0.0	61.2±0.0	49.4±0.0	48.3±0.0	62.9±0.0	59.4±0.0	45.7±0.0	51.9±0.0
Binoculars	71.0±0.0	70.2±0.0	59.3±0.0	52.9±0.0	59.7±0.0	55.3±0.0	63.4±0.0	67.2±0.0	57.6±0.0	60.5±0.0
Fast-DetectGPT	71.3±0.0	70.1±0.0	66.1±0.0	60.5±0.0	56.4±0.0	57.4±0.0	66.2±0.0	64.5±0.0	54.9±0.0	62.1±0.0
Ghostbuster	90.5±0.0	86.0±0.0	66.2±0.0	65.0±0.0	83.6±0.0	78.8±0.0	74.0±0.0	94.1±0.0	72.4±0.0	80.9±0.0
R-SENTRA	98.5±0.2	95.2±0.7	84.6±0.6	87.3±0.6	97.9±0.1*	94.1±0.3*	93.4±0.3	98.6±0.3	93.8±1.7	94.4±0.2
SENTRA	98.6±0.2*	95.4±0.4*	86.0±0.3	88.2±0.5	97.6±0.8	93.9±0.6	94.1±0.4*	98.9±0.1*	94.4±1.0*	95.1±0.2*

Table 7: Mean and standard deviation of the AUROC across random seeds on the MAGE-OOD dataset.

	GLM130B	_7B	bloom_7b	flan_t5_small	gpt.3.5.trubo	gpt_j	opt_125m
RoBERTa	77.1±28.7	96.9±0.6*	94.6±1.3*	69.9±22.0	90.3±0.5	85.4±19.6	95.3±0.9*
Text-Fluoroscopy	28.3±0.0	35.7±0.0	42.4±0.0	55.7±0.0	39.0±0.0	41.2±0.0	48.4±0.0
UAR-D	80.4±1.3	70.5±0.6	75.3±0.8	66.3±1.1	70.3±1.8	73.3±0.9	65.8±1.3
UAR-L	82.8±0.6	71.4±0.7	83.9±0.5	70.7±0.6	77.4±0.6	92.3±0.2	84.4±1.2
PPL	91.9±0.0	92.8±0.0	41.8±0.0	35.7±0.0	90.5±0.0	25.4±0.0	35.1±0.0
Binoculars	94.7±0.0	94.8±0.0	48.1±0.0	52.3±0.0	95.2±0.0*	14.7±0.0	32.6±0.0
Fast-DetectGPT	3.8±0.0	2.8±0.0	54.5±0.0	53.1±0.0	7.6±0.0	85.8±0.0	57.8±0.0
Ghostbuster	88.8±0.0	79.8±0.0	78.1±0.0	54.5±0.0	65.7±0.0	78.1±0.0	34.3±0.0
R-SENTRA	96.8±0.2	93.9±0.9	92.5±0.8	89.9±0.6	93.3±0.3	96.4±0.3	91.5±1.0
SENTRA	97.2±0.3*	93.3±1.5	94.1±0.4	92.4±2.0*	92.6±1.4	97.5±0.5*	88.0±2.3

Table 8: Mean and standard deviation of the AUROC across random seeds on the MAGE-OOLLM dataset.

Method	Parameter Count
RoBERTa-base	124M
Text Fluoroscopy	7B (LLM) + 5.1M (FCN) ≈ 7B
UAR	82M
Perplexity	7B (LLM)
Binoculars	14B (2 LLMs)
Fast-DetectGPT	2.7B + 6B (2 LLMs) = 8.7B
Ghostbuster	7B (LLM) + N (LR, N « 7B) ≈ 7B
SENTRA	57M (training), 14B (inference)
R-SENTRA	57M (training), 14B (inference)

Table 9: Parameter count of all methods with the actual LLM(s) used in evaluation. LR stands for logistic regression, FCN stands for fully connected network. For Ghostbuster, we observed N to be between 20 to 40.

D LLM Prompting Case Study

As part of our benchmarking, we evaluated OpenAI's proprietary models, `gpt-4o-2024-08-06` ("4o") and `o1-2024-12-17` ("o1"), by prompting them directly to classify whether a given text was written by a human or generated by an AI.

To control inference costs, we limited the evaluation to 100 samples per domain/model on the same datasets from OOD and OOLLM experiments. At the time of writing, we estimate evaluating all samples would cost about $10,000 for GPT-4o and $60,000 for o1. This inference cost is many orders of magnitude more expensive than any other method.

The system prompt used to obtain the label and a confidence score is the following: *"You are an expert in identifying whether text was written by a human or generated by an AI language model. You are tasked to identify if a provided text is written by a human or generated by an AI language model. Return your answer on the first line as one word only: 'Human' or 'AI'. On the second line, provide a confidence score between 0 and 1. Do not output anything else."*. The returned confidence score was interpreted as the model's probability of the predicted class, thus to compute AUROC fairly, scores were flipped for predictions labeled as "Human". The evaluation results are shown in table 10 alongside with the SENTRA scores. Due to the randomness arose from the reasoning mechanism, we ran o1 model 3 times and averaged the scores. For 4o model, we used temperature=0. These result suggests that the reasoning model (o1) is more capable of performing machine-generated text detection and the standard 4o model, achieving competitive performance on the RAID and M4GT datasets. Nevertheless, our SENTRA model consistently outperforms both 4o and o1 across all datasets.

E Hyper-parameter Selection

For RoBERTa, we chose one domain from the MAGE dataset to tune the learning rate. RoBERTa was initialized from RoBERTa base for both the supervised baseline and during contrastive pre-training. With this learning rate, the RoBERTa diverged before the first epoch on one MAGE split and one RAID split. We then turned down the learning rate for these two splits and reran RoBERTa, but the models still diverged. It is possible with additional tuning, RoBERTa could better fit these datasets, but we did not want to pay special attention to the fine-tuning any one method.

For SENTRA, we did a small search over the number of layers, {2,4,8}, for the CMV-MAGE data split by looking at the in-domain development loss. We found four layers to work best. We later found SENTRA had trouble fitting the in-distribution validation data of a data. We found that varying the LR and batch size on this dataset had no significant effect, so we kept the defaults of a LR of $1e-4$ and a batch size of 128 which were the defaults from RoBERTa. We then manually tuned the pre-training model while looking at this in-distribution loss. We ultimately found that eight layers and and two pre-training phases produced the best performance on this in distribution validation dataset.

Dataset	4o	o1	SENTRA
RAID-OOD	79.5	90.0	**91.1**
M4GT-OOD	65.4	91.1	**92.9**
MAGE-OOD	75.1	78.4	**92.9**
MAGE-OOLLM	72.1	75.3	**93.8**

Table 10: AUROC scores for OpenAI models and SENTRA. Best score per dataset is bolded.