SpecDetect: Simple, Fast, and Training-Free Detection of LLM-Generated Text via Spectral Analysis

Haitong Luo[1, 2], **Weiyao Zhang**[1], **Suhang Wang**[3], **Wenji Zou**[1, 2], **Chungang Lin**[1, 2], **Xuying Meng**[1, 4*], **Yujun Zhang**[1, 5, 6*]

[1]Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Pennsylvania State University
[4]Purple Mountain Laboratory
[5]Nanjing Institute of InforSuperBah
[6]University of Chinese Academy of Sciences, Nanjing
{luohaitong21s, mengxuying, nrcyujun}@ict.ac.cn

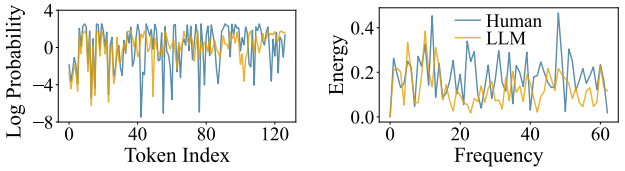
Abstract

The proliferation of high-quality text from Large Language Models (LLMs) demands reliable and efficient detection methods. While existing training-free approaches show promise, they often rely on surface-level statistics and overlook fundamental signal properties of the text generation process. In this work, we reframe detection as a signal processing problem, introducing a novel paradigm that analyzes the sequence of token log-probabilities in the frequency domain. By systematically analyzing the signal's spectral properties using the global Discrete Fourier Transform (DFT) and the local Short-Time Fourier Transform (STFT), we find that human-written text consistently exhibits significantly higher spectral energy. This higher energy reflects the larger-amplitude fluctuations inherent in human writing compared to the suppressed dynamics of LLM-generated text. Based on this key insight, we construct SpecDetect, a detector built on a single, robust feature from the global DFT: DFT total energy. We also propose an enhanced version, SpecDetect++, which incorporates a sampling discrepancy mechanism to further boost robustness. Extensive experiments show that our approach outperforms the state-of-the-art model while running in nearly half the time. Our work introduces a new, efficient, and interpretable pathway for LLM-generated text detection, showing that classical signal processing techniques offer a surprisingly powerful solution to this modern challenge.

Code — https://github.com/luohaitong/SpecDetect

1 Introduction

Recent advancements in Large Language Models (LLMs) have enabled the generation of high-quality text that is often indistinguishable from human writing (Crothers, Japkowicz, and Viktor 2023). While beneficial, this capability poses significant challenges related to potential misuse, including the spread of misinformation (Opdahl et al. 2023; Fang et al. 2024) and threats to academic integrity (Else 2023; Currie 2023), which underscore the urgent need for reliable LLM-generated text detection methods.



(a) Time-Domain Signal(Log-Probability) (b) Frequency-Domain Spectrum

Figure 1: Comparison of a representative human vs. LLM-generated sample (XSum dataset, LLaMA3-8B as source model, GPT-J-6B as proxy model), showing both time-domain signals and the frequency-domain spectrum.

In contrast, training-free methods offer improved robustness by analyzing intrinsic statistical discrepancies. These approaches implicitly operate on the premise that human writing possesses a greater "generative vitality", which is characterized by more dynamic and unpredictable fluctuations in the token probability sequence, than the more constrained output of LLMs. Various methods attempt to model this vitality, evolving from simple global statistics to more sophisticated modeling of the sequence itself, such as treating it as a time series to capture local fluctuations (Xu et al. 2024). Nevertheless, a key limitation of these time-domain methods is their inability to fully capture the essence of this vitality. This is because the raw, token-by-token probability sequence is often noisy and volatile, making it difficult to distinguish fundamental structural patterns from local, stochastic variations. Consequently, these methods often depend on complex, multi-stage feature engineering pipelines that are sensitive to hyperparameters. This highlights the need for an approach that can effectively capture the subtle differences between human and LLM-generated text in a manner that is both conceptually simple and free from sensitive parameterization.

In this work, we diverge from the time domain to propose a novel **frequency-domain paradigm** for LLM-generated text detection. We hypothesize that the essential differ-

*Corresponding Authors

ence between human and machine text lies in the **dynamic range of its token probability fluctuations**, a property we term "generative vitality". We introduce **SpecDetect**(**Spec**tral **Detect**or), a training-free detector that operationalizes this idea by treating the log-probability sequence as a signal and analyzing its spectral properties. The power of a frequency-domain transformation lies in its ability to decompose a signal into its constituent components based on their rate of fluctuation. This process converts the complex time-domain signal into a more structured frequency spectrum, allowing for a precise quantification of its energy and making the hypothesized differences between human and machine text easier to measure. As visualized in Figure 1, this analysis reveals a distinct and consistent separation. The spectrum of human-written text exhibits frequency components with significantly larger magnitudes, which translates to higher overall spectral energy. This provides a direct and robust measure of the greater "generative vitality" inherent in human writing.

Our analysis further reveals that a single, hyperparameter-free feature, the DFT Total Energy, serves as a powerful discriminator. This metric effectively quantifies the suppressed "generative vitality" of LLM text, which stems from the models' inherent constraint of sampling from high-probability token distributions (Holtzman et al. 2019), a stark contrast to the unconstrained nature of human expression. Our approach, grounded in classical signal processing theory, is exceptionally simple and computationally efficient. Extensive experiments validate our method, demonstrating that SpecDetect and its sampling-based extension, SpecDetect++, achieve a superior combination of effectiveness and efficiency. Notably, our method outperforms the latest state-of-the-art model with significantly lower computational cost. Our main contributions are:

- We are the first to robustly capture the fundamental difference between human and LLM-generated text in the frequency domain, providing a more essential perspective than prior time-domain analyses.
- We propose a detector based on a single, hyperparameter-free spectral feature, the DFT Total Energy, that is simpler and faster than current state-of-the-art methods.
- We empirically demonstrate our method sets a new state-of-the-art in the effectiveness-efficiency trade-off, outperforming the prior SOTA at half the runtime.

2 Related Work

2.1 Detection of LLM-Generated Text

Existing detection approaches are broadly categorized into training-based and training-free methods. Training-based detectors learn a classifier (Bhattacharjee et al. 2023; Li et al. 2023; Tian et al. 2023) but often suffer from poor generalization to out-of-distribution data (Uchendu et al. 2020; Chakraborty et al. 2023). To overcome this, training-free methods identify intrinsic statistical differences between human and machine text. These methods have evolved significantly, beginning with simple global statistics like average Log-Likelihood and Log-Rank (Solaiman et al. 2019). A major advancement is the distributional discrepancy paradigm, pioneered by DetectGPT (Mitchell et al. 2023), which contrasts original text with its perturbed versions, though often at a high computational cost. Subsequent work like Fast-DetectGPT (Bao et al. 2023) has aimed to improve the efficiency of this approach. More recently, the state-of-the-art method, Lastde (Xu et al. 2024), introduces a time-series perspective to analyze the token probability sequence. However, its innovative approach relies on a complex, multi-stage feature engineering pipeline with sensitive hyperparameters, suggesting that a simpler, more fundamental signal has yet to be found. Our work diverges from this time-domain analysis, proposing that this essential distinction lies in the frequency domain.

2.2 Frequency-Domain Methods in NLP

Frequency-domain analysis is a cornerstone of signal processing and has been applied to NLP tasks like sentiment analysis (Chakraborty 2022) and to efficiently approximate self-attention in Transformers (Lee-Thorp et al. 2021). However, to our knowledge, no prior work has applied spectral analysis to the token probability sequence for LLM text detection. Our work is the first to bridge this gap, demonstrating that a direct frequency-domain analysis can reveal fundamental and highly discriminative features.

3 The SpecDetect Method

In this section, we introduce our proposed method, SpecDetect. Our approach is grounded in a frequency-domain analysis of the token probability sequence, from which we derive and systematically evaluate a suite of candidate spectral features. Based on this investigation, which identifies DFT Total Energy as the most robust indicator, we formally define two detectors: the simple, single-feature SpecDetect and its sampling-based extension, SpecDetect++. The overall pipeline of our method is illustrated in Figure 2.

3.1 Frequency-Domain Analysis

A Signal Processing Perspective. Our analysis begins with the core intuition that human and machine writing differ in "generative vitality." Human text, with its surprising word choices, produces high-amplitude fluctuations in its token probability sequence. In contrast, LLMs are fundamentally constrained to a high-probability vocabulary (Holtzman et al. 2019), resulting in lower-amplitude variations. While present in the time domain, these amplitude differences are obscured by the signal's noisy nature. We hypothesize that the frequency domain provides a more effective means to quantify these fundamental fluctuation patterns.

To test this, we adopt a novel perspective: we treat the token probability sequence as a signal. Given an input text $\mathbf{x} = (x_1, \ldots, x_n)$ and a proxy LLM M_θ, we first obtain the token log-probability sequence $\mathbf{l}(\mathbf{x}) = (l_0, l_1, \ldots, l_{n-1})$. Each element l_i is the log-probability of the $(i+1)$-th token x_{i+1}, which is calculated as:

$$l_i = \log P_\theta(x_{i+1} \mid x_{<i+1}). \tag{1}$$

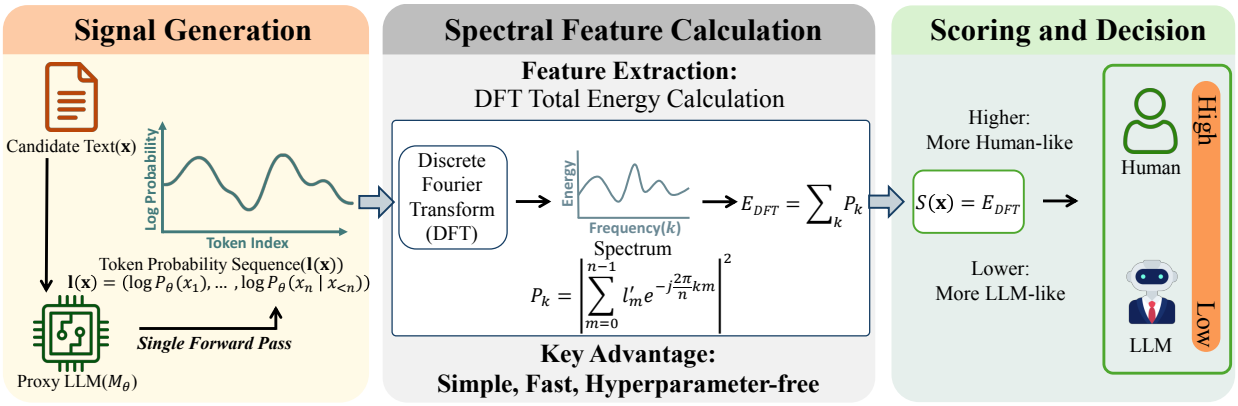


Figure 2: The overall framework of the SpecDetect method. The process consists of three main stages: (1) Inference to obtain the token probability sequence, (2) Frequency-domain transformation to compute the DFT Total Energy, and (3) Scoring to make a final classification.

Here, P_θ is the probability distribution of the language model parameterized by θ, and $x_{<i+1}$ represents the preceding context tokens $(x_1, \ldots, x_i)$. We regard this sequence $\mathbf{l}(\mathbf{x})$ as a **discrete-time signal**. To focus on its dynamic fluctuations, we make the signal zero-mean: $\mathbf{l}'(\mathbf{x}) = \mathbf{l}(\mathbf{x}) - \mu_1$, where μ_1 is the mean of the sequence $\mathbf{l}(\mathbf{x})$.

Spectral Feature Extraction. We hypothesize a key difference between human and machine-generated text lies in the dynamic fluctuations of their token log-probabilities. To characterize these (e.g., amplitude and speed), we analyze the log-probability sequence as a frequency-domain signal, moving beyond individual token probabilities to examine its overall structure. We adopt two complementary tools: the Discrete Fourier Transform (DFT) (Holtzman et al. 2019) for a global view and the Short-Time Fourier Transform (STFT) (Griffin and Lim 1984) for a local perspective.

The DFT provides a global view by decomposing the zero-mean log-probability sequence $\mathbf{l}'(\mathbf{x})$ into constituent frequency components, revealing dominant frequencies across the text. For a real-valued input, it produces a symmetric sequence of complex numbers $\mathbf{X} = (X_0, \ldots, X_{n-1})$, where:

$$X_k = \mathcal{F}(\mathbf{l}')_k = \sum_{m=0}^{n-1} l'_m e^{-j\frac{2\pi}{n}km}, \tag{2}$$

with $\mathcal{F}$ as the DFT operator and k as the frequency index. X_k is a complex value and $P_k = |X_k|^2$ denotes energy at frequency k. We focus on the first $n/2$ components since they contain all the unique frequency information.

In contrast, the STFT offers a local perspective by computing DFT over short, overlapping windows, revealing how frequency content evolves from beginning to end. Formally:

$$S(f,t) = \sum_{m=0}^{L-1} l'_{t+m} w_m e^{-j\frac{2\pi}{L}fm}, \tag{3}$$

where f is the frequency index, t is the token position, and w_m is a window function. Here we use a Hann window (Shimauchi and Ohmuro 2014) of length $L = 20$ and hop size $h = 10$. $|S(f,t)|^2$ represents the energy of each frequency at each time window.

From the spectral transformations, we extract two feature categories: energy-based features and frequency-based features. Specifically, as energy directly reflects the magnitude of sequence fluctuations, the energy-based features below quantify the amplitude of log-probability fluctuations:

- **DFT Total Energy** (E_{DFT}): Measures total fluctuation intensity by summing energy across all frequency components: $E_{DFT} = \sum_{k=0}^{n/2} P_k$. A higher value indicates greater overall variation.
- **STFT Total Energy** (E_{STFT}): Measures total local fluctuation intensity by aggregating spectral energy across all time windows and frequencies: $E_{STFT} = \sum_{t=0}^{T-1} \sum_{f=0}^{L-1} |S(f,t)|^2$. A higher value suggests larger cumulative local fluctuations.
- **Mean Spectral Flux** ($\overline{F}_{spec}$): Captures energy "fluidity" via the change in local spectra: $\overline{F}_{spec} = \frac{1}{T-1} \sum_{t=1}^{T-1} \sum_f |S(f,t) - S(f,t-1)|$. A lower value indicates more stable, self-similar signals over time.

Frequency-based features, by contrast, characterize the shape and complexity of the spectrum, which correspond to the speed and dynamics of the signal's fluctuations:

- **Spectral Centroid** (C_{spec}): Identifies the spectrum's "center of mass" (average fluctuation speed): $C_{spec} = \frac{\sum_{k=0}^{n/2} k \cdot P_k}{\sum_{k=0}^{n/2} P_k}$. Higher values indicate faster variations.
- **Spectral Entropy** (H_{spec}): Measures global spectrum uniformity: $H_{spec} = -\sum_{k=0}^{n/2} P_k \log_2(P_k)$. A lower value signifies energy concentration in fewer frequencies (more predictable/less complex signals).
- **Spectral Entropy Variance** ($V_{H_{spec}}$): Measures stability of complexity over time (variance of local spectral entropy): $V_{H_{spec}} = \text{Var}_t[H_{spec}(t)]$ where $H_{spec}(t) = -\sum_{f=0}^{L-1} S(f,t) \log_2(S(f,t))$. Lower values indicate more consistent frequency distribution across windows.

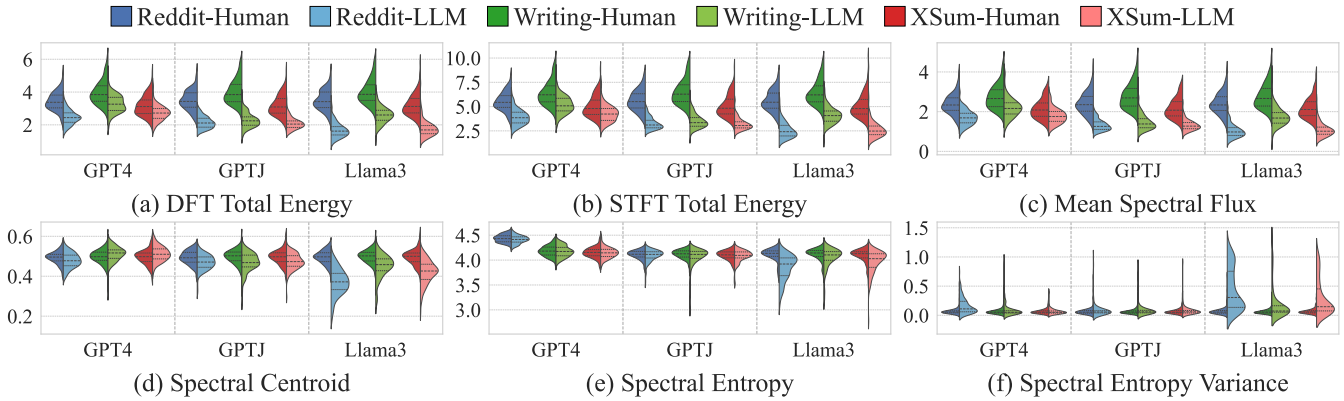


Figure 3: Distributions of spectral features for human and LLM-generated text across different datasets and source models. The top row displays energy-based features (E_{DFT}, E_{STFT}, $\overline{F}_{spec}$), while the bottom row shows frequency-based features (C_{spec}, H_{spec}, $V_{H_{spec}}$). The visualization clearly demonstrates that energy-based features provide a strong and consistent separation between the two classes, whereas the distributions for frequency-based features show significant overlap.

These spectral metrics enable comprehensive quantitative analysis of human and LLM-generated texts.

Analysis and Selection of Spectral Features. To systematically explore spectral differences between human and LLM-generated text, we analyze three datasets: XSum (Narayan, Cohen, and Lapata 2018), Reddit ELI5 (Fan et al. 2019), and WritingPrompts (Fan, Lewis, and Dauphin 2018). Using source models GPT-J (Wang et al. 2023), LLaMA3-8B (AI@Meta 2024), and GPT-4-Turbo (Achiam et al. 2024) (with GPT-J as proxy), we test both white-box (GPT-J) and black-box (LLaMA3-8B, GPT-4-Turbo) conditions. Figure 1 shows a sample pair for intuition (STFT visualizations in Appendix Figure 10), suggesting human text has a more volatile time-domain signal, with a spectrum of significantly higher overall energy.

To verify this observation statistically, we evaluate our candidate features by visualizing their distributions in Figure 3. The results reveal a decisive pattern. The frequency-based metrics, which characterize the speed of fluctuations, show substantial overlap between human and LLM text. This suggests that due to sampling strategies like top-k, LLMs can produce variations in token choice at a speed that is not distinctly different from that of human writing.

However, the **energy-based metrics consistently and significantly separate the two classes**. This confirms our core hypothesis. Because LLM decoding is largely restricted to a narrow set of high-probability tokens (Holtzman et al. 2019), the **amplitude** of its log-probability fluctuations is inherently constrained and much smaller than the wider, more varied fluctuations found in human text. This low-amplitude signature is a fundamental artifact of the generation process.

Given that the energy-based metrics are most effective, we perform a Pearson correlation analysis to understand their relationships. The analysis reveals that **all three metrics are highly correlated with one another**: the correlation between the global E_{DFT} and the local E_{STFT} is exceptionally high (0.97), and both are also strongly correlated with the Mean Spectral Flux, $\overline{F}_{spec}$ (0.89 and 0.94, respectively). This strong linear relationship suggests that all three features capture the same underlying low-amplitude phenomenon and are therefore largely redundant. Given this informational redundancy, and in accordance with the principle of parsimony, we select the **DFT Total Energy** (E_{DFT}) as our single robust feature, as it is a global, hyperparameter-free, and the simplest of the three metrics.

3.2 The Proposed Detector: SpecDetect

Based on our frequency-domain analysis, we propose SpecDetect, a simple and training-free detector. As shown in Figure 2, the detection process follows three main steps:

1. **Signal Generation:** For a given candidate text $\mathbf{x}$, retrieve its zero-mean log-probability sequence, $\mathbf{l}'(\mathbf{x})$, using a proxy model M_θ.
2. **Spectral Feature Calculation:** Apply the DFT to the signal to convert it to the frequency domain and compute its DFT total energy, $E_{DFT} = \sum_{k=0}^{n/2} |X_k|^2$, where X_k is the k-th component of the DFT output.
3. **Scoring and Descision:** Use the DFT total energy E_{DFT} as the final detection score. A higher score indicates a greater likelihood that the text is human-written.

The entire process of deriving the SpecDetect score $S(\mathbf{x})$ from $\mathbf{l}(\mathbf{x})$ can be concisely summarized as:

$$S(\mathbf{x}) = \sum_{k=0}^{n/2} |\mathcal{F}(\mathbf{l}(\mathbf{x}) - \mu_{\mathbf{l}})_k|^2 \tag{4}$$

where $\mu_{\mathbf{l}}$ is the mean of the sequence $\mathbf{l}(\mathbf{x})$, and $\mathcal{F}$ denotes the Discrete Fourier Transform operator. Following prior work (Bao et al. 2023; Xu et al. 2024), we treat the output score $S(\mathbf{x})$ as the likelihood of text being human-written. For a sequence of length n, the naive DFT has a time complexity of $O(n^2)$, but its efficient implementation via the Fast Fourier Transform (FFT) (Cooley and Tukey 1965) reduces this to $O(n \log n)$, ensuring computational efficiency.

Method	GPT-2	Neo-2.7	OPT-2.7	LLaMA-13	LLaMA2-13	LLaMA3-8	OPT-13	BLOOM-7.1	Phi-4	Qwen3-8	Claude-3	GPT4-Turbo	Avg.
sample-based													
Likelihood	0.6645	0.6710	0.6740	0.6575	0.6861	0.9730	0.6880	0.6180	0.7185	0.9838	0.9838	0.7970	0.7596
LogRank	0.7076	0.7116	0.7228	0.7030	0.7264	0.9789	0.7297	0.6749	0.7756	**0.9871**	0.2104	0.7920	0.7267
Entropy	0.6123	0.5866	0.5454	0.4913	0.4517	0.2028	0.5308	0.6084	0.3910	0.2840	0.9858	0.3510	0.5034
DetectLRR	0.7916	0.7921	0.8188	0.7850	0.7882	0.9603	0.8013	0.7955	0.7662	0.9710	0.9777	0.7384	0.8322
Lastde	0.9011	0.9040	0.8989	**0.8068**	0.8019	**0.9851**	0.8990	**0.8834**	0.7353	0.9654	**0.9893**	0.7609	0.8776
SpecDetect	**0.9078**	**0.9171**	**0.9118**	0.8051	**0.8090**	0.9761	**0.9005**	0.8781	**0.7773**	0.9783	0.9833	**0.8054**	**0.8875**
distribution-based													
DetectGPT	0.6933	0.7683	0.6775	0.6483	0.6925	0.7550	0.7342	0.5767	0.7733	0.8275	0.9308	0.7525	0.7358
DetectNPR	0.7133	0.7683	0.6817	0.6808	0.7167	0.9450	0.7300	0.6367	0.7575	0.9383	0.9542	0.7758	0.7749
DNA-GPT	0.6254	0.6295	0.6362	0.6029	0.6649	0.9581	0.6777	0.6254	0.7250	0.9558	0.9592	0.6860	0.7288
Fast-DetectGPT	0.8967	0.8874	0.8651	0.7758	0.7761	0.9724	0.8616	0.8458	0.7992	**0.9877**	**0.9996**	0.8818	0.8791
Lastde++	0.9463	0.9543	0.9395	0.8517	0.8556	**0.9842**	0.9350	0.9215	**0.8534**	0.9720	0.9994	0.8818	0.9246
SpecDetect++	**0.9518**	**0.9587**	**0.9465**	**0.8636**	**0.8643**	0.9686	**0.9360**	**0.9333**	0.8289	0.9867	0.9896	**0.8833**	**0.9259**

Table 1: Detection AUC under black box scenarios, with values averaged across three datasets: XSum, WritingPrompts, and Reddit. The last column ("Avg") denotes the mean AUC across all source models, calculated over the three datasets. Best and second-best results are highlighted in **bold** and underline, respectively.

3.3 Extension with Sampling: SpecDetect++

To further enhance robustness, we introduce **SpecDetect++**, including a sampling discrepancy method similar to previous work (Bao et al. 2023). The core intuition is that LLM text scores are typical within their LLM-generated variation distribution, while human text scores act as outliers.

To formalize this, for a given text $\mathbf{x}$ and a proxy model M_θ, we first consider a distribution of contrastive samples $\{\tilde{\mathbf{x}}\}$. Following (Xu et al. 2024), we generate these samples using the same model, such that $\tilde{\mathbf{x}} \sim M_\theta(\cdot|\mathbf{x})$. We then compute the mean μ_S and variance σ_S^2 of the base SpecDetect scores, $S(\tilde{\mathbf{x}})$, over this sample distribution. Formally, these statistics are the expectations:

$$\mu_S = \mathbb{E}_{\tilde{\mathbf{x}}\sim M_\theta}[S(\tilde{\mathbf{x}})] \quad \text{and} \quad \sigma_S^2 = \mathbb{E}_{\tilde{\mathbf{x}}\sim M_\theta}[(S(\tilde{\mathbf{x}}) - \mu_S)^2].$$

With the mean and standard deviation of the sample distribution, the final SpecDetect++ score for the original text, $S_{++}(\mathbf{x})$, is calculated as its z-score:

$$S_{++}(\mathbf{x}) = \frac{S(\mathbf{x}) - \mu_S}{\sigma_S},$$

where $S(\mathbf{x})$ is the base SpecDetect score of the original text. In practice, μ_S and σ_S are estimated by generating a finite number of samples N. This normalization sharpens the classification boundary and improves robustness against adversarial attacks by amplifying the intrinsic differences between human and machine-generated text.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our methods on a diverse suite of datasets, following the configurations of prior work (Bao et al. 2023; Xu et al. 2024). Our experiments are conducted in two settings: a white-box scenario using the XSum (Narayan, Cohen, and Lapata 2018), SQuAD (Rajpurkar et al. 2016), and WritingPrompts (Fan, Lewis, and Dauphin 2018) datasets, and a more realistic black-box scenario using XSum, Reddit ELI5 (Fan et al. 2019), and WritingPrompts. To assess performance on other languages, our analysis also includes two non-English datasets: WMT16-De (German) (Bojar et al. 2016) and Zhihu-Economy (Chinese) (Xu et al. 2024). In all cases, LLM-generated text is created by prompting source models with the initial 30 tokens of each human-written example, ensuring the human and machine-generated texts have matching lengths. More details are in Appendix A.2.

Models. We evaluate our methods under two distinct scenarios: white-box and black-box. Following previous work (Xu et al. 2024), the white-box setting assumes that the proxy model used for detection is the same as the source model that generated the text. In contrast, the more challenging and realistic black-box setting assumes that the detector does not have access to the source model. Therefore, a different proxy model must be used. For each setting, we test against a diverse set of 12 source models, encompassing a wide range of open-source architectures and state-of-the-art closed-source models. Unless otherwise specified, the open-source GPT-J-6B serves as the sole proxy model for all black-box tasks. Detailed specifications for all models are provided in Appendix A.3.

Baselines. We compare our approach against 10 training-free baselines, which are divided into two groups (see Appendix A.1 for full details). Sample-based methods directly use token statistics for scoring and include Log-Likelihood (Solaiman et al. 2019), LogRank (Solaiman et al. 2019), Entropy (Gehrmann, Strobelt, and Rush 2019; Ippolito et al. 2019), DetectLRR (Su et al. 2023), and Lastde (Xu et al. 2024). In contrast, distribution-based methods score text by contrasting it with generated samples and include DetectGPT (Mitchell et al. 2023), DetectNPR (Su et al. 2023), DNA-GPT (Yang et al. 2023), Fast-DetectGPT (Bao et al. 2023), and Lastde++ (Xu et al. 2024). Consistent with prior work (Xu et al. 2024), we use AUC as the evaluation metric and set the sampling number for SpecDetect++ to 100.

4.2 Effectiveness-Efficiency Trade-Off

Detection Performance. We primarily evaluate our methods in the black-box setting, which most closely mirrors real-world applications, though comprehensive white-box results are also provided in Appendix B.3 for completeness. Table 1 presents the average detection AUC across three datasets, while the detailed per-dataset results can be found in Appendix B.2. Our analysis demonstrates that the

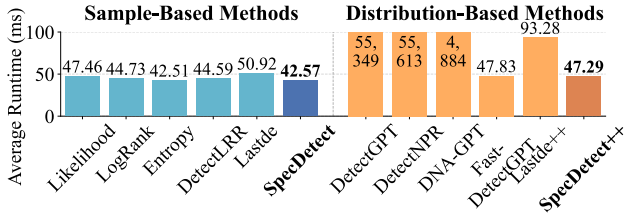


Figure 4: Efficiency comparison of detection methods. Runtimes are averaged per sample (in milliseconds) and labeled on each bar. The bars for perturbation-based methods with extremely long runtimes have been truncated.

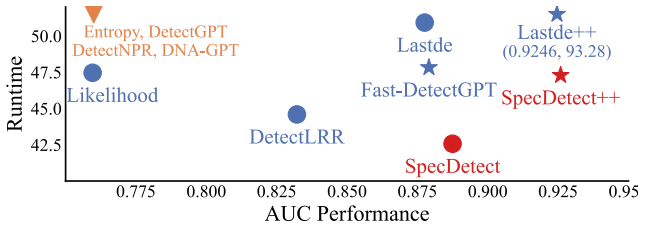


Figure 5: Performance vs. Efficiency trade-off. The x-axis represents detection performance (AUC), and the y-axis represents average runtime (in milliseconds). Methods are grouped by category using marker shapes, and our proposed methods are highlighted in red. Several methods (i.e., Entropy, DetectGPT, DetectNPR, DNA-GPT) are excluded as outliers due to non-competitive performance or prohibitive runtimes. The coordinates for Lastde++ are explicitly labeled because its runtime exceeds the y-axis range.

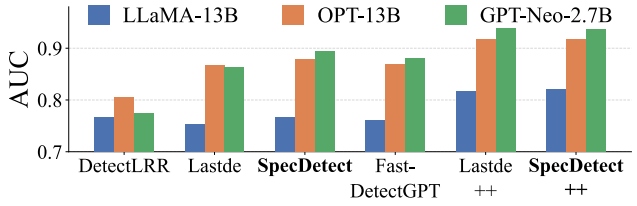


Figure 6: Average AUC performance on paraphrased texts under a black-box setting. Each group of bars represents a different source model.

SpecDetect framework establishes a new state-of-the-art in both categories of training-free detection.

Among single-pass, sample-based methods, our base detector, SpecDetect, consistently outperforms the previous SOTA, Lastde. As shown in Table 1, SpecDetect achieves a higher average AUC (0.8875 vs. 0.8776) and shows particular strength against a wide range of architectures, including both older models (e.g., GPT-2, Neo-2.7) and modern closed-source models like GPT-4 Turbo. This highlights the broad generalizability of our spectral energy feature.

In the more powerful distribution-based category, our enhanced method, SpecDetect++, also achieves state-of-the-art performance. It surpasses the prior SOTA, Lastde++, with a higher average AUC (0.9259 vs. 0.9246) and demonstrates more consistent, robust detection across the majority of tested models. This is a critical advantage for real-world scenarios where the source model is unknown. In summary, SpecDetect and SpecDetect++ each set a new state-of-the-art in their respective classes, offering an unparalleled combination of performance for training-free detection.

Efficiency Analysis. Beyond detection accuracy, computational efficiency is a critical requirement for any practical detector. To evaluate this, we measure the average inference time per sample for each method under a specific black-box condition: detecting text generated by GPT-4-Turbo on the XSum dataset, using GPT-J-6B as the proxy model. All runtimes were measured on a single NVIDIA H800 GPU.

The results, visualized in Figure 4, clearly show that our methods achieve high performance at a significantly lower computational cost. Specifically, our base method SpecDetect is **16.4% faster** than its main competitor, Lastde. More strikingly, our enhanced method, SpecDetect++, is nearly **twice as fast** as the SOTA model Lastde++. The results show that our method achieves superior performance at approximately **half the computational cost** of the latest SOTA.

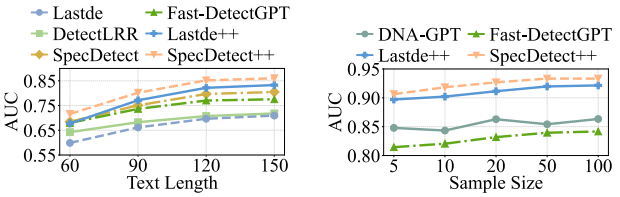
Trade-off Comparison. A critical aspect of a practical detector is its ability to balance high performance with computational efficiency. To visualize this trade-off, we plot the average black-box detection AUC against the average inference runtime for each method, with the results presented in Figure 5. The ideal detector is located in the bottom-right corner, representing both high accuracy and low runtime.

Our methods demonstrate a superior performance-efficiency profile, with SpecDetect achieving the optimal balance within the fast, sample-based category by having the lowest runtime and a high AUC. It's notably effective, outperforming more complex methods like Fast-DetectGPT despite being significantly faster. For the high-performance, distribution-based category, our enhanced method, SpecDetect++, sets a new state-of-the-art, surpassing the previous best, Lastde++, with a higher AUC at approximately half the computational cost. In summary, our SpecDetect framework dominates the performance-efficiency landscape, offering an unparalleled combination of speed and effectiveness for training-free detectors.

4.3 Robustness and In-depth Analysis

We evaluate our method's robustness in challenging, black-box scenarios. Further analyses on decoding strategies and alternative detectors are in Appendix B.7 and B.8.

Robustness to Paraphrasing Attacks. Paraphrasing attacks rewrite human and LLM-generated texts to alter style while preserving meaning, significantly degrading detector performance (Krishna et al. 2023; Sadasivan et al. 2023). Following previous work (Bao et al. 2023; Xu et al. 2024), we evaluate this Robustness by using a T5-Paraphraser (Alisetti 2021) on texts from three source models: GPT-Neo-2.7B, LLaMA-13B, and OPT-13B. Figure 6 shows average results across three datasets (details in Appendix Table 10).



(a) Impact of Text Length (b) Impact of Contrast Sample Size

Figure 7: Analysis of detector performance as a function of (a) text length and (b) number of contrast samples.

Results highlight the robustness of our frequency-domain approach. SpecDetect++ is highly resilient, matching state-of-the-art Lastde++ under adversarial conditions. Notably, our base method SpecDetect consistently outperforms baselines—critical evidence that its core spectral energy feature inherently resists paraphrasing-induced changes. SpecDetect's strong performance without sampling indicates the underlying spectral signature is a fundamental property of generated text, ensuring robustness against such attacks.

Impact of Text Length. Prior studies show shorter texts are harder to detect (Verma et al. 2023; Mao et al. 2024). We investigate this using texts generated by the LLaMA-13B source model across the XSum, WritingPrompts, and Reddit datasets. Since the first 30 tokens of each human-LLM pair are identical, we evaluate on texts truncated to lengths of 60, 90, 120, and 150 words. Figure 7(a) shows the average AUC across these three datasets by text length, with detailed results in Appendix B.5. Consistent with prior work, most methods improve with longer texts. Importantly, our SpecDetect and SpecDetect++ consistently outperform competitors across all lengths, with the gap widening for longer texts. This highlights that our spectral energy feature becomes increasingly powerful with signal length, underscoring its fundamental robustness.

Impact of Contrast Sample Size. For distribution-based methods, we analyze the impact of the number of contrast samples on performance. We conduct experiments using the BLOOM-7.1B source model, with results averaged across three datasets. The results are in Figure 7(b) and detailed results in Appendix B.6. From the results, SpecDetect++ not only consistently outperforms all baselines across all sample sizes but also demonstrates remarkable sample efficiency. Notably, SpecDetect++ with only 10 contrast samples already surpasses the performance of Lastde++ with 50 samples, and nearly matches its performance with 100 samples. This superior sample efficiency highlights the strong and stable discriminative power of our underlying spectral energy feature, which allows for robust detection with significantly fewer samples, further enhancing its efficiency.

Generalization to Different Proxy Models. Proxy model choice critically impacts black-box detection. We investigate this by evaluating detectors on XSum across four different source-proxy pairs, with results in Figure 8. While performance varies significantly, confirming the challenge, our methods demonstrate superior generalization. Our SpecDetect consistently outperforms baselines by a large margin, indicating its spectral feature is more robust to proxy mismatch than complex time-series features. Similarly, SpecDetect++ consistently outperforms other distribution-based methods, showcasing its robustness across all tested pairs.

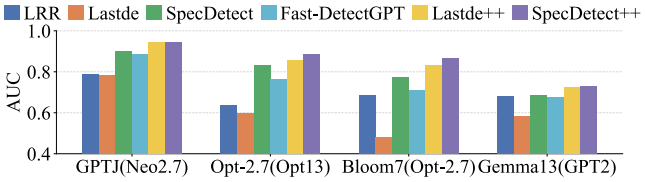


Figure 8: AUC performance with different source and proxy model pairs on the XSum dataset. Each group of bars represents a source-proxy combination.

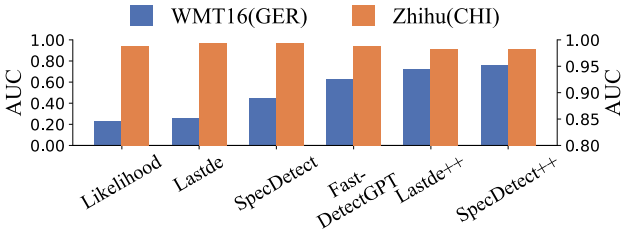


Figure 9: Performance in two non-English scenarios: a German task on WMT16-De (left y-axis; source: mGPT, proxy: GPT-J) and a Chinese task on Zhihu-Economy (right y-axis; source: Qwen-8B, proxy: Yi-1.5-6B).

Performance on Non-English Datasets. To evaluate generalizability beyond English, we conduct German and Chinese experiments, shown in Figure 9. Following previous work (Xu et al. 2024), we test German on WMT16 (source: mGPT, proxy: GPT-J) and Chinese on an economics dataset (source: Qwen-7B, proxy: Yi-1.5-6B). The results show our framework's strong cross-lingual performance. Our base method, SpecDetect, achieves top-tier AUC. The enhanced SpecDetect++ is also highly competitive, outperforming the SOTA Lastde++ in the German scenario, highlighting our approach's effectiveness in non-English settings.

5 Conclusion

We introduce SpecDetect, a novel, training-free LLM text detector using a frequency-domain perspective. By treating the token probability sequence as a signal, it uses a single, hyperparameter-free feature, DFT Total Energy, to robustly distinguish human from machine text. This method is simple and efficient. Experiments confirm SpecDetect set a new SOTA in effectiveness-efficiency trade-off, achieving superior or comparable performance at half the computational cost with strong robustness. This work advances LLM text detection via classical signal processing, and future research can explore more spectral features for fine-grained tasks.

Acknowledgments

This work was supported in whole or in part, by National Natural Science Foundation of China (U24B6012, U2333201, 62372429), the Innovation Funding of ICT, CAS under Grant No. E461040, Pilot for Major Scientific Research Facility of Jiangsu Province of China (No.BM2021800).

References

Abdin, M.; Aneja, J.; Behl, H.; Bubeck, S.; Eldan, R.; Gunasekar, S.; Harrison, M.; Hewett, R. J.; Javaheripi, M.; Kauffmann, P.; et al. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2024. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ackley, D. H.; Hinton, G. E.; and Sejnowski, T. J. 1985. A learning algorithm for Boltzmann machines. *Cognitive science*, 9(1): 147–169.

AI@Meta. 2024. Llama 3 Model Card.

Alisetti, S. V. 2021. Paraphrase generator with t5.

Anthropic. 2024. Introducing the next generation of Claude: Claude 3. https://www.anthropic.com/news/claude-3-family. Accessed: 2025-08-08.

Bao, G.; Zhao, Y.; Teng, Z.; Yang, L.; and Zhang, Y. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

Bhattacharjee, A.; Kumarage, T.; Moraffah, R.; and Liu, H. 2023. Conda: Contrastive domain adaptation for ai-generated text detection. *arXiv preprint arXiv:2309.03992*.

Black, S.; Leo, G.; Wang, P.; Leahy, C.; and Biderman, S. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Bojar, O.; Chatterjee, R.; Federmann, C.; Graham, Y.; Haddow, B.; Huck, M.; Yepes, A. J.; Koehn, P.; Logacheva, V.; Monz, C.; et al. 2016. Findings of the 2016 conference on machine translation (wmt16). In *First conference on machine translation*, 131–198. Association for Computational Linguistics.

Chakraborty, A. 2022. Aspect based sentiment analysis using spectral temporal graph neural network. *arXiv preprint arXiv:2202.06776*.

Chakraborty, S.; Bedi, A. S.; Zhu, S.; An, B.; Manocha, D.; and Huang, F. 2023. On the possibilities of ai-generated text detection. *arXiv preprint arXiv:2304.04736*.

Cooley, J. W.; and Tukey, J. W. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 19(90): 297–301.

Crothers, E. N.; Japkowicz, N.; and Viktor, H. L. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11: 70977–71002.

Currie, G. M. 2023. Academic integrity and artificial intelligence: is ChatGPT hype, hero or heresy? In *Seminars in Nuclear medicine*, volume 53, 719–730. Elsevier.

Else, H. 2023. By chatgpt fool scientists. *Nature*, 613: 423.

Fan, A.; Jernite, Y.; Perez, E.; Grangier, D.; Weston, J.; and Auli, M. 2019. ELI5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Fang, X.; Che, S.; Mao, M.; Zhang, H.; Zhao, M.; and Zhao, X. 2024. Bias of AI-generated content: an examination of news produced by large language models. *Scientific Reports*, 14(1): 5224.

Gehrmann, S.; Strobelt, H.; and Rush, A. M. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Griffin, D.; and Lim, J. 1984. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2): 236–243.

Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; and Choi, Y. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Ippolito, D.; Duckworth, D.; Callison-Burch, C.; and Eck, D. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Krishna, K.; Song, Y.; Karpinska, M.; Wieting, J.; and Iyyer, M. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36: 27469–27500.

Lee-Thorp, J.; Ainslie, J.; Eckstein, I.; and Ontanon, S. 2021. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*.

Li, Y.; Li, Q.; Cui, L.; Bi, W.; Wang, L.; Yang, L.; Shi, S.; and Zhang, Y. 2023. Deepfake text detection in the wild. *arXiv preprint arXiv:2305.13242*.

Mao, C.; Vondrick, C.; Wang, H.; and Yang, J. 2024. Raidar: generative ai detection via rewriting. *arXiv preprint arXiv:2401.12970*.

Mitchell, E.; Lee, Y.; Khazatsky, A.; Manning, C. D.; and Finn, C. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International conference on machine learning*, 24950–24962. PMLR.

Narayan, S.; Cohen, S. B.; and Lapata, M. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Opdahl, A. L.; Tessem, B.; Dang-Nguyen, D.-T.; Motta, E.; Setty, V.; Throndsen, E.; Tverberg, A.; and Trattner, C. 2023. Trustworthy journalism through AI. *Data & Knowledge Engineering*, 146: 102182.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822*.

Rajpurkar, P.; Zhang, J.; Lopyrev, K.; and Liang, P. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Sadasivan, V. S.; Kumar, A.; Balasubramanian, S.; Wang, W.; and Feizi, S. 2023. Can AI-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

Shimauchi, S.; and Ohmuro, H. 2014. Accurate adaptive filtering in square-root Hann windowed short-time Fourier transform domain. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1305–1309. IEEE.

Solaiman, I.; Brundage, M.; Clark, J.; Askell, A.; Herbert-Voss, A.; Wu, J.; Radford, A.; Krueger, G.; Kim, J. W.; Kreps, S.; et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Su, J.; Zhuo, T. Y.; Wang, D.; and Nakov, P. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.

Team, G. 2025a. Gemma 3.

Team, Q. 2025b. Qwen3 Technical Report. arXiv:2505.09388.

Tian, Y.; Chen, H.; Wang, X.; Bai, Z.; Zhang, Q.; Li, R.; Xu, C.; and Wang, Y. 2023. Multiscale positive-unlabeled detection of ai-generated texts. *arXiv preprint arXiv:2305.18149*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Uchendu, A.; Le, T.; Shu, K.; and Lee, D. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, 8384–8395.

Verma, V.; Fleisig, E.; Tomlin, N.; and Klein, D. 2023. Ghostbuster: Detecting text ghostwritten by large language models. *arXiv preprint arXiv:2305.15047*.

Wang, B.; and Komatsuzaki, A. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax. Accessed: 2025-08-08.

Wang, P.; Li, L.; Ren, K.; Jiang, B.; Zhang, D.; and Qiu, X. 2023. SeqXGPT: Sentence-level AI-generated text detection. *arXiv preprint arXiv:2310.08903*.

Workshop, B.; Scao, T. L.; Fan, A.; Akiki, C.; Pavlick, E.; Ilić, S.; Hesslow, D.; Castagné, R.; Luccioni, A. S.; Yvon, F.; et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Xu, Y.; Wang, Y.; Bi, Y.; Cao, H.; Lin, Z.; Zhao, Y.; and Wu, F. 2024. Training-free LLM-generated Text Detection by Mining Token Probability Sequences. *arXiv preprint arXiv:2410.06072*.

Yang, X.; Cheng, W.; Wu, Y.; Petzold, L.; Wang, W. Y.; and Chen, H. 2023. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *arXiv preprint arXiv:2305.17359*.

Young, A.; Chen, B.; Li, C.; Huang, C.; Zhang, G.; Zhang, G.; Wang, G.; Li, H.; Zhu, J.; Chen, J.; et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M.; Li, X.; Lin, X. V.; et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.