Whispers of Attention: Detecting AI-Generated Text through Transformer Spectra

Anonymous ACL submission

Abstract

The rapid advancements of large language models (LLMs) have made it very difficult to distinguish human-written text from LLM-generated content, raising serious concerns about misinformation, academic integrity, and authenticity. We propose SPECTRAL, a lightweight detector that uses simple features from transformer attention maps. For each attention head, we summarize where attention mass is concentrated and how the corresponding attention graph is structured. The resulting values are converted into a fixed-size feature vector and classified with PCA followed by logistic regression, without modifying the underlying language model. We evaluate SPECTRAL on four benchmarks dataset with six backbone models ranging from GPT-2 to Llama-3.2-3B. SPECTRAL achieves near-perfect Accuracy and F1-score on HC3 and remains strong across datasets, languages, and attack settings. On Arabic and Hindi data, it performs well using multilingual backbone models, without adding language-specific detector features. It is also robust when given text with paraphrasing and adversarial perturbations. These results suggest that attention patterns provide useful signals for distinguishing human-written text from LLM-generated text across different evaluation settings.

1 Introduction

Recent advancements in large language models (LLMs) have substantially improved the quality, alignment, and controllability of LLM-generated text (Radford et al., 2019; Grattafiori et al., 2024; Abdin et al., 2024; Qwen et al., 2025). Modern LLMs are now widely used for writing reports, summaries, paraphrasing, content refinement, and academic content. While these capabilities are useful, they also make it increasingly difficult to distinguish human-written text from LLM-generated text. This raises practical concerns about misinformation, plagiarism, authorship verification, and academic integrity (Solaiman et al., 2019; Wu et al., 2023).

In response, a wide range of LLM-generated text detection methods have been proposed. Probability-based detectors rely on likelihood, entropy, rank (Gehrmann et al., 2019), or perplexity signals from language models (Solaiman et al., 2019). Perturbation-based methods such as DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2024), and DNA-GPT (Yang et al., 2024) examine how model likelihood changes under input modifications. Other approaches use supervised neural classifiers or feature-based systems trained on human and LLM-generated examples (Liu et al., 2019; He et al., 2021; Verma et al., 2024; Hu et al., 2023; Mao et al., 2024; Liu et al., 2023).

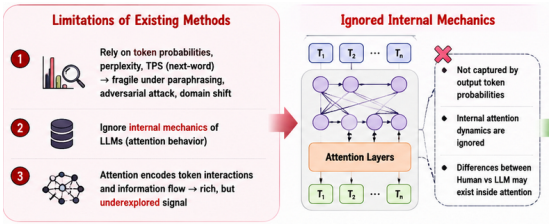

Figure 1: Motivation of SPECTRAL. Existing detectors mainly rely on token probabilities and surface-level features, while ignoring transformer attention dynamics. SPECTRAL leverages attention-based spectral representations to capture structural differences between human-written and LLM-generated text.

Despite significant progress in LLM-generated text detection, existing methods still suffer from several important limitations: (i) Most existing detectors (Luo et al., 2026; Xu et al.; Bao et al.) primarily rely on token probability statistics and perplexity derived from next-word prediction behavior. Although these signals capture surface-level generation patterns, they often fail under adversarial rewriting and unseen generators. (ii) Many supervised detectors (Li et al., 2025; Liu et al., 2023) require large labeled datasets and computationally expensive fine-tuning procedures. (iii) Existing approaches mainly analyze observable textual artifacts while largely ignoring the internal mechanics of transformer language models. In particular, limited work investigates how transformer attention behaves differently when processing human-written versus LLM-generated text.

Motivated by these limitations, we investigate whether the internal attention behavior of frozen transformer models provides discriminative signals for distinguishing human-written and LLM-generated text, as illustrated in Figure 1. In this

work, we study the internal attention structure of frozen transformer language models. We interpret attention maps as weighted graphs, where tokens are nodes and attention weights define edges (Vaswani et al., 2017; von Luxburg, 2007). Instead of relying on token likelihood scores or fine-tuning a neural detector, we extract compact spectral features from graph Laplacian representations of attention maps to capture structural differences between human-written and LLM-generated text.

We propose SPECTRAL, a lightweight detector based on attention-derived spectral representations. For each input, it extracts attention maps from an LLM, computes spectral features across layers and heads, and trains a logistic regression classifier. Experiments across multiple LLM-generated text benchmarks show that SPECTRAL achieves strong Accuracy and F1-score across different backbone models and remains effective under multilingual, paraphrased, and adversarial settings. Figure 2 visualizes the extracted GPT-J feature maps for representative RAID human-written and LLM-generated examples. The heatmaps show normalized feature magnitudes rather than token probabilities or statistical p-values. They illustrate that the spectral representation can expose different layer-head activation patterns for human and LLM-generated text. Our contributions are as follows:

- We propose SPECTRAL, a lightweight detector for LLM-generated text that uses spectral features extracted from transformer attention maps.
- We show that attention-derived spectral features provide a strong signal across four benchmarks and six backbone models, achieving near-perfect performance on several datasets.
- We conduct ablation studies showing how performance changes with context length, number of retained spectral features, and the use of single-layer versus all-layer representations.
- We evaluate robustness and generalization, showing that SPECTRAL remains effective under paraphrasing and adversarial perturbations and works well with multilingual backbones on Arabic and Hindi data.

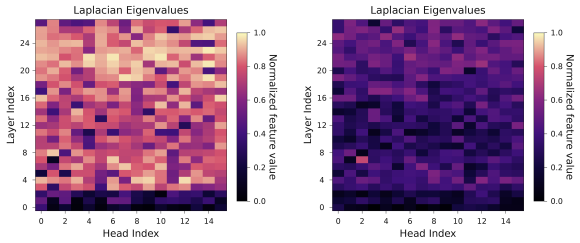

Figure 2: Representative GPT-J Laplacian eigenvalue heatmaps for a human-written RAID example (left) and an LLM-generated RAID example (right). Each panel visualizes layer-head feature magnitudes from the SPECTRAL pipeline, with color denoting normalized feature value. These plots show extracted Laplacian spectral structure for individual samples; they do not represent token probabilities or p-values.

2 Related Work

LLM-generated Text Detection: Existing approaches can generally be grouped into probability-based, perturbation-based, watermarking, and supervised classification methods (Wu et al., 2023). Early methods relied on language-model likelihood, entropy, rank statistics, and perplexity-based signals to separate human-written and generated text (Gehrmann et al., 2019; Solaiman et al., 2019). Methods such as DetectGPT, Fast-DetectGPT, and DNA-GPT further explored curvature and perturbation properties of generated text distributions (Mitchell et al., 2023; Bao et al., 2024; Yang et al., 2024). Although effective in some settings, these approaches often degrade under paraphrasing, domain shift, or unseen generators. Another line of work uses supervised neural classifiers trained directly on human-written and LLM-generated text. RoBERTa- and DeBERTa-based detectors have shown strong performance on benchmark datasets (Liu et al., 2019; He et al., 2021), but their effectiveness often depends on the availability of large labeled datasets and may decrease when evaluated on generators or domains not seen during training. More recent systems such as Ghostbuster, Radar, RAIDAR, and CoCo combine handcrafted features with neural representations to improve robustness (Verma et al., 2024; Hu et al., 2023; Mao et al., 2024; Liu et al., 2023). **Internal Representations of Transformers:** Several recent works have also studied internal representations of transformer models for downstream analysis tasks. Attention maps have been explored for interpretability and model analysis, while probing studies show that

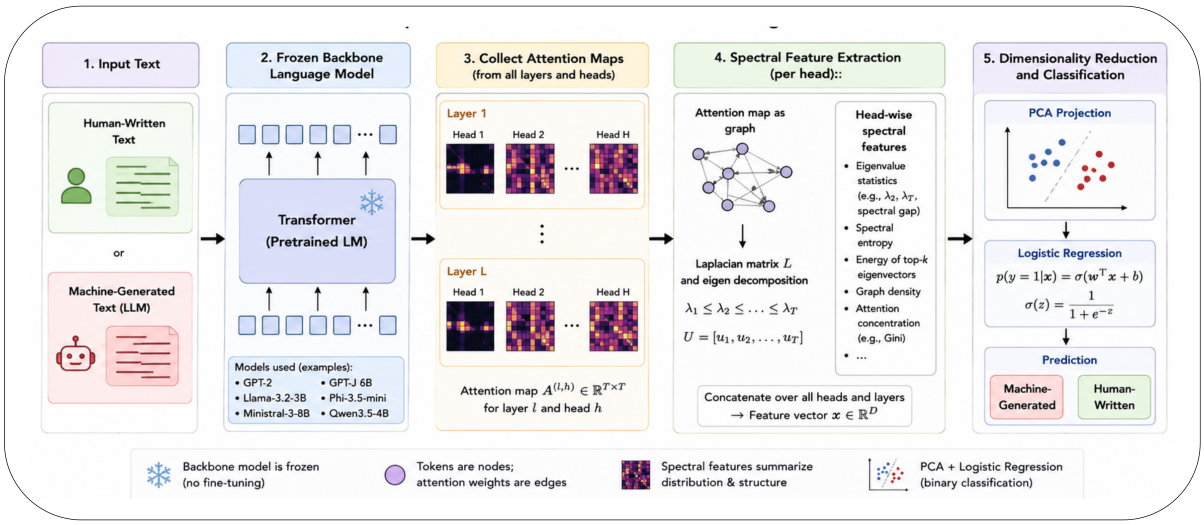

Figure 3: Overview of the proposed SPECTRAL framework. Given an input text, a frozen transformer language model extracts attention maps from all layers and heads. Each attention map is interpreted as a weighted graph, from which Laplacian-based spectral features are computed. The extracted features are concatenated across layers and heads, projected using PCA, and finally classified using logistic regression to distinguish human-written text from LLM-generated text.

transformer representations encode linguistic and structural information (Clark et al., 2019; Rogers et al., 2020; Tenney et al., 2019). Prior studies also caution that attention weights are not always faithful explanations of model decisions (Jain and Wallace, 2019). Inspired by this observation, our work investigates whether compact spectral summaries derived from attention graphs can serve as reliable indicators for distinguishing human-written and LLM-generated text.

Unlike prior detectors that primarily rely on output probabilities or surface textual statistics, SPECTRAL focuses on the structural organization of transformer attention patterns.

3 Methodology

In this section, we describe the proposed SPECTRAL framework for detecting LLM-generated text using attention-derived spectral representations. An overview of the complete pipeline is shown in Figure 3.

Input Text For an input sequence x of length T, the goal is to extract structural information from the way a pretrained transformer attends over the tokens.

Backbone Language Model We pass each x through a pretrained transformer language model and keep the backbone frozen throughout the pipeline. The frozen backbone is used only to extract internal attention structure during inference.

Attention Map Collection Given an input sequence x of length T, let $A^{(l,h)} \in R^{T\times T}$ denote the attention map from layer $l \in \{1, \ldots, L\}$ and attention head $h \in \{1, \ldots, H\}$. The attention matrix contains token-to-token attention scores produced during inference. Because transformer attention is normalized by a softmax, each row of $A^{(l,h)}$ sums to one. For causal language models, the matrix is also lower triangular, since each token can only attend to previous tokens and itself. We collect these attention maps from all layers and heads of the frozen backbone.

Spectral Feature Extraction Following the graph view of attention (Vaswani et al., 2017; von Luxburg, 2007), we interpret each attention map $A^{(l,h)}$ as a weighted adjacency matrix over tokens. Each token is treated as a node, and each attention score is treated as an edge weight. For each layer and head, we construct a graph Laplacian from the attention map:

$$L^{(l,h)} = D^{(l,h)} - A^{(l,h)}, \tag{1}$$

where $D^{(l,h)}$ is a diagonal degree matrix. Since attention maps define directed graphs, we use the out-degree form that measures how much attention each token receives from later tokens. To reduce dependence on sequence length, we normalize this degree by the number of following tokens:

$$d_{ii}^{(l,h)} = \frac{\sum_{u=i}^{T-1} a_{ui}^{(l,h)}}{T-i}, \tag{2}$$

where i and u index token positions. This produces

a Laplacian representation that summarizes how attention flows through the sequence.

We then extract spectral features from each $L^{(l,h)}$. Because the Laplacian remains triangular under causal masking, its eigenvalues correspond to its diagonal entries. We sort these values:

$$\tilde{z}^{(l,h)} = sort\left(diag\left(L^{(l,h)}\right)\right). \quad (3)$$

For each layer and head, we keep the top-k values and concatenate them across all layers and heads:

$$z = l \in L,\ h \in H concat\left[\tilde{z}_T^{(l,h)}, \tilde{z}_{T-1}^{(l,h)}, \ldots, \tilde{z}_{T-k}^{(l,h)}\right]. \quad (4)$$

Here, k controls the number of spectral features retained per layer-head pair. The resulting vector $z \in R^{L \cdot H \cdot k}$ represents the attention-derived spectral structure of the input text.

Dimensionality Reduction and Classification The concatenated spectral vector can be high-dimensional because it includes features from all layers and heads. We therefore project z using PCA (Jolliffe, 2002) (Appendix Sec. C.1) to obtain a compact representation. The projected features are then passed to a logistic regression classifier, which predicts whether the input is human-written or LLM-generated. The detector is lightweight because only PCA and logistic regression are trained; the backbone language model remains frozen.

4 Experimental Setup

4.1 Dataset

We evaluate SPECTRAL on existing human-written versus LLM-generated text detection benchmarks. Specifically, we use HC3 (Guo et al., 2023), RAID (Dugan et al., 2024), M4 (Wang et al., 2024), and MAGE (Li et al., 2024), each containing human-written and LLM-generated examples. Table 1 summarizes the sample counts and train/validation/test splits used in our experiments.

4.2 Baselines

We compare SPECTRAL against several representative baselines from different categories of LLM-generated text detection. These include probability-based approaches such as Likelihood (Gehrmann et al., 2019), LogRank (Gehrmann et al., 2019), Entropy (Gehrmann et al., 2019), and NPR, perturbation-based methods including Detect-GPT (Mitchell et al., 2023), DNA-GPT (Yang et al., 2024), and Fast-DetectGPT (Bao et al., 2024), as well as supervised detectors such as RoBERTa (Liu et al., 2019). We also compare against recent competitive systems including RAIDAR (Mao et al., 2024) and CoCo (Liu et al., 2023)(AppendixA).

Dataset	Human	LLM	Train	Validation	Test
HC3	20k	20k	16k/class	2k/class	2k/class
RAID	20k	20k	16k/class	2k/class	2k/class
MAGE	20k	20k	16k/class	2k/class	2k/class
M4	20k	20k	16k/class	2k/class	2k/class

Table 1: Dataset sample counts and splits used in the main experiments

4.3 Experiment Setup

We evaluate six backbone models: GPT-2 (Radford et al., 2019), GPT-J 6B (Wang and Komatsuzaki, 2021), Llama-3.2-3B (Grattafiori et al., 2024), Phi-3.5-mini (Abdin et al., 2024), Ministral-3-8B (Mistral AI, 2025), and Qwen3.5-4B (Qwen et al., 2025). For each backbone, we extract attention-based features for every example and train a PCA plus logistic regression classifier on the training split. Hyperparameter selection is performed on the validation split, and final performance is reported on the held-out test split. We report Accuracy and F1-score. For multilingual evaluation, we use multilingual backbone models for Arabic (Al-Shaibani and Ahmed, 2025) and Hindi data (Kavathekar et al., 2024), without adding language-specific detector features or separate language-specific classifiers. The final classifier is a scikit-learn logistic regression model trained with the binary cross-entropy loss function and L_2 regularization. We use balanced class weights, a maximum of 2000 solver iterations, and a fixed random seed for reproducibility. Because the detector uses classical logistic regression, it is not trained for a fixed number of epochs; training stops when the optimizer converges or reaches the iteration limit. Detail present in Appendix Section. A and experiments B.

5 Results

Table 2 reports the performance of SPECTRAL across four datasets and six backbone models. SPECTRAL achieves consistently high Accuracy and F1-score on HC3, RAID, and M4 for every backbone on these datasets. The strongest results are obtained with Llama-3.2-3B and Phi-3.5-mini, which reach near-perfect Accuracy and F1-score on HC3 and RAID while also performing strongly on M4. These results indicate that compact attention features remain informative across model families and parameter scales, rather than depending on a

Category	Model	HC3		RAID		M4		MAGE	
		ACC	F1	ACC	F1	ACC	F1	ACC	F1
Zero-shot	Likelihood	96.8	96.8	69.9	69.9	79.1	79.1	61.1	61.1
	LogRank	98.0	98.0	71.0	71.0	77.1	77.1	62.2	62.2
	Entropy	93.0	93.0	63.9	63.9	65.6	65.6	61.2	61.2
	NPR	56.4	56.4	52.2	52.2	57.4	57.4	52.6	52.6
Perturbation	DetectGPT	68.2	67.1	50.1	41.2	51.2	40.4	52.4	51.0
	DNA-GPT	80.8	80.8	65.8	65.8	64.8	64.8	53.4	53.4
	Fast-DetectGPT	49.4	33.1	49.5	33.1	50.0	33.3	50.0	33.3
Supervised	RAIDAR	53.4	53.4	44.5	44.5	46.9	46.9	49.1	49.1
	RoBERTa	95.0	95.0	76.5	75.7	76.8	75.8	69.1	68.0
	CoCo	46.6	46.6	61.4	61.4	71.7	71.7	53.8	53.8
SPECTRAL	GPT-2	97.05	97.06	94.25	94.28	91.92	91.89	66.70	67.58
	GPT-J-6B	99.60	99.60	99.70	99.70	98.18	98.17	76.27	77.68
	Llama-3.2-3B	**99.93**	**99.93**	**99.80**	**99.80**	98.41	98.41	**82.99**	**83.69**
	Phi-3.5-mini	99.85	99.85	99.75	99.75	**98.95**	**98.95**	82.30	82.00
	Ministral-3-8B	99.73	99.72	99.40	99.40	94.18	94.12	82.04	82.81
	Qwen3.5-4B	99.08	99.07	98.55	98.55	93.05	93.00	71.54	72.99

Table 2: Performance comparison of different categories of AI-generated text detectors across HC3, RAID, M4, and MAGE datasets. We compare zero-shot, perturbation-based, and supervised baselines against the proposed SPECTRAL framework with different backbone models. Results are reported using Accuracy (ACC) and F1-score.

single architecture. MAGE is the most challenging benchmark in our evaluation. Performance remains above chance for all backbones, but the gap between MAGE and the other datasets is substantial, especially for GPT-2 and Qwen3.5-4B. This suggests that MAGE contains generation patterns or domain variation that are less easily captured by the current feature representation. Nevertheless, larger and more recent multilingual models, such as Llama-3.2-3B, Phi-3.5-mini, and Ministral-3-8B, substantially improve over GPT-2 on this dataset. Overall, the results show that attention-derived features provide a strong signal for human versus LLM text detection, while also highlighting MAGE as an important stress test for future analysis.

5.1 Robustness Analysis

Cross-Dataset Evaluation: We train SPECTRAL on one dataset and evaluate it on the others using GPT-J 6B. Table 3 reports F1-score for each train-test dataset pair, making the strong in-domain diagonal performance and weaker off-domain transfer easier to compare. The diagonal entries correspond to in-domain evaluation, while the off-diagonal entries measure transfer to unseen datasets. The results show strong transfer between HC3 and RAID, but weaker transfer to M4 and MAGE in several settings. This suggests that dataset-specific generation style and domain composition affect how well attention-derived spectral features generalize across benchmarks.

Out-of-Distribution Evaluation: We also evaluate the GPT-J 6B detector trained on RAID on a separate Reddit human-versus-LLM dataset (Qazi et al., 2024). For each generator subset, the evaluation contains 150 human-written and 150 LLM-generated Reddit examples. Table 5 reports the resulting Accuracy and F1-score for each generator. The results show strong transfer for several modern generators, including Claude-3 Haiku, GPT-4o, and GPT-4 Turbo, while performance is lower for some older open-source generators. This suggests that the RAID-trained detector captures useful Reddit-domain signals, but generalization still varies across the source model used to generate the text.

Train/Test	HC3	RAID	M4	MAGE
HC3	0.9980	0.8884	0.7236	0.6217
RAID	0.9081	0.9975	0.6441	0.6171
M4	0.7071	0.3807	0.9822	0.6535
MAGE	0.8020	0.6496	0.7028	0.7706

Table 3: Cross-dataset evaluation of SPECTRAL using GPT-J 6B. Each row denotes the dataset used for training, while each column denotes the dataset used for testing. Results are reported using F1-score.

Robustness to Attack Types: We assess whether SPECTRAL remains reliable when LLM-generated text is modified after generation. This setting is important because users may paraphrase, perturb, or adversarially edit LLM outputs before detection. Table 4 reports results for GPT-J 6B on 20k RAID samples per class under three attack types.

Attack Type	Accuracy	F1
Perturbation	99.40	99.40
Adversarial	96.95	96.95
Paraphrasing	95.20	95.20

Table 4: Robustness of GPT-J 6B to different attack types on RAID using 20k samples per class.

Metric	Claude3-Haiku	GPT-4o	GPT-4-Turbo	Llama3-8B	Gemma-7B
Accuracy	0.9267	0.8467	0.8433	0.7000	0.6200
F1-score	0.9252	0.8516	0.8469	0.6786	0.6667

Table 5: Out-of-domain Reddit evaluation using the GPT-J-6B-based SPECTRAL detector trained on the RAID dataset. Each Reddit generator subset contains 150 human-written and 150 LLM-generated samples. Results are reported using Accuracy and F1-score.

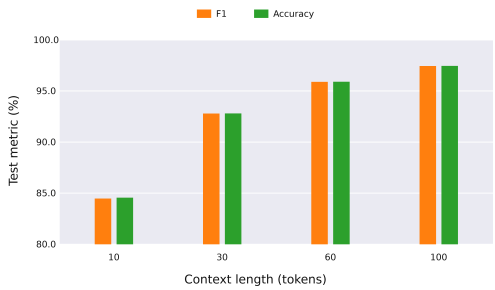

Figure 4: Context length ablation on the RAID dataset for GPT-J 6B. We report Accuracy and F1-score only. Performance improves steadily as the available context increases from 10 to 100 tokens.

SPECTRAL remains strong across all three settings. Performance is highest under perturbation, where the surface form changes but much of the original generation structure is still preserved. Adversarial editing and paraphrasing are more challenging because they can alter lexical choices and sentence structure, reducing some of the visible cues used by the detector. Even in these harder cases, the detector keeps high Accuracy and F1, suggesting that attention-based spectral features capture signals that are not removed by simple rewriting. Among the three attacks, paraphrasing gives the lowest scores, which is expected because it most directly changes the wording while keeping the same meaning.

6 Ablation Studies

(i) How does context length affect performance? We investigate the impact of context window size on model performance using GPT-J 6B on the RAID dataset. Since SPECTRAL derives its features from attention maps, very short inputs may provide only limited relational structure between tokens. Longer contexts can expose more token-to-token interactions, making both the attention diagonal and graph-based summaries more informative. Figure 4 presents results for four context lengths: 10, 30, 60, and 100 tokens.

Results demonstrate consistent gains with increased context. Accuracy improves from 84.55% at 10 tokens to 97.45% at 100 tokens, while F1-score increases from 84.46 to 97.44. The improvement in both reported metrics suggests that additional context provides more informative attention structure for separating human-written and LLM-generated examples. This supports the intuition that attention-based features benefit from a minimum amount of surrounding context, because longer inputs allow the model's attention patterns to reflect broader syntactic and semantic dependencies. At the same time, performance continues to improve up to 100 tokens in this setting, so we treat this as evidence that context length matters rather than as a complete characterization of the optimal context size.

6.1 How does the number of eigenvalues influence performance?

We next study how the number of spectral features affects performance. In SPECTRAL, the value of k controls how many top spectral features are kept from each layer-head feature representation. A smaller k produces a compact feature vector, while a larger k preserves more information from the attention structure but also increases feature dimensionality. For the main results in Table 2, we use $k = 5$ for all reported models and datasets. We test different top-k eigenvalue settings, $k \in \{3, 5, 10, 25, 50, 100\}$, on the RAID dataset using GPT-J 6B. This allows us to measure how much performance changes as more spectral features are retained. Figure 5 shows that performance improves as k increases. At $k = 3$, the feature vector is too small to capture enough attention structure. Moving from $k = 3$ to $k = 5$ improves both Accuracy and F1. The $k = 5$ setting is the value used in Table 2 for GPT-J 6B on RAID. Further increases continue to help both reported metrics. At $k = 100$, Accuracy and F1 reach 99.78.Detail analysis present in Appendix Section B.1 Table 7.

The trend suggests that additional spectral features contain useful signal for detection, especially when moving from very small values of k to moderate values. However, the improvement becomes smaller at larger k as the additional features start to

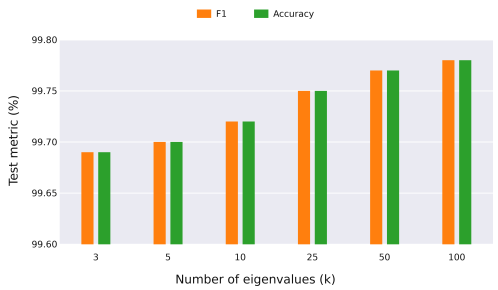

Figure 5: Performance across different values of k, where $k \in \{3, 5, 10, 25, 50, 100\}$, on the RAID dataset using GPT-J 6B. Larger k values show consistent gains.

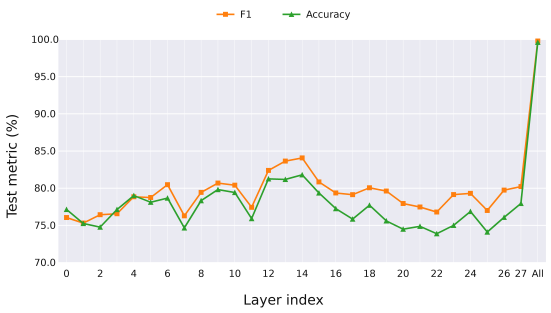

Figure 6: Layer ablation on RAID using GPT-J 6B with $k = 5$. We show Accuracy and F1-score only. Combining all layers gives substantially stronger performance than using any single layer alone.

add less new information. For example, Accuracy improves by 0.03 points from $k = 10$ to $k = 25$, but only by 0.01 points from $k = 50$ to $k = 100$. This indicates diminishing returns: after a certain point, adding more features still helps, but the benefit is much smaller compared to the added cost of computation and storage. We therefore use $k = 5$ in the main experiments as a compact setting that keeps the detector lightweight and consistent across all datasets, while noting that the ablation shows larger values of k can be useful.

6.2 Does using all layers at once improve performance?

We also evaluate whether it is recommended to use features from a single transformer layer or to combine features from all layers. This ablation uses GPT-J 6B on RAID with the same $k = 5$ setting used in Table 2. Each individual layer provides a much smaller feature vector, while the all-layer setting concatenates the spectral features extracted across the model. Figure 6 shows that no individual layer is sufficient to match the all-layer representation. The best single layer is layer 14, which reaches 81.89 Accuracy and 83.98 F1. In contrast, using all layers reaches 99.70 Accuracy and 99.70 F1, which is the same GPT-J 6B result reported for RAID in Table 2. This large gap suggests that the detection signal is distributed across the network rather than concentrated in one layer. Lower and higher layers may capture different kinds of attention structure, and combining them gives the classifier a more complete view of the text. Therefore, we use all layers in the main experiments.

6.3 How does SPECTRAL generalize?

We evaluate whether SPECTRAL remains effective beyond the English benchmarks used in the main results. For this experiment, we use multilingual backbone models and keep the detector design unchanged: attention-based spectral features are extracted, projected with PCA, and classified with logistic regression. For Arabic, we use the Arabic AI Fingerprint dataset (Al-Shaibani and Ahmed, 2025). For Hindi, we use the Counter Turing Test dataset (Kavathekar et al., 2024). We do not add language-specific features or train separate detector architectures for each language.

Language	Model	Accuracy	F1
Arabic	Phi-3.5-mini	94.67	94.69
Arabic	Qwen-3.5-4B	95.00	95.06
Arabic	Ministral-8B	96.89	96.90
Hindi	Llama-3.2-1B	100.00	100.00

Table 6: Multilingual evaluation of SPECTRAL. Scores are reported on held-out test splits.

Table 6 shows that SPECTRAL performs strongly on both Arabic and Hindi. On Arabic, all three multilingual backbones achieve strong Accuracy and F1-score, with Ministral-8B giving the best overall result. On Hindi, the Llama-3.2-1B run reaches 100.00 Accuracy and F1-score on the held-out test split. These results suggest that the detector is not limited to English-only surface patterns. At the same time, we interpret this as multilingual generalization of the detector features, not as evidence that the backbone models are language-agnostic: the evaluated backbones have multilingual training exposure. The important point is that SPECTRAL does not require language-specific detector features to work well in these settings.

6.4 Computational Efficiency

SPECTRAL is lightweight at the classifier stage, since it trains only PCA followed by logistic regression while keeping the backbone language model frozen. However, the feature extraction stage is

Dataset	Main Error	Observation
HC3	False positives	Wiki-style human answers are often predicted as LLM-generated.
RAID	Creative-domain errors	Poetry remains the hardest domain across backbone models.
M4	Generator/domain sensitivity	Davinci, Dolly, WikiHow, arXiv, and ELI5 subsets are harder to detect.
MAGE	Severe domain shift	Short summaries, paraphrases, and GPT-4-style generations produce the highest error rates.

Table 7: Summary of error patterns across datasets.

computationally expensive. Extracting attention maps requires a forward pass through the backbone model for every input, and the cost grows with the number of layers and attention heads.

For a sequence of length T, computing the spectral features is dominated by constructing the degree values from the attention matrix, which requires aggregating over the $T \times T$ attention map. This gives an $O(T^2)$ cost per layer-head pair. Sorting the resulting diagonal values adds an $O(T \log T)$ term. Therefore, the total feature extraction cost scales with the number of layers, heads, and input tokens. The storage cost also increases with the number of retained spectral values k, since the final feature vector has dimension $L \cdot H \cdot k$. Larger k can improve performance, as shown in our ablation study, but it also increases storage and downstream processing cost. In practice, the main bottleneck is attention extraction rather than logistic regression training. For this reason, SPECTRAL is best viewed as lightweight after feature extraction, but not free: it trades expensive neural fine-tuning for frozen-model attention extraction and compact classical classification.

6.5 Error Analysis

Table 7 summarizes the major failure patterns observed across datasets. HC3 and RAID are the easiest benchmarks for SPECTRAL, where most backbone models achieve near-perfect performance. M4 is moderately harder, while MAGE is substantially more challenging and exposes the main weaknesses of the detector. Across backbone models, GPT-2 consistently performs worst, suggesting that smaller or older language models provide less informative attention representations for spectral detection. In contrast, Llama-3.2-3B and Phi-3.5-mini achieve the strongest and most stable performance across datasets. These trends indicate that stronger backbone attention structures generally improve robustness. Dataset-specific analysis reveals distinct error modes. On RAID, remaining errors are concentrated in creative domains such as poetry, where generated text is frequently classified as human-written. On HC3, false positives mainly occur in structured human-written domains, especially wiki-style answers. For M4, the detector struggles more with Davinci-, Dolly-, and Dolly2-generated outputs, along with WikiHow, arXiv, and Reddit ELI5 examples. MAGE is the most difficult benchmark and behaves differently from the other datasets. Most errors are false positives on short and constrained human-written summaries or continuations, while paraphrased outputs and GPT-4-style generations often produce false negatives because their spectral patterns resemble human-written text. We also observe a strong length effect. Overall, the analysis shows that SPECTRAL is highly effective on standard AI-text detection benchmarks, but remains sensitive to short-form human writing, paraphrased generations, and outputs from stronger modern LLMs.

7 Conclusions

We proposed SPECTRAL, a lightweight detector for distinguishing human-written and LLM-generated text using spectral features derived from transformer attention maps. Instead of relying on surface-level textual statistics or full neural fine-tuning, SPECTRAL extracts compact graph-based representations from frozen transformer attention patterns and uses PCA with logistic regression for classification. Experiments across diverse benchmarks demonstrate that attention-derived spectral features provide a strong detection signal across multiple backbone models and evaluation settings. The proposed method achieves near-perfect performance on several datasets while remaining computationally lightweight compared to fully fine-tuned neural detectors. Additional ablation studies further show that performance benefits from larger context windows, richer spectral representations, and the use of features from all transformer layers. We also demonstrate that SPECTRAL generalizes across multilingual settings and remains robust under paraphrasing, perturbation, and adversarial rewriting attacks. These findings suggest that transformer attention structure contains meaningful information. Overall, the proposed framework highlights the effectiveness of attention spectral representations as a promising direction for robust and lightweight LLM-generated text detection. In future work we will make training free and also other internal evaluation.

Limitations

Despite strong performance across multiple benchmarks, SPECTRAL has several limitations. First, the detector still depends on access to internal transformer attention maps, which may not always be available for closed-source or API-only language models. Second, performance on challenging datasets such as MAGE remains lower than on HC3 or RAID, suggesting that certain generation styles and domains are more difficult to capture using the current feature representation. Third, although the detector is robust to several rewriting attacks, heavy semantic rewriting or human post-editing may reduce the structural differences between human-written and LLM-generated text.

In addition, extracting attention maps from large backbone models can introduce computational overhead during feature extraction, even though the downstream classifier itself remains lightweight. Finally, the current study focuses on binary detection and does not explicitly identify the source generator model or handle mixed human-AI collaborative writing.

8 Ethical Considerations

SPECTRAL is intended to support AI-generated text detection for applications such as academic integrity and misinformation analysis. However, the detector may produce false positives or false negatives, particularly for paraphrased or short-form text. Therefore, it should not be used as the sole basis for high-stakes decisions. The framework relies on pretrained transformer attention maps and is designed for research purposes only. All experiments were conducted using publicly available datasets and pretrained models without collecting personally identifiable information.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Maged S. Al-Shaibani and Moataz Ahmed. 2025. The arabic AI fingerprint: Stylometric analysis and detection of large language models text. *arXiv preprint arXiv:2505.23276*.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-detectgpt: Efficient zero-shot detec-tion of machine-generated text via conditional probability curvature.

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 (ICLR)*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

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.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. 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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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. *arXiv preprint arXiv:2301.07597*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: Robust AI-text detection via adversarial learning. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, New Orleans, LA, USA.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3543–3556.

Ian T. Jolliffe. 2002. *Principal Component Analysis*, 2nd edition. Springer Series in Statistics. Springer, New York.

Ishan Kavathekar, Anku Rani, Ashmit Chamoli, Ponnurangam Kumaraguru, Amit Sheth, and Amitava Das. 2024. Counter turing test (ct^2): Investigating ai-generated text detection for hindi – ranking llms based on hindi ai detectability index (adi_{hi}). *Preprint*, arXiv:2407.15694.

Xiang Li, Zhiyi Yin, Hexiang Tan, Shaoling Jing, Du Su, Yi Cheng, Huawei Shen, and Fei Sun. 2025. Prdetect: Perturbation-robust llm-generated text detection based on syntax tree. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 8290–8301.

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.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2023. CoCo: Coherence-enhanced machine-generated text detection under low resource with contrastive learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16167–16188, Singapore. Association for Computational Linguistics.

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*.

Haitong Luo, Weiyao Zhang, Suhang Wang, Wenji Zou, Chungang Lin, Xuying Meng, and Yujun Zhang. 2026. Specdetect: Simple, fast, and training-free detection of llm-generated text via spectral analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32356–32364.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: geneRative AI detection viA rewriting. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Mistral AI. 2025. Ministral 3. *arXiv preprint arXiv:2601.08584*.

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*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pages 8024–8035.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Naveed Qazi, William Shiao, and Evangelos E. Papalexakis. 2024. GRiD: The GPT-Reddit dataset. *arXiv preprint arXiv:2403.07321*.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, and 1 others. 2025. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Technical Report*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

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. *arXiv preprint arXiv:1908.09203*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5998–6008.

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.

Ulrike von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 billion parameter autoregressive language model. https://github.com/kingoflolz/mesh-transformer-jax.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: Multi-generator, multi-domain, and multilingual 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

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*.

Yihuai Xu, Yongwei Wang, Yifei Bi, Huangsen Cao, Zhouhan Lin, Yu Zhao, and Fei Wu. Training-free llm-generated text detection by mining token probability sequences.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Petzold, William Yang Wang, and Haifeng Chen. 2024. DNA-GPT: Divergent N-gram analysis for training-free detection of GPT-generated text. In *The Twelfth International Conference on Learning Representations (ICLR)*.

A Details

Our experiments were implemented using HuggingFace Transformers, PyTorch, and scikit-learn. We used HuggingFace models to run backbone LLM inference and collect attention maps. The SPECTRAL detector was trained using PCA followed by logistic regression from scikit-learn. We used standard Python data-processing tools such as NumPy and Pandas for feature handling and result aggregation, and plotting libraries for visualizations.

For all datasets, we used the benchmark-provided labels. Each dataset was treated as a binary classification task between human-written and LLM-generated text. Unless otherwise stated, we used 20k samples per class for each dataset and split them into train, validation, and test sets using an 80/10/10 ratio.

Attention-map extraction was performed with GPU acceleration because it requires running the backbone transformer models over all examples. After attention features were extracted and saved, PCA and logistic regression training were run as lightweight classical LLM-learning steps. The backbone models were kept frozen throughout all experiments; only the downstream detector was trained. For multilingual experiments, we used multilingual backbone models for Arabic and Hindi data while keeping the same detector architecture and feature extraction procedure.

B Experimental Details

We used six frozen backbone LLMs for attention-feature extraction: Llama-3.2-3B (Grattafiori et al., 2024)[1], GPT-2 (Radford et al., 2019)[2], GPT-J 6B (Wang and Komatsuzaki, 2021)[3], Phi-3.5-mini-instruct (Abdin et al., 2024)[4], Ministral-3-8B-Instruct-2512 (Mistral AI, 2025)[5], and Qwen3.5-4B (Qwen et al., 2025)[6].

Our experiments were implemented using HuggingFace Transformers, PyTorch, and scikit-learn (Wolf et al., 2020; Paszke et al., 2019; Pedregosa et al., 2011). We used HuggingFace models to run backbone LLM inference and collect attention maps. The SPECTRAL detector was trained using PCA followed by logistic regression from scikit-learn (Jolliffe, 2002; Pedregosa et al., 2011). We used standard Python data-processing tools such as NumPy and Pandas for feature handling and result aggregation, and plotting libraries for visualizations.

For all datasets, we used the benchmark-provided labels and did not construct new labels with a judge model. Each dataset was treated as a binary classification task between human-written and LLM-generated text. Unless otherwise stated, we used 20k samples per class for each dataset and split them into train, validation, and test sets using an 80/10/10 ratio.

[1] https://huggingface.co/meta-llama/Llama-3.2-3B
[2] https://huggingface.co/openai-community/gpt2
[3] https://huggingface.co/EleutherAI/gpt-j-6b
[4] https://huggingface.co/microsoft/Phi-3.5-mini-instruct
[5] https://huggingface.co/mistralai/Ministral-3-8B-Instruct-2512
[6] https://huggingface.co/Qwen/Qwen3.5-4B

Attention-map extraction was performed with GPU acceleration because it requires running the backbone transformer models over all examples. After attention features were extracted and saved, PCA and logistic regression training were run as lightweight classical LLM-learning steps. The backbone models were kept frozen throughout all experiments; only the downstream detector was trained. For multilingual experiments, we used multilingual backbone models for Arabic and Hindi data while keeping the same detector architecture and feature extraction procedure.

A Baseline Implementation Details

We compare proposed against eleven representative AI-generated text detectors under both binary setups: Human vs LLM-generated and Cross-LLM detection. All baselines were evaluated on the same train–test splits and preprocessing settings.

Zero-shot methods: Likelihood (Gehrmann et al., 2019), LogRank (Gehrmann et al., 2019), and Entropy (Gehrmann et al., 2019) were implemented using pretrained language-model scoring functions without supervised training.

Perturbation-based methods: DetectGPT (Bao et al., 2024), Fast-DetectGPT (Bao et al.), NPR, and DNA-GPT (Yang et al., 2024) were implemented following their original perturbation and probability-based detection strategies.

Adversarial and supervised methods: RAIDAR (Mao et al., 2024) detects AI-generated text through LLM-based rewriting and edit-distance comparison, while RoBERTa (Liu et al., 2019) was fine-tuned as a supervised transformer baseline for binary classification.

Graph-based methods: CoCo (Liu et al., 2023) constructs entity co-occurrence graphs with contrastive learning, whereas our proposed RST-GCN models discourse dependency structures using RST-based graphs and graph convolutional networks.

B Extended Results

B.1 Best Found Hyperparameters

We report the hyperparameter settings used for the main SPECTRAL results and summarize the trends observed in the ablation studies. The main experiments use $k = 5$ retained spectral values per layer-head pair, PCA for dimensionality reduction, logistic regression for classification, and features from all transformer layers. This setting was chosen as a compact configuration that works consistently across datasets and backbone models.

The top-k ablation in Table 7 shows that increasing k generally improves performance on RAID across the six backbone models. However, the gains become small after moderate values of k. For example, performance improves only marginally between $k = 50$ and $k = 100$, while the feature dimension and storage cost continue to grow linearly with k. We therefore use $k = 5$ in the main experiments to keep the representation lightweight, while noting that larger values of k can provide small additional gains when computational and storage budgets allow.

The layer ablation shows that using all layers is substantially stronger than using any single layer alone. Figure 8 shows this trend on RAID across all six backbone models. Although some middle layers perform better than others, no individual layer matches the all-layer representation. This suggests that the detection signal is distributed across the transformer rather than concentrated in a single depth. For this reason, all main experiments concatenate spectral features from every layer and attention head. Together, the top-k and layer ablations support the main configuration: $k = 5$ with features concatenated from all layers.

C Why PCA?

C.1 Effect of PCA Dimensionality Reduction

We evaluate the importance of PCA in the SPECTRAL pipeline by comparing classification performance with and without dimensionality reduction. PCA helps compress redundant spectral representations while preserving the most informative structural components.

Experiments on the RAID dataset using GPT-J 6B show that PCA consistently improves both stability and classification performance by reducing feature redundancy across layers and attention heads. Without PCA, the classifier becomes more sensitive to high-dimensional noise and correlated spectral features.

These results suggest that PCA provides an effective compact representation for attention-derived spectral features while maintaining strong discriminative capability.

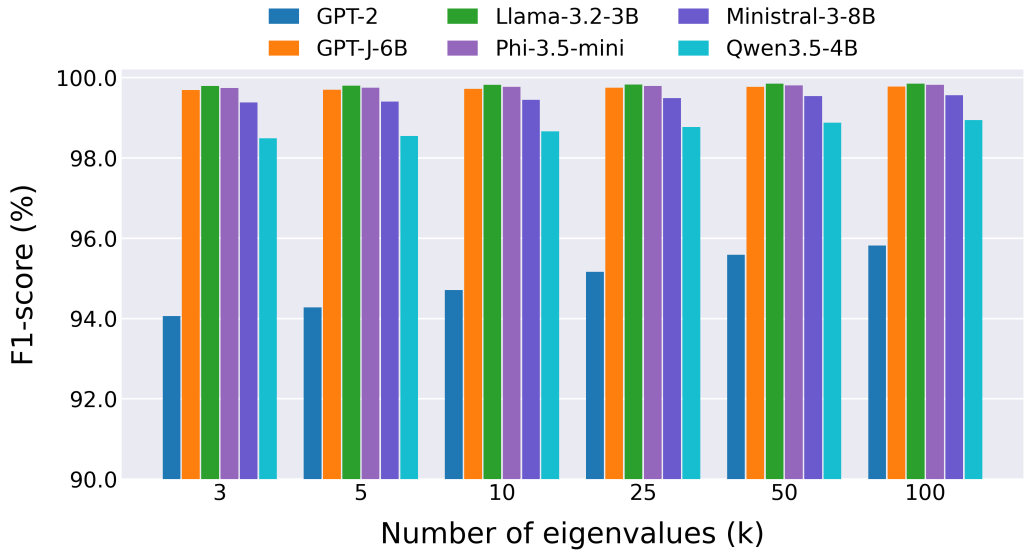

Figure 7: F1-score across different numbers of retained eigenvalues on RAID for all six backbone models. The trends are consistent with the Accuracy results, showing diminishing returns as k increases.

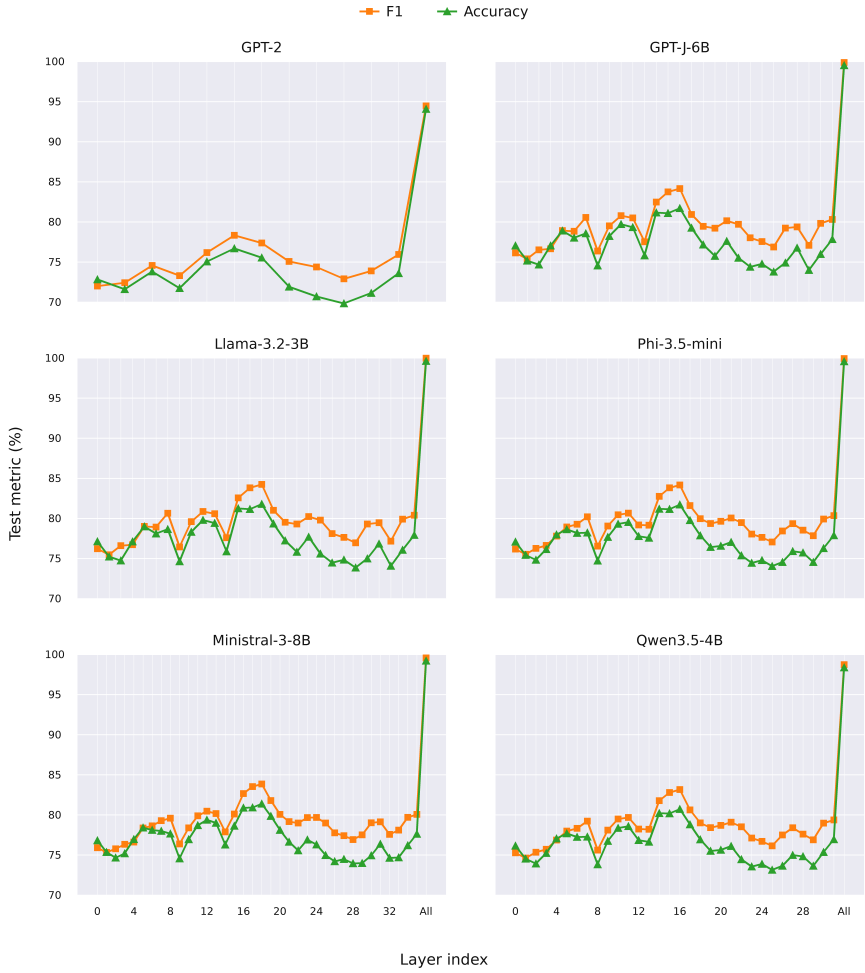

Figure 8: Layer ablation on RAID for all six backbone models with $k = 5$. We report F1-score and Accuracy for each individual layer and the all-layer representation.