WAVEDETECT: Robust Framework for Machine-Generated Text Detection via Wavelet Transform

Anonymous ACL submission

Abstract

As Large Language Models asymptotically approach human-level fluency in natural language generation, solely relying on surface-level semantic artifacts for detecting LLM-generated texts has become increasingly precarious. Existing detectors often falter when facing three critical challenges: adversarial perturbations, cross-domain shifts, and the rapid temporal evolution of the foundation model. To address these issues, we propose WAVEDETECT, a novel framework that reformulates text detection as a signal processing task within the time-frequency domain. Unlike previous methods that analyze static token probability distributions, WAVEDETECT models the generated output as a probability signal, upon which a differentiable Continuous Wavelet Transform is applied to convert them into learnable spectral representations. This process reveals the intrinsic "spectral fingerprints" in machine-generated texts–patterns that remain invisible in time domain. Comprehensive evaluations on three well-curated datasets (RAID, EvoBench, and Domain-Shift) show that our method achieves a new state-of-the-art. It not only achieves superior accuracy but also exhibits remarkable robustness against sophisticated attacks, generalization across out-of-distribution topics and unseen evolving LLMs. Our results validate the efficacy of spectral analysis as a promising paradigm for LLM-generated texts detection.

1 Introduction

The advent of Large Language Models (LLMs) has fundamentally reshaped the landscape of natural language generation, bringing us asymptotically close to a reality where machine-generated text is nearly indistinguishable from human writing. While this capability offers immense productivity gains, it simultaneously precipitates a crisis of information integrity (Crothers et al., 2023). The potential for misuse ranges from academic dishonesty and automated plagiarism to the industrial-scale proliferation of disinformation (Chairs, 2025). Consequently, the development of robust Machine-Generated Text (MGT) detection systems has escalated from a technical curiosity to a critical imperative for AI safety.

Despite urgency of the issue, the detection landscape is currently locked in an asymmetric arms race. As LLMs scale in parameter count and reasoning capability, the subtle artifacts traditionally used for detection are rapidly vanishing, such as perplexity gaps or entropy differences (Gehrmann et al., 2019). Contemporary detection frameworks grapple with three pivotal challenges for deployment:

- **Robustness against Adversarial Attacks:** Real-world adversaries actively employ obfuscation techniques, such as paraphrasing or synonym substitution, to evade detection. Prior work indicates that semantic-based detectors are notoriously brittle, often collapsing under minimal perturbations (Dugan et al., 2024; Wang et al., 2024).
- **Temporal Stability (Model Evolution):** The rapid iteration of LLMs (e.g., from GPT-3 to GPT-4o) renders detectors obsolete effectively overnight. A core challenge is preventing detectors from overfitting to the idiosyncrasies of legacy models, thereby ensuring they remain effective against unseen, next-generation architectures (Yu et al., 2025).
- **Domain Generalization:** Detectors trained on general corpora frequently fail when applied to specialized fields such as law or medicine (Chen et al., 2025b). A reliable detector must discern the fundamental distributional signature of machine generation, rather than latching onto domain-specific surface features.

Existing methodologies are typically categorized into training-based and zero-shot approaches (Yang et al., 2024). Zero-shot methods (Mitchell et al.,

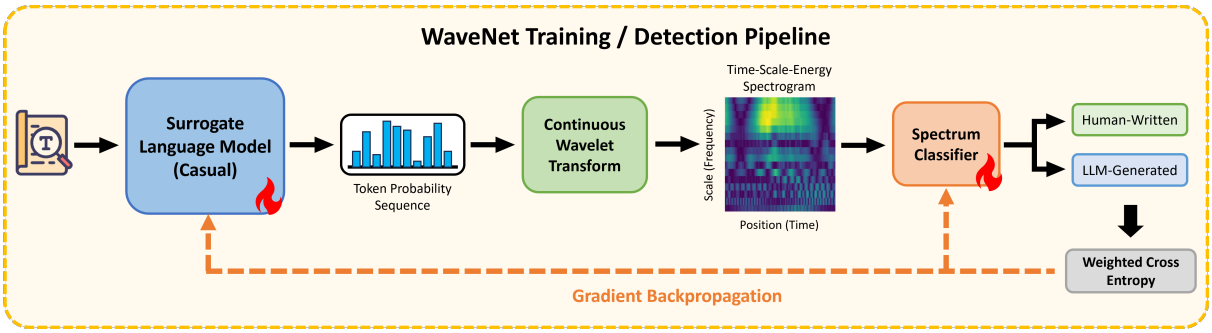

Figure 1: The overall workflow of WAVEDETECT. A surrogate language model is used to obtain probabilities sequence, CWT transforms the sequential features into time-frequency domain, and a CNN is used to extract spectral features for classification.

2023; Hans et al., 2024), favored for their interpretability, often lack the resilience required for high-stakes environments such as under attack (Wang et al., 2024). Conversely, training-based supervised classifiers (Solaiman et al., 2019) offer higher accuracy but are prone to the aforementioned overfitting, struggling to generalize out-of-distribution domains. We hypothesize that this limitation stems from a reliance on the *time-domain* representation of text (i.e., the raw sequence of tokens), where the boundary between human and machine semantics is increasingly blurred.

To overcome these limitations, we propose a paradigm shift from semantic analysis to **spectral analysis**. Recent psycholinguistics-inspired studies (Xu et al., 2024; West et al., 2025) suggest that while human and machine texts may be semantically close, they exhibit distinct regular patterns in their probability distributions, fingerprint-like features of the MGTs. Nevertheless, simplistic Fourier-based methods (e.g., FourierGPT (Xu et al., 2024)) tend to lose essential temporal localization information, which is pivotal for capturing fine-grained differences.

Therefore, drawing on these insights, we hypothesize that while human-written texts (HWTs) and MGTs may be semantically indistinguishable, they exhibit distinct rhythmic patterns in their probability distributions. Basing on such assumption, we introduce **WAVEDETECT**, a novel framework that reformulates text detection as a signal processing problem in the time-frequency domain. By transforming the sequence of token probabilities into continuous signals and decomposing them via Continuous Wavelet Transform (CWT) (Mallat, 1989), captures both the global spectral features and local transient features, which allows a trainable detector to learn the underlying "spectral fingerprints" that are invariant to semantic paraphrasing and domain shifts.

We conduct a comprehensive evaluation within RAID (Dugan et al., 2024), EvoBench (Yu et al., 2025), and Domain Shift (Chen et al., 2025b). To summarize, our main contributions are as follows:

- **WAVEDETECT**, a novel MGT detector is proposed, which leverages CWT to extract robust time-frequency features, hence effectively characterizing the temporal dynamics of token probabilities that goes beyond surface text features.
- New state-of-the-art performance is achieved in detection robustness, significantly outperforming baselines under challenges from adversarial attacks, model evolution, and domain shifts.
- Visualization of the spectrum features show that human and machine text distributions can be disentangled in spectral space.

2 Related Work

2.1 Machine-Generated Text Detection

Early approaches trained BERT-like models for machine text classification, as seen in (Solaiman et al., 2019; Abassy et al., 2024). Instead of direct training, Guo et al. (2024) employed multi-level contrastive learning to train the detector. Specifically, RADAR (Hu et al., 2023) utilized adversarial training to develop a detector, ensuring robustness against adversarial attacks. Other zero-shot methods exploit inherent behavioral differences between machine-generated and human-written text. For example, DetectGPT (Mitchell et al., 2023) observed that machine-generated text is more sensitive to minor perturbations; substituting words in machine text leads to significant changes in the loss. Fast-DetectGPT (Bao et al., 2024) subsequently found

that such perturbations can be simulated directly during the final vocabulary decision step, which largely accelerates detection speed. Binoculars (Hans et al., 2024) normalizes the perplexity of the input text using the inherent differences between two scoring models, establishing a strong zero-shot baseline. RepreGuard (Chen et al., 2025a) uses the internal hidden state representations of a surrogate model to calculate a hyperplane that distinguishes between machine and human text, achieving stable out-of-distribution (OOD) performance.

2.2 Spectrum Methods

Previous studies have demonstrated significant differences between the spectra of human-written and model-generated text (Yang et al., 2023; Liu et al., 2025). FourierGPT (Xu et al., 2024) leverages these differences to detect machine-generated text by applying the Fourier transform to the surprisal sequence of the input text. Based on the same assumption, West et al. (2025) employed the Wavelet transform instead of the Fourier transform, allowing for the utilization of both spectral and temporal information. These works demonstrate the effectiveness of detectors based on spectral analysis, highlighting their high potential for future developments in machine text detection.

3 Methodology

In this section, we introduce the workflow of WAVEDETECT, with focus on the technique of extracting the latent spectral fingerprints of the input text. As illustrated in Figure 1, we build a signal processing-inspired pipeline, in which discrete input tokens are transformed into continuous-valued probability signals, and then decompose them via a differentiable CWT, and finally leverage a convolutional neural network (CNN) to capture the most salient time-frequency features.

3.1 Signal Extraction and Spectral Hypothesis

Formally, let $\mathcal{X} = \{x_1, \ldots, x_N\}$ denote a sequence of tokens. We formulate detection as a binary hypothesis test between human distribution $\mathcal{D}_\mathcal{H}$ and machine distribution $\mathcal{D}_\mathcal{M}$. We define a mapping $\Phi : \mathcal{V}^N \to \mathbb{R}^N$ utilizing a surrogate language model $\mathcal{M}_\theta$ to transform discrete tokens into a continuous probability signal $\mathbf{p}$:

$$\mathbf{p}_t = \Phi(x_t) = P_{\mathcal{M}_\theta}(x_t \mid x{<}\, t), \quad t = 1, \ldots, N \tag{1}$$

We hypothesize that human and machine signals exhibit distinct spectral characteristics due to their generative mechanisms. Human writing is inherently stochastic, characterized by a high-entropy selection process that introduces irregular "spikes" and chaotic fluctuations in $\mathbf{p}_t$. In contrast, machine generation is limited by decoding strategies that truncate the probability tail[1]: $x_t \sim \text{Truncate}\left(P_{\text{gen}}(\cdot \mid x_{<t})\right)$ where the truncation acts as a **low-pass filter** or regularizer, suppressing high-frequency complexity in the probability signal. Therefore, detecting $\mathcal{D}_\mathcal{M}$ is equivalent to identifying these spectral artifacts.

3.2 Continuous Wavelet Transform

Since natural language probability signals are inherently *non-stationary*, as their statistical properties shift with context, global spectral methods like Fourier Transform fail to capture localized transient features. We therefore employ CWT to decompose $\mathbf{p}$ into a time-frequency representation.

We select the Morlet wavelet (Grossmann and Morlet, 1984) as the mother wavelet $\psi(t)$ due to its Gaussian envelope, which offers an optimal balance between time and frequency resolution:

$$\psi(t) = \pi^{-1/4} e^{i\omega_0 t} e^{-t^2/2} \tag{2}$$

where ω_0 is the central frequency. The CWT of the signal $\mathbf{p}(t)$ at scale s and translation τ is defined as the convolution of the signal with the scaled and translated wavelet:

$$W(s, \tau) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} \mathbf{p}(t) \psi^* \left(\frac{t - \tau}{s} \right) dt \tag{3}$$

In our implementation, we treat this as a fixed-weight 1D convolution layer with a bank of kernels corresponding to distinct scales. The magnitude $|W| \in \mathbb{R}^{K \times N}$ serves as the spectral map. Since there exists meaningless padding tokens in batch training, we apply right-side zero-padding on the spectrum as:

$$\mathbf{S}_{ij} = \begin{cases} |W(s_i, \tau_j)| & \text{if } j \leq N \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

which ensures that the spectral features of the valid text are preserved without distortion from padding artifacts in the time domain.

[1] Greedy decoding can be a special case of decoding strategies when $\text{Truncate}(\cdot) = \text{argmax}(\cdot)$

3.3 Spectrum Map Feature Extraction

The spectral map $\mathbf{S}$ serves as a visual fingerprint of the text's generation process. To capture the special spectrum patterns between MGT and HWT, we feed $\mathbf{S}$ into a lightweight CNN, effectively treating the detection task as image classification. The network outputs a probability $\hat{y} = \sigma(\text{CNN}(|\mathbf{W}|))$ to classify the text source as human or model.

3.4 Optimization Strategy

To handle the optimization gap between the pre-trained surrogate model and the randomly initialized CNN, we employ a two-stage training protocol: (1) **Warm-up:** Freezing $\mathcal{M}_\theta$ to align the CNN with spectral features, it forces the CNN to learn extracting robust features from the intrinsic spectral patterns of the pre-trained surrogate's output; (2) **Joint Training:** Unfreezing $\mathcal{M}_\theta$ to refine the probability distribution for detection, it allows the surrogate model to dig deeper in the differences between MGT and HWT, and demonstrate such differences in spectrum space.

We optimize using a Weighted Cross-Entropy (WCE) loss to address class imbalance and maximize the decision margin:

$$\mathcal{L} = -\frac{1}{B}\sum_{i=1}^{B}\left[\alpha y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{5}$$

where α balances the contributions of positive (machine) and negative (human) samples. Comparing to other contrastive learning based approaches (Guo et al., 2024), WCE allows an easy implementation in distribute data parallel, which is friendly for large scale training.

4 Experiments

We conduct a comprehensive suite of experiments to validate our method, aiming to answer the three research questions mentioned in Section 1:

- **RQ1:** Is WAVEDETECT's performance robust against adversarial attack on in-distribution dataset?
- **RQ2:** Can we maintain the performance as LLMs continually evolve?
- **RQ3:** Can we generalize the detection capabilities to domains that are unseen before?

4.1 Experiment Setup

Training Data Construction To ensure the model learns generalized spectral features rather than overfitting to specific content, we utilize the RAID benchmark (Dugan et al., 2024) as our primary training corpus. RAID encompasses 11 diverse LLMs employing 3 distinct decoding strategies across 8 domains (details in Appendix A). To strictly mitigate data leakage, we implement a prompt-level splitting strategy. We randomly withhold 150 source prompts (100 for testing, 50 for validation) and all their corresponding generated texts. The remaining prompts and their derivations constitute the training set (approx. 6 million samples). This protocol ensures that the detector never encounters generations derived from the same semantic context during inference, forcing it to rely on structural generation artifacts. In this work, we denote the training set as RAID-all, and the subset that excludes all sampling and adversarial augmentation as RAID-base, and the test set as RAID-test[2].

Evaluation Protocol Echoing the challenges outlined in Section 1, we evaluate WAVEDETECT across three rigorous dimensions:

1. **Adversarial Robustness (RAID-test):** We utilize the withheld RAID test set to evaluate performance under standard conditions and 11 adversarial attack scenarios (e.g., homoglyph substitution, zero-width injection, and paraphrasing). This assesses the detector's resilience to active evasion attempts.
2. **Temporal Stationarity (EvoBench):** To assess robustness against "Model Evolution," we employ EvoBench (Yu et al., 2025), which contains generations from evolving versions of LLMs (e.g., GPT-4o-0513 → GPT-4o-0806 → GPT-4o-1120). Crucially, these newer models and the closed-source families (Claude-3.5, Gemini-1.5) were never seen during WAVEDETECT's training.
3. **OOD Domain Generalization:** We utilize the dataset from DivScore (Chen et al., 2025b) to test Out-of-Distribution (OOD) generalization. The dataset includes two domains: *Medical* (PubMed (Jin et al., 2019), MIMIC (Johnson et al., 2023)) and *Legal* (LawStack (Moslem, 2023), OALC (Butler, 2023)). These distributions differ significantly from the training set, and we ensure that they were never seen during training.

[2]The formal evaluation on RAID's test set requires a pull-request in RAID's github repository: https://github.com/liamdugan/raid. We manually conduct train-test split in this study for a convenient local evaluation.

Detectors	ChatGPT	GPT-2	GPT-3	GPT-4	Cohere	Cohere-C	Llama-C	Mistral	Mistral-C	MPT	MPT-C	Total
AUROC (↑)												
RoBERTa	0.6825	0.6144	0.5985	0.5954	0.5449	0.6529	0.6955	0.5267	0.6921	0.4400	0.5772	0.6018
RADAR	0.8968	0.7881	0.8750	0.8661	0.7516	0.8375	0.8777	0.7185	0.8823	0.7589	0.8518	0.8277
Fast-DetectGPT	0.8715	0.7735	0.8578	0.8014	0.8060	0.8277	0.8653	0.6513	0.8023	0.5391	0.6369	0.7666
Binoculars	0.9483	0.7803	0.9473	0.9047	0.8813	0.9213	0.9419	0.6962	0.9008	0.5751	0.7582	0.8414
FourierGPT	0.7454	0.8035	0.8379	0.7854	0.6885	0.7410	0.7855	0.7537	0.7110	0.8416	0.7764	0.7700
RepreGuard	0.8184	0.5501	0.5758	0.7844	0.5681	0.6525	0.8538	0.5510	0.7691	0.4395	0.6528	0.6560
WaveDetect-base	0.8986	0.7548	0.9265	0.8696	0.8415	0.8836	0.8929	0.7313	0.8966	0.6528	0.8460	0.8359
WaveDetect-all	**0.9944**	**0.9597**	**0.9953**	**0.9924**	**0.9462**	**0.9737**	**0.9957**	**0.9638**	**0.9931**	**0.9601**	**0.9895**	**0.9785**
TPR@0.1%FPR (↑)												
RoBERTa	0.0572	0.0385	0.0111	0.0185	0.0080	0.0614	0.0671	0.0175	0.1240	0.0127	0.0791	0.0450
RADAR	0.0060	0.0078	0.0312	0.0025	0.0010	0.0105	0.0029	0.0027	0.0094	0.0079	0.0193	0.0092
Fast-DetectGPT	0.0051	0.0045	0.0047	0.0038	0.0036	0.0041	0.0054	0.0031	0.0043	0.0028	0.0026	0.0040
Binoculars	0.4838	0.3094	0.5470	0.3212	0.1667	0.3854	0.2257	0.2256	0.4468	0.1979	0.3284	0.3584
FourierGPT	0.0101	0.0976	0.0216	0.0096	0.0013	0.0021	0.0123	0.0496	0.0013	0.0859	0.0015	0.0266
RepreGuard	0.0687	0.0192	0.0145	0.0619	0.0119	0.019	0.0971	0.013	0.0539	0.012	0.0108	0.0347
WaveDetect-base	0.0743	0.0366	0.2642	0.0283	0.0354	0.1003	0.0392	0.0238	0.0971	0.0135	0.1488	0.0783
WaveDetect-all	**0.7141**	**0.3753**	**0.6751**	**0.5937**	**0.4466**	**0.5404**	**0.6709**	**0.4054**	**0.6512**	**0.3643**	**0.5803**	**0.5470**

Table 1: **Detection Performance on RAID Test Set.** We report the Area Under the Receiver Operating Characteristic (AUROC) and the True Positive Rate at 0.1% False Positive Rate (TPR@0.1%FPR). The best results are highlighted in **bold**.

Baseline Detectors We benchmark WAVEDETECT against a comprehensive suite of detectors, categorized by their operational paradigm:

• **Supervised Detectors:** We include **OpenAI-RoBERTa** (Solaiman et al., 2019), a classic RoBERTa-large model fine-tuned for binary classification; and **RADAR** (Hu et al., 2023), a detector trained by an adversarial training framework, where a paraphraser and a detector play a min-max game to improve robustness of text detection.

• **Zero-Shot Methods:** We select **Fast-DetectGPT** (Bao et al., 2024), an efficient successor to DetectGPT, based on the hypothesis that MGT lies in regions of positive curvature within the model's log-probability landscape. It estimates the conditional probability curvature to distinguish machine text from human writing; **Binoculars** (Hans et al., 2024), which calculates a score based on the ratio of the perplexity computed by an observer model to that of a performer model; **RepreGuard** (Chen et al., 2025a), which assumes that human and machine texts form distinct manifolds in the latent space. It applies PCA on human and model texts' hidden states to define a separation hyperplane for classification; and **FourierGPT** (Xu et al., 2024), a most relevant baseline which applies Fourier transform to the surprisal sequences and conduct classification in the spectrum space. Comparing against this highlights the advantage of our Wavelet-based approach (Time-Frequency vs. Frequency-only).

Implementation Details

• **Architecture:** We use **Qwen2.5-0.5B-Base** (Team, 2024) as the surrogate probability estimator $\mathcal{M}_\theta$ for its efficiency and strong capabilities. The spectral encoder is based on a ResNet-18 backbone modified for single-channel spectrogram input.

• **Training Strategy:** We adopt the two-stage protocol described in Section 3.4 using the AdamW optimizer with a batch size of 64 and epochs of 3. In the warm-up stage, the CNN is trained from scratch with a learning rate of 1×10^{-3}, and $\mathcal{M}_\theta$ is frozen. In the joint training stage, we jointly train the entire WAVEDETECT (all parameters) with a learning rate of 1×10^{-5}. To address the data imbalance (HWT vs. MGT) issue, we apply the Weighted Cross-Entropy loss with a weight ratio of 9:1 (HWT:MGT). All experiments are conducted on $4\times$ NVIDIA RTX 6000 Ada GPUs. The detector trained on RAID-all is referred to as WAVEDETECT-ALL, and the one trained on RAID-base is WAVEDETECT-BASE.

4.2 Basic Performance on RAID-test

Table 1 summarizes the performance of WAVEDETECT against four representative baselines across 11 diverse source models. WAVEDETECT establishes a decisive state-of-the-art performance on RAID-test dataset, achieving an average AUROC of **0.9785**. This represents a substantial improvement over the strongest zero-shot baseline, Binoculars (0.8414), and the supervised baseline, RADAR

Detectors	GPT-4o			GPT-4			Claude-Sonnet			Gemini-1.5-Flash		
	AUC-m	Std	max|Δ|	AUC-m	Std	max|Δ|	AUC-m	Std	max|Δ|	AUC-m	Std	max|Δ|
RoBERTa	0.6216	0.0802	0.1507	0.6866	0.0154	0.0376	0.6866	0.0226	0.0307	0.3132	0.2578	0.5453
RADAR	0.7851	0.0153	0.0292	0.7825	0.0122	0.0150	0.7803	0.0309	0.0742	0.7328	**0.0129**	0.0308
Fast-DetectGPT	0.7559	0.0253	0.0676	0.7621	0.0120	0.0237	0.8007	0.0490	0.0982	0.7371	0.0557	0.1104
Binoculars	0.8364	0.0118	0.0323	0.8276	0.0036	0.0087	0.8406	0.0334	0.0644	**0.8013**	0.0398	0.0801
FourierGPT	0.6489	0.0423	0.0882	0.5984	0.0172	0.0306	0.6728	0.0149	0.0264	0.5867	0.0222	0.0381
RepreGuard	0.6618	0.0117	0.0172	0.6562	0.0081	0.0160	0.6434	**0.0116**	**0.0231**	0.6319	0.0295	**0.0298**
WaveDetect-base	**0.8536**	**0.0048**	**0.0087**	**0.8404**	**0.0023**	**0.0052**	**0.8767**	0.0304	0.0682	0.7844	0.0404	0.0869
WaveDetect-all	0.8145	0.0063	0.0179	0.8009	0.0049	0.0086	0.8180	0.0251	0.0572	0.7390	0.0380	0.0794

Table 2: **Temporal Robustness on EvoBench.** We evaluate detectors on evolving versions of LLMs. AUC-m denotes the mean AUROC across versions, Std is the standard deviation, and max$|\Delta|$ represents the maximum performance changes between versions. Lower Std and max$|\Delta|$ indicate better stability.

(0.8277). Notably, purely semantic-based supervised methods like RoBERTa struggle significantly (0.6018), often degenerating to near-random guessing on unseen models (e.g., MPT: 0.44). This supports our hypothesis that spectral features are more invariant across different generator architectures than surface-level semantic artifacts.

In real-world scenarios, maintaining a low False Positive Rate (FPR) is critical to prevent false accusations. As shown in the bottom half of Table 1, WAVEDETECT demonstrates exceptional reliability in this strict regime. It achieves a total TPR of **54.70%** at 0.1% FPR, outperforming Binoculars by nearly 19 percentage points. In contrast, other methods fail to distinguish machine text effectively at such high precision thresholds. This result highlights WAVEDETECT's potential for deployment in sensitive applications such as academic integrity review.

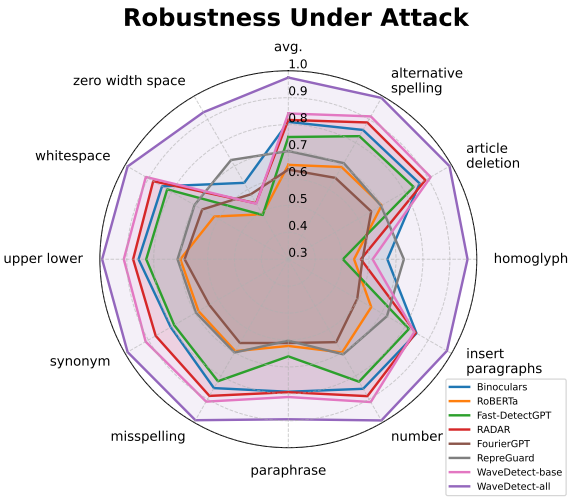

Figure 2: **Robustness against Adversarial Attacks.** We visualize the detection performance (AUROC) under 11 different attack methods using a radar chart. WAVEDETECT exhibits a dominant performance envelope, completely enclosing all baseline methods, maintaining a larger consistent envelope compared to baselines.

4.3 RQ1: Robustness against Adversarial Attacks

Real-world deployment of detectors often faces adversarial attempts to bypass detection. We evaluate WAVEDETECT against 11 diverse attack methods ranging from character-level noising to semantic rewriting. Figure 2 visualizes the performance comparison using a radar chart.

As illustrated in Figure 2, WAVEDETECT (WAVEDETECT-ALL, represented by the purple line) demonstrates a dominant advantage in adversarial attacks, largely outperforming all the baselines. In the hardest two attack types, zero-width space and homoglyph, WAVEDETECT maintains a very high performance (above 90% acc.), while most other baselines suffer from significant performance degradation. This indicates that detectors trained on spectral features exhibit greater robustness compared to those based on sequential features in time domain.

When comparing WAVEDETECT-ALL with WAVEDETECT-BASE, we observe that although WAVEDETECT-BASE outperforms other baselines in overall performance, it shows a noticeable decline in the zero-width-pad and homoglyph tasks. This suggests that while the model trained on RAID-base performs relatively well against adversarial attacks, its robustness is still insufficient. Incorporating domain-specific data for augmentation during training can significantly enhance the model's robustness against adversarial attacks.

WAVEDETECT experiences a biggest performance drop occurs to paraphrase task, however, a cross-comparison with other baselines reveals

Detectors	Medical			Legal			All Avg.
	MIMIC	PubMed	Avg.	LawStack	OALC	Avg.	
RoBERTa	0.6078	0.3941	0.5010	0.2142	0.4839	0.3490	0.4250
RADAR	0.2319	0.1852	0.2085	0.6199	0.6264	0.6232	0.4159
Fast-DetectGPT	0.6992	0.8793	0.7892	0.7961	0.8267	0.8114	0.8003
Binoculars	0.9549	0.9387	0.9468	0.9240	0.8754	0.8997	0.9232
FourierGPT	0.9380	**0.9956**	0.9668	0.8928	0.7976	0.8452	0.9060
RepreGuard	0.9482	0.9950	**0.9716**	0.8882	0.7215	0.8048	0.8882
WaveDetect-base	0.9249	0.6915	0.8082	0.8511	0.8382	0.8447	0.8264
WaveDetect-all	**0.9554**	0.9619	0.9586	**0.9608**	**0.9461**	**0.9535**	**0.9560**

Table 3: **OOD Generalization on Domain Shift.** We demonstrate the AUROC performance on specialized domains (Medical and Law), ensuring that these domains were not seen during training.

that its margin of decline is smaller. Paraphrasing manifests as changes in structural features such as wording and syntax, it demonstrates that WAVEDETECT captures the high-level, deep intrinsic features of MGT and is less susceptible to variations in text structure.

4.4 RQ2: Effectiveness under LLM Evolution

A persistent challenge in deepfake detection is "model evolution", as detectors often become obsolete as LLMs are updated. Table 2 investigates the stability of WAVEDETECT across different versions of closed-source models. WAVEDETECT-BASE demonstrates superior temporal stability, achieving the highest mean AUROC and the lowest performance fluctuation (Std and max$|\Delta|$) on GPT-4o, GPT-4, and Claude-Sonnet. Meanwhile, WAVEDETECT-ALL achieves a secondary performance, very close to that of Binoculars. The mixed training of adversarial data harms the performance of WAVEDETECT in a certain degree, as they blur the demarcation line between human texts and model texts. However, the underlying probabilistic signature is still able to be captured by our surrogate model and CWT, thereby remains relatively stationary. Although Binoculars performs relatively closer to WAVEDETECT on AUROC, WAVEDETECT often offers a more consistent performance across these model families. FourierGPT and RepreGuard performs well in stationary, however, they performs poor in AUROC. In sum, WAVEDETECT gives an overall well performance on both AUROC and temporal stationary.

4.5 RQ3: Performance under Domain-Shift

We assess the ability of detectors to generalize to OOD topics in Table 3. Traditional training-based methods suffer catastrophically from domain shifts. RoBERTa and RADAR drop to near-random performances in many tasks, and even worse than random (Avg. $\sim$0.42) on average performance, likely due to overfitting on the vocabulary of the training domain. Zero-shot detectors, Binoculars, FourierGPT and RepreGuard, achieve relatively good average performance while generalizing to other domains, especially FourierGPT and RepreGuard on medical domains. However, there exists a significant performance drop on legal domain of these methods. In contrast, WAVEDETECT-ALL exhibits remarkable robustness, achieving the highest average AUROC of **0.9560**, and performs stable across all OOD areas and all datasets. This confirms that the features captured by WAVEDETECT from spectral patterns of machine-generated text are "topic-agnostic". The rhythmic probability fluctuations exist whether the text is about clinical trials or legal precedents, irrelevant to the domain.

5 Explanation Study

5.1 Spectral Mechanism Analysis

To demystify the decision-making process of WAVEDETECT, we visualize both the raw wavelet spectrum and the corresponding Class Activation Maps (CAM) (Selvaraju et al., 2019) in Figure 3. This analysis reveals how the detector disentangles MGT and HWT within the time-frequency domain.

From direct visual inspection, the raw wavelet spectra of MGT and HWT do not exhibit a clear separation, as energy is broadly distributed across time–scale regions for both classes. Nevertheless, the wavelet transform remains essential by reorganizing the token-level probability sequence into a multi-scale representation that exposes scale-specific patterns beyond simplistic spectral inspection–The CAM results reveal a clear con-

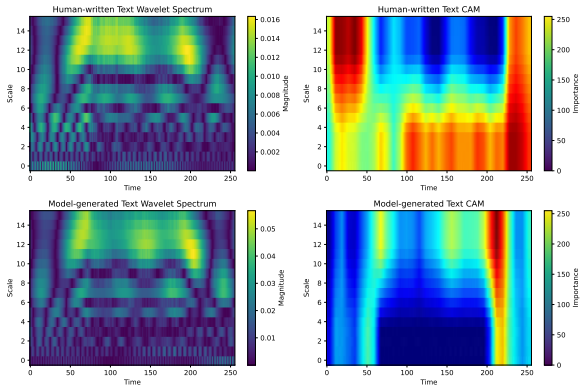

Figure 3: Wavelet spectrum and CAM visualizations. Left: wavelet spectrum representations (no clear visual separation). Right: CAM visualizations obtained from the trained CNN classifier applied to the wavelet spectra (highlighting discriminative regions).

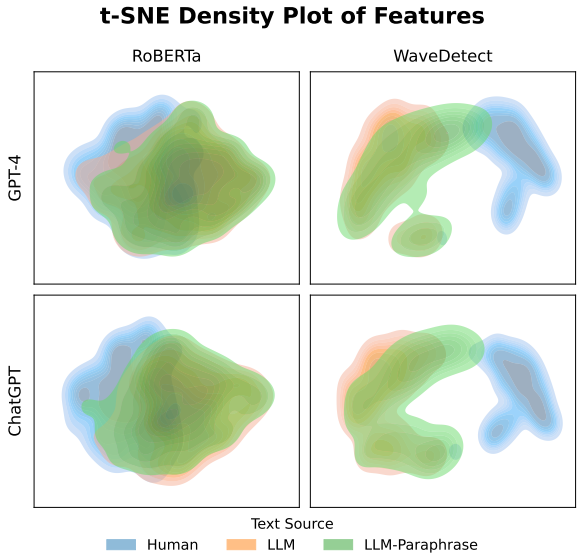

Figure 4: t-SNE visualization of dimensionality-reduced features extracted from the last layer of the detector. We present density plots representing the features of human-written text, the corresponding LLM-generated text (via GPT-4 and ChatGPT), and the LLM-generated text subjected to paraphrase attacks.

trast in the detector's focus: activations for HWT are dominated by smaller-scale (higher-frequency) components, corresponding to short-range temporal structure, whereas MGT induces stronger responses at larger-scale (lower-frequency) regions, capturing long-range regularities. Taken together, it suggests that the CNN leverages relative, scale-dependent structures rather than absolute spectral energy, revealing discriminative cues that are otherwise obscured under direct spectrum inspection.

5.2 Distribution of Human vs. LLM Latent Features in WAVEDETECT

Figure 4 presents the t-SNE visualization of the text classification features extracted from the last layer of WAVEDETECT, illustrating the distributional differences between HWT and MGT (from GPT-4 and ChatGPT, including those subjected to paraphrase attacks). As observed in the figure, there is a clear, distinguishable gap between the two kinds. It shows that WAVEDETECT is able to achieve nearly 100% accuracy in distinguishing standard GPT-4 and ChatGPT texts. However, regarding the machine texts modified by paraphrase attacks, although their distribution shape remains largely similar to the original machine texts with significant overlap, a small portion of these texts overlaps with the human text cluster. Therefore, we believe it is on these specific overlapping features that WAVEDETECT fails to distinguish, rendering them relatively undetectable.

6 Conclusion

In this paper, we introduced WAVEDETECT, a training-based text detection framework based on wavelet transform. Our approach converts texts to probability sequences using a surrogate model, and then captures their frequency-domain representations via wavelet transform, and utilizes a CNN to extract and aggregate the features for final classification. By jointly training the surrogate model and the CNN, we enable the model to maximally learn the discrepancies between human-written and machine-generated text.

We carry out comprehensive evaluations on the RAID dataset, test the method's validity against adversarial attacks (RAID), LLM evolution (EvoBench), and out-of-domain tasks (Domain Shift). Experimental results indicate that WAVEDETECT exhibits a dominant advantage on these benchmarks, with its robustness and generalization capabilities significantly outperforming state-of-the-art baselines. These findings confirm that using spectral features can filter out noise in token-level probabilities, and thus can better capture implicit temporal dynamics of text generation. In sum, our work offers a novel and valuable perspective for further advancing the development of robust, generalizable machine-generated text (MGT) detectors.

7 Limitations

Our approach is currently limited to the RAID dataset, which was released over a year ago. The LLMs included in RAID are relatively old or weak compared to the current generation of models. Therefore, we believe that a detector trained on text from modern LLMs would perform even better, while adversarial attacks on modern MGTs might also prove more challenging to detect. We emphasize the urgent need for large-scale text detection datasets that incorporate modern LLMs.

Although WAVEDETECT exhibits strong and robust performance, the underlying mechanism of the wavelet transform requires further exploration. Specifically, factors such as the choice of mother wavelet and the range of scales need to be better understood. Our future work aims to investigate these aspects for further enhancement.

References

Mervat Abassy, Kareem Elozeiri, Alexander Aziz, Minh Ngoc Ta, Raj Vardhan Tomar, Bimarsha Adhikari, Saad El Dine Ahmed, Yuxia Wang, Osama Mohammed Afzal, Zhuohan Xie, Jonibek Mansurov, Ekaterina Artemova, et al. 2024. LLM-DetectAIve: a tool for fine-grained machine-generated text detection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 336–343, Miami, Florida, USA. Association for Computational Linguistics.

Micael Arman. 2020. Poems dataset (nlp). https://www.kaggle.com/datasets/michaelarman/poemsdataset. Kaggle dataset.

David Bamman and Noah A. Smith. 2013. New alignment methods for discriminative book summarization. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

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.

Aaditya Bhat. 2023. Gpt-wiki-intro. https://doi.org/10.57967/hf/0326. Dataset for Wikipedia-style introductions.

Michał Bień, Michał Gilski, Martyna Maciejewska, Wojciech Taisner, Dawid Wiśniewski, and Agnieszka Ławrynowicz. 2020. Recipenlg: A cooking recipes dataset for semi-structured text generation. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 22–28, Dublin, Ireland. Association for Computational Linguistics.

Matyáš Boháček, Michal Bravanský, Filip Trhlík, and Václav Moravec. 2022. Fine-grained czech news article dataset: An interdisciplinary approach to trustworthiness analysis.

Umar Butler. 2023. Open australian legal qa.

ICLR 2026 Program Chairs. 2025. ICLR 2026 Response to LLM-Generated Papers and Reviews.

Xin Chen, Junchao Wu, Shu Yang, Runzhe Zhan, Zeyu Wu, Ziyang Luo, Di Wang, Min Yang, Lidia S. Chao, and Derek F. Wong. 2025a. Repreguard: Detecting llm-generated text by revealing hidden representation patterns. *Preprint*, arXiv:2508.13152.

Zhihui Chen, Kai He, Yucheng Huang, Yunxiao Zhu, and Mengling Feng. 2025b. Divscore: Zero-shot detection of llm-generated text in specialized domains. *Preprint*, arXiv:2506.06705.

Cohere. 2024. Cohere.

Evan Crothers, Nathalie Japkowicz, and Herna Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, PP:1–1.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, et al. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, et al. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 377–384. ACM.

Alexander Grossmann and Jean Morlet. 1984. Decomposition of hardy functions into square integrable wavelets of constant shape. *Siam Journal on Mathematical Analysis*, 15:723–736.

Xun Guo, Shan Zhang, Yongxin He, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. 2024. Detective: Detecting ai-generated text via multi-level contrastive learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 88320–88347. Curran Associates, Inc.

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.

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

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *Preprint*, arXiv:1909.06146.

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-wei H. Lehman, Leo A. Celi, and Roger G. Mark. 2023. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1.

Zhichen Liu, Yongyuan Li, Yang Xu, Yu Wang, Yingfang Yuan, and Zuhao Yang. 2025. Evaluating text generation quality using spectral distances of surprisal. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 2444–2463, Suzhou, China. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

S.G. Mallat. 1989. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *Preprint*, arXiv:2301.11305.

MosaicML. 2023. Mpt-30b: Raising the bar for open-source foundation models.

Yasmin Moslem. 2023. Law-stackexchange dataset.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Sayak Paul and Soumik Rakshit. 2021. arxiv paper abstracts. `https://www.kaggle.com/datasets/spsayakpaul/arxiv-paper-abstracts`. Kaggle dataset.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Veselin Raychev, Pavol Bielik, and Martin Vechev. 2016. Probabilistic model for code with decision trees. *SIGPLAN Notices*, 51(10):731–747.

Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. One million posts: A data set of german online discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1241–1244. Association for Computing Machinery.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2019. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359.

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

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl;dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.

Yichen Wang, Shangbin Feng, Abe Bohan Hou, Xiao Pu, Chao Shen, Xiaoming Liu, Yulia Tsvetkov, and Tianxing He. 2024. Stumbling blocks: Stress testing the robustness of machine-generated text detectors under attacks. *Preprint*, arXiv:2402.11638.

Alva West, Yixuan Weng, Minjun Zhu, Luodan Zhang, Zhen Lin, Guangsheng Bao, and Yue Zhang. 2025. Ai-generated text is non-stationary: Detection via temporal tomography. *Preprint*, arXiv:2508.01754.

Yang Xu, Yu Wang, Hao An, Zhichen Liu, and Yongyuan Li. 2024. Detecting subtle differences between human and model languages using spectrum of relative likelihood. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10108–10121, Miami, Florida, USA. Association for Computational Linguistics.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Ruth Petzold, William Yang Wang, and Wei Cheng. 2024. A survey on detection of LLMs-generated content. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9786–9805, Miami, Florida, USA. Association for Computational Linguistics.

Zuhao Yang, Yingfang Yuan, Yang Xu, Shuo Zhan, Huajun Bai, and Kefan Chen. 2023. Face: Evaluating natural language generation with fourier analysis of cross-entropy. *Preprint*, arXiv:2305.10307.

Xiao Yu, Yi Yu, Dongrui Liu, Kejiang Chen, Weiming Zhang, Nenghai Yu, and Jing Shao. 2025. EvoBench: Towards real-world LLM-generated text detection benchmarking for evolving large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14605–14620, Vienna, Austria. Association for Computational Linguistics.

A RAID Details

The details of RAID benchmark construction are listed as follows:

A.1 Models

- GPT-4 (`gpt-4-0613`), ChatGPT (`gpt-3.5-turbo-0613`), GPT-3 (text-davinci-002), GPT-2 XL (Radford et al., 2019; Ouyang et al., 2022)
- MPT-30B, MPT-30B-Chat (MosaicML, 2023)
- Mistral-7B (Mistral-7B-v0.1), Mistral-7B-Chat (Mistral-7B-Instruct-v0.1) (Jiang et al., 2023)
- Cohere, Cohere-Chat (Cohere, 2024)
- Llama2-70B-Chat (Touvron et al., 2023)

A.2 Dataset

Dataset	Genre	Size
(Paul and Rakshit, 2021)	Abstracts	1966
(Bamman and Smith, 2013)	Books	1981
(Raychev et al., 2016)	Code	920
(Greene and Cunningham, 2006)	News	1980
(Arman, 2020)	Poetry	1971
(Bień et al., 2020)	Recipes	1972
(Völske et al., 2017)	Reddit	1979
(Maas et al., 2011)	Reviews	1143
(Bhat, 2023)	Wiki	1979
(Boháček et al., 2022)	Czech	1965
(Schabus et al., 2017)	German	1970

Table 4: Details of data source in RAID.

A.3 Decoding Strategy

RAID contains 3 decoding settings: greedy decoding, sampling with temperature=1 and top_p=1, sampling with temperature=1 and top_p=1 and repetition_penalty=1.2.

A.4 Adversarial Attacks

Spelling and Character-Level Attacks

- **Alternative Spelling:** Replaces American English spellings with British English variants (e.g., "favorite" to "favourite") using a predefined mapping dictionary.
- **Misspelling:** Inserts common human misspellings based on a manually constructed dictionary.
- **Homoglyph:** Swaps standard ASCII characters with non-standard Unicode characters that look

identical to the human eye (e.g., replacing a Latin 'e' with a Cyrillic 'e')

- **Upper-Lower Swap:** Randomly selects a percentage of tokens and flips the case of their first letter (uppercase to lowercase or vice versa).
- **Zero-Width Space:** Inserts the invisible Unicode character $U + 200B$ before and after every visible character in the generation.

Lexical and Structural Attacks

- **Synonym Swap:** Replaces tokens with highly similar candidates identified by BERT. Candidates are filtered by part-of-speech tags and FastText embedding cosine similarity to ensure high-quality, diverse substitutions.
- **Paraphrase:** Utilizes the DIPPER-11B model (a fine-tuned T5-11B) to rewrite the text while maintaining semantic meaning.
- **Article Deletion:** Searches for and deletes the articles "a", "an", and "the" at a fixed mutation rate.
- **Number Swap:** Identifies numerical digits using regular expressions and replaces a sampled percentage of them with a random alternate digit between 0 and 9.
- **Whitespace Addition:** Randomly selects inter-token spaces and adds an extra space character to simulate irregular formatting.
- **Insert Paragraphs:** Splits sentences and inserts double newline characters (`\n\n`) between a sampled percentage of them to simulate paragraph breaks.

B EvoBench

EvoBench is designed to evaluate the generalization capabilities of detectors against the real-world challenge of continuously evolving LLMs, which change over time through version updates, fine-tuning, and pruning. In our experiments, we mainly focus on the challenge brought by close-sourced LLMs version evolving.

The principle for model selection is to identify those with multiple revisions for a single scale, where no significant alterations occurred during pre-training, resulting in a count of >2 valid revisions. Therefore, we choose GPT-4o, GPT-4, Claude-Sonnet and Gemini-Flash in our experiments.

Evolving LLMs	Source
GPT-4o	gpt-4o-2024-05-13
GPT-4o	gpt-4o-2024-08-06
GPT-4o	gpt-4o-2024-11-20
GPT-4o	chatgpt-4o-latest
GPT-4o-mini	gpt-4o-mini-2024-07-18
GPT-4	gpt-4-0613
GPT-4	gpt-4-1106-preview
GPT-4	gpt-4-0125-preview
GPT-4	gpt-4-turbo-2024-04-09
Claude-Sonnet	claude-3-sonnet-20240229
Claude-Sonnet	claude-3-5-sonnet-20240620
Claude-Sonnet	claude-3-5-sonnet-20241022
Claude-Haiku	claude-3-haiku-20240307
Claude-Haiku	claude-3-5-haiku-20241022
Claude-Opus	claude-3-opus-20240229
Gemini-Flash	gemini-1.5-flash
Gemini-Flash	gemini-1.5-flash-exp-0827
Gemini-Flash	gemini-1.5-flash-latest
Qwen	Qwen/Qwen1.5-7B-Chat
Qwen	Qwen/Qwen2-7B-Instruct
Qwen	Qwen/Qwen2.5-7B-Instruct
LLaMA3	meta-llama/Meta-Llama-3.1-8B-Instruct
LLaMA3	meta-llama/Meta-Llama-3.1-70B-Instruct
LLaMA3	meta-llama/Meta-Llama-3.2-1B-Instruct
LLaMA3	meta-llama/Meta-Llama-3.2-3B-Instruct
LLaMA3	meta-llama/Meta-Llama-3.3-70B-Instruct
Fine-tuning	meta-llama/Llama-2-7b-chat-hf
Fine-tuning	lmsys/vicuna-7b-v1.5
Fine-tuning	WizardLMTeam/WizardMath-7B-V1.0
Pruning	princeton-nlp/Sheared-LLaMA-1.3B
Pruning	princeton-nlp/Sheared-LLaMA-1.3B-Pruned
Pruning	princeton-nlp/Sheared-LLaMA-2.7B
Pruning	princeton-nlp/Sheared-LLaMA-2.7B-Pruned

Table 5: Details of LLMs in EvoBench.

We use MGTs over these three domains in EvoBench: XSum (Narayan et al., 2018), WritingPrompts (Fan et al., 2018) and PubMed (Jin et al., 2019).

C Domain Shift

Domain Shift is a new problem proposed by Chen et al. (2025b). It aims to evaluate whether detectors could maintain their capabilities in specialized, high-stakes domains like medicine and law. Details about datasets used in domain shift are described in Section 4.1. For MGT curation, they select 1000 pairs of texts for each dataset, and utilized GPT-4o, O3-mini, DeepSeek-V3 (DeepSeek-AI et al., 2025b) and DeepSeek-R1 (DeepSeek-AI et al., 2025a) to generate machine texts.

D Metrics

To describe the metrics we used, we first need to define the basic components: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Based on these, the True Positive Rate (TPR) and False Positive Rate (FPR) are defined as:

$$\text{TPR} = \frac{TP}{TP + FN}, \quad \text{FPR} = \frac{FP}{FP + TN} \tag{6}$$

D.1 AUROC

The Area Under the Receiver Operating Characteristic Curve (AUROC) is a widely used threshold-independent metric that assesses the classifier's ability to distinguish between classes.

The ROC curve is generated by plotting the TPR against the FPR at various threshold settings. The AUROC is calculated as the definite integral of the ROC curve:

$$\text{AUROC} = \int_0^1 \text{TPR}(\text{FPR})\, d(\text{FPR}) \tag{7}$$

The AUROC value ranges from 0 to 1. An AUROC of 0.5 indicates random guessing, while an AUROC of 1.0 represents a perfect classifier. Intuitively, the AUROC corresponds to the probability that a randomly selected positive instance is ranked higher than a randomly selected negative instance.

D.2 TPR@0.1%FPR

While AUROC provides an overall summary of performance, it may obscure weaknesses in the low false-positive region, which is critical for safety-sensitive applications (e.g., fraud detection, medical diagnosis, or biometrics).

TPR@0.1%FPR measures the True Positive Rate specifically when the False Positive Rate is strictly controlled at 0.1% (i.e., 10^{-3}). Let τ be the decision threshold. We find τ^* such that the FPR is fixed at the target value:

$$\begin{aligned} \text{TPR@0.1\%FPR} &= \text{TPR}(\tau^*) \\ &\text{subject to } \text{FPR}(\tau^*) \leq 0.001 \end{aligned} \tag{8}$$

This metric evaluates how much of the positive class can be recalled while maintaining a very low frequency of false alarms.

E Training Cost

WAVEDETECT is trained on 4 NVIDIA RTX 6000 Ada GPUs for 5 hours and 33 minutes. Total training consisted of approximately 20.1k steps.