AI Text Detectors as a House of Cards:From Vulnerability Induced by Decoding Strategies to Robustness Through Restoration

Anonymous ACL submission

Abstract

With the rapid advancement of Large Language Models (LLMs), their generated text has become increasingly fluent and human-like, making it harder to distinguish machine-generated content from human-written text. Although many existing detectors report over 90% AUC under their experimental settings, their robustness remains questionable. Prior work on robustness has primarily focused on sentence-level perturbations and rewriting, overlooking a crucial factor—distribution shifts introduced by token-level decoding strategies. We find that even minor changes to decoding parameters or strategies can drastically reduce the AUC of strong detectors to around 50%, revealing a severe lack of robustness to decoding-induced variations. To systematically analyze this issue, we study how different token-level decoding strategies affect textual features and the internal state distributions of detectors. Based on these insights, we propose a restore transformation that restores the internal-state distributions induced by diverse decoding strategies to the original distribution, thereby improving detector robustness.We opened source our code in Github Repo

1 Introduction

In recent years, Large Language Models (LLMs) have achieved remarkable progress in natural language generation, producing highly fluent text often indistinguishable from human-authored works (Brown et al., 2020; Blog, 2022; Ippolito et al., 2019; Radford et al., 2019; Touvron et al., 2023). While this capability offers unprecedented productivity opportunities, it also brings growing societal concerns and risks, including misinformation (Ahmed et al., 2021) and academic misconduct in educational settings (Lee et al., 2023; Heaven, 2023; Zhang et al., 2024). Consequently, developing efficient and robust AI-generated text detectors has become a critical challenge for maintaining information credibility, academic fairness, and fostering a trustworthy AI ecosystem.

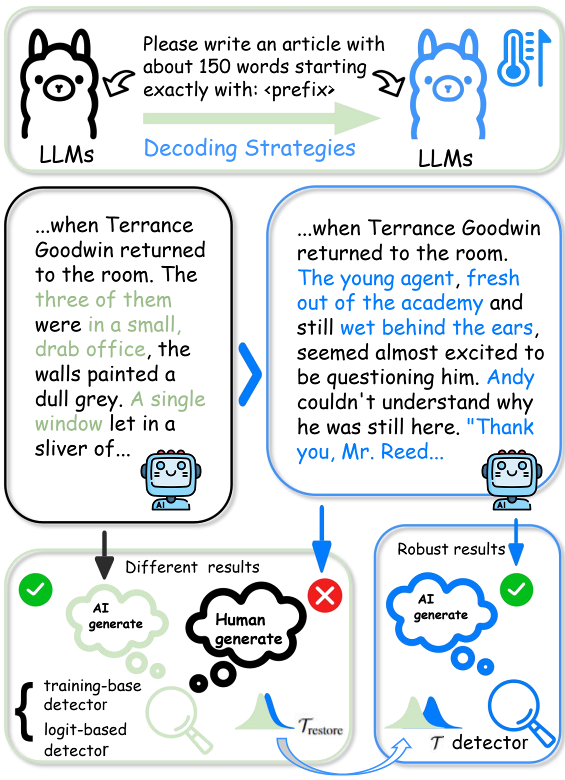

Figure 1: AI-generated text detectors experience a decline in detection performance when encountering texts produced by LLMs' complex decoding strategies. The $\mathcal{T}_{restore}$ transform aims to mitigate this problem.

Current AI text detection methods, including supervised classifiers (Solaiman et al., 2019; Fagni et al., 2021; Mitrović et al., 2023) and zero-shot detectors (Mitchell et al., 2023; Su et al., 2023; Yang et al., 2023; Gehrmann et al., 2019; Bao et al., 2023), have made significant strides.

However, despite their effectiveness in identifying typical machine-generated text, their robustness remains insufficient when confronting increasingly complex generation scenarios (Dugan et al., 2024). A core, underexplored challenge lies in how LLMs' diverse token-level decoding strategies profoundly impact the statistical properties of generated text. Existing detectors often fail to adequately account for these variations, leading to significant performance declines. Unlike prior work focusing on post-generation text obfuscation or rewriting to evade detection (Krishna et al., 2023; Sadasivan et al., 2023; Liyanage and Buscaldi, 2023; Wolff and Wolff, 2020), the direct influence of these fundamental decoding strategies on detector robustness remains systematically under-explored (Massarelli et al., 2019; Holtzman et al., 2019; Vijayakumar et al., 2016; Li et al., 2016). Our research, for the first time, systematically explores and reveals this widespread lack of robustness against diverse decoding strategies. We found that most current AI-generated text detection methods suffer significant performance degradation when dealing with high temperature and high repetition penalty, with many even yielding an AUC score below 0.5.

To address this fundamental issue, this study investigates the impact of different token-level decoding strategies on AI-generated text detectors and proposes an improved detection strategy. Based on observations of the distributions of features inside the detector that separate different texts, we find that more stochastic decoding strategies induce a distributional shift. Specifically, the overlap of the distributions increases, making it difficult to distinguish whether the text is from AI or human.

We aim to restore the original distribution, but we cannot apply the same transformation recovery uniformly to all texts; otherwise, it would merely result in a distribution shift and remain indistinguishable. Therefore, we extracted Distinct-2, Trigram Repeat Ratio, and Character Entropy from a large number of text features, which are correlated with the randomness introduced by complex decoding. These features are then utilized by our **Adaptive Distribution Rectifier**, a learnable module designed to estimate the restorative bias needed for distribution correction. Subsequently, a $\mathcal{T}_{restore}$ transformation is applied to the original distribution, enabling this transformation to adaptively select the appropriate recovery distribution strategy based on the text's characteristics, meaning that more random texts adopt a more aggressive recovery strategy.

Extensive experiments on multiple datasets and across different types of language models show that our proposed recovery transformation $\mathcal{T}_{restore}$ makes excellent detectors robust to more stochastic decoding strategies, consistently achieving strong detection performance.

The three main contributions of this work are:

(1) Systematically demonstrate that decoding strategies degrade detector robustness: The first systematic token-level study of how different decoding strategies affect AI-generated-text detectors, as opposed to sentence-level rewriting. We show that certain decoding strategies can severely degrade the performance of many detectors.

(2) Analyze the impact of decoding strategies on detectors' internal feature distributions: An analysis of how texts produced by different decoding strategies alter detectors' internal distributions, characterizing the resulting distributional shifts within the detectors.

(3) Propose a restorative transformation that enhances detection robustness: Based on this distributional understanding, we design a transformation $\mathcal{T}_{restore}$ that captures the distinctive signatures of different decoding strategies and restores the original distribution—substantially improving detector robustness to variations in decoding strategy.

2 Analysis & Method

In an effort to evade detector identification, much previous work has focused on sentence-level rewriting (Krishna et al., 2023; Sadasivan et al., 2023). However, such rewriting is often costly, lacks interpretability, and is not always effective. Furthermore, few in this field have paid attention to this aspect at the decoding level, yet this very point proves effective across nearly all detectors.We found that the vast majority of detectors exhibited the same decreasing trend when faced with high temperature and repetition penalties, as shown in Table 2 Consequently, we raise the following questions: (a) Why do all detectors exhibit a consistent trend? (b)How does it specifically impact detectors? (c) How can we utilize this insight to improve our detectors?

2.1 Commonalities of Detection Models

Logit-based models Rather than learning a classifier from large labeled corpora, these methods exploit statistics derived from the model logits, which represent the LLM's next-token probability distribution, to compute a scalar score that differentiates human-written from machine-generated text.

Most of these logit-based detectors show the same qualitative failure mode when confronted with complex decoding strategies. We argue

Categories	Methods	Equations	Common parts
Possibility	Likelihood (Gehrmann et al., 2019)	$p_{det}(X_i = \hat{X}_i \| X_{1:i-1})$	$p_{det}(X_i = \hat{X}_i \| X_{1:i-1})$
	Entropy (Gehrmann et al., 2019)	$-\sum_w p_{det}(X_i = w \| X_{1:i-1}) \log p_{det}(X_i = w \| X_{1:i-1})$	$p_{det}(X_i = \hat{X}_i \| X_{1:i-1})$
	DNA-GPT (Yang et al., 2023)	$\log p(Y_0 \| X) - \frac{1}{K} \sum_{k=1}^{K} \log p(Y_k \| X)$	$\log p(Y_k \| X)$
	DetectGPT (Mitchell et al., 2023)	$\log p_\theta(x) - \mathbb{E}_{\tilde{x} \sim q(\cdot \| x)} \log p_\theta(\tilde{x})$	$\log p_\theta(x)$
	Fast-DetectGPT (Bao et al., 2023)	$\mathbf{d}(x, p_\theta, q_\varphi) = (\log p_\theta(\tilde{x} \| x) - \tilde{\mu}) / \tilde{\sigma}$	$\log p_\theta(\tilde{x} \| x)$
Rank	Rank (Gehrmann et al., 2019)	$rank$ in $p_{det}(X_i \| X_{1:i-1})$	$r_\theta(x)$
	LRR (Su et al., 2023)	$\frac{\frac{1}{n} \sum_{p=1}^{n} \log r_\theta(\tilde{x}_p)}{\log r_\theta(x)}$	$r_\theta(x)$
Possibility&Rank	NPR (Su et al., 2023)	$\left\| \frac{\frac{1}{t} \sum_{i=1}^{t} \log p_\theta(x_i \| x_{<i})}{\frac{1}{t} \sum_{i=1}^{t} \log r_\theta(x_i \| x_{<i})} \right\|$	$\log p_\theta(x_i \| x_{<i}), r_\theta(x)$

Table 1: In terms of mathematical expression, these methods exhibit highly similar formulations, where both $p_\theta(x)$ and are r_θ derived from the elements of matrix M, which is also why they exhibit the same vulnerability.

that there is a unified theoretical explanation for this phenomenon, which stems from the shared transformer-based decoding mechanism used by modern LLMs.

We formulate the decoding mechanism as follows. Given the final hidden states $H_{\text{final}} \in \mathbb{R}^{n \times d_{\text{model}}}$, the model projects them into the vocabulary space via W_o to obtain logits, which are normalized by softmax to form the probability matrix $M \in \mathbb{R}^{n \times V}$. Logit-based detectors then compute a scalar score $s = f(M, r_\theta)$, where r_θ denotes the token ranking derived from M. Thus, detection fundamentally relies on the properties of M. Intuitively, standard AI generation follows a "path of least resistance," consistently selecting tokens located in the high-probability accumulation zone of M. In contrast, human writing introduces unpredictable deviations that manifest as dispersed token choices. Existing detectors essentially measure the degree of adherence to this greedy tendency. However, complex decoding strategies disrupt this pattern by artificially suppressing high-probability tokens or elevating the tail, thereby masking the statistical fingerprints of the machine.

We hypothesize that this phenomenon arises because LLMs tend to select tokens with higher model probabilities, whereas human authors exhibit greater variability in token choice. This means that when AI selects tokens for each row of the M matrix, it will always tend to choose those with higher probabilities, whereas humans will not. This observation underlies the design of essentially all logit-based detectors and will remain relevant so long as LLMs use transformer-style decoding that prefers higher-probability next tokens. Table 1 visualizes this unifying intuition.

Training-based models Training-based models include classifiers such as RoBERTa-base and RoBERTa-large that are trained on large labeled datasets. These models generally exhibit better robustness to complex decoding strategies than many logit-based methods, but several caveats apply: (1) under standard (non-adversarial) decoding their AUC is often inferior to state-of-the-art logit-based detectors; (2) as LLMs evolve, these supervised detectors may require periodic retraining to retain peak performance.

The reason training-based models also show some degradation under diverse decoding strategies is straightforward: their supervised training is predominantly exposed to text produced by standard decoding regimes, so when confronted with underrepresented or unseen decoding variants, their classification performance naturally declines. It is precisely for this reason that RoBERTa-large, trained on a broader dataset, exhibits better robustness when facing complex decoding strategies, whereas its performance on normal text is inferior to that of RoBERTa-base, as shown in Table 2.

2.2 Impact of Commonalities on Robustness

Based on the above discussion, any decoding strategy that influences the probability of token selection for each row of the M matrix, thereby making the choice of high-probability tokens less certain, will inevitably impact the final detection performance. As an illustrative example, consider the standardized log-likelihood statistic used in Fast-DetectGPT-style methods:

$$d(x, p_\theta, q_\varphi) = \frac{\log p_\theta(\tilde{x} \mid x) - \tilde{\mu}}{\tilde{\sigma}}, \qquad (1)$$

where

$$\tilde{\mu} = \sum_{\tilde{x}} q_\varphi(\tilde{x} \mid x) \log p_\theta(\tilde{x} \mid x) \qquad (2)$$
$$\hat{\sigma}^2 = \mathbb{E}_{\tilde{x} \sim q_\varphi(\tilde{x} \mid x)}[(\log p_\theta(\tilde{x} \mid x) - \tilde{\mu})^2].$$

Here, the model likelihood factorizes as

$$p_\theta(\tilde{x} \mid x) = \prod_j p_\theta(\tilde{x}_j \mid x_{<j}). \quad (3)$$

When the input x is machine-generated under typical decoding, the factor $p_\theta(x_j \mid x_{<j})$ tends to be relatively large for specific positions j, yielding larger values of the $d(x, p_\theta, q_\varphi)$. Conversely, when x is human-written, the next token conditional probabilities are typically lower and more diffuse, producing smaller values of the corresponding statistic $d(x, p_\theta, q_\varphi)$.

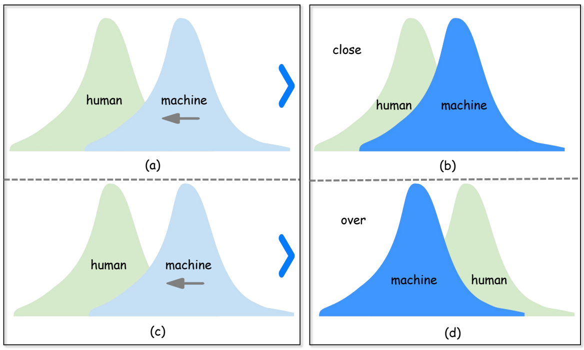

Figure 2: Figures (a) and (c) show the initial distributions of human-written and machine-generated texts, respectively. Figure (b) represents the distribution where machine-generated text approaches that of human text, and Figure (d) illustrates the distribution where machine-generated text surpasses the human text's distribution.

However, when decoding strategies that increase diversity are applied, the machine-generated text will include relatively lower-probability tokens than in the normal setting. This pushes the machine-generated distribution's statistics closer to the human distribution, increasing overlap between the score distributions and making separation more difficult, as shown in Figure 2 (a)(b).

Anomalous behavior of entropy-based detectors can be explained similarly. Entropy-based methods effectively combine the per-token conditional probabilities multiplicatively, as shown in Equation 4, which tends to amplify the impact of high-temperature sampling and repetition penalties. As a result, entropy scores may first move toward the human distribution and then move away as decoding parameters vary—this non-monotonicity is illustrated in Figure 2 (c)(d). Local anomalies observed for some methods when repetition penalties are applied can be interpreted under the same framework, although the effect is most pronounced for entropy-based statistics.

$$-\sum_w P(X_i = w \mid X_{1:i-1}) \log P(X_i = w \mid X_{1:i-1}) \quad (4)$$

where w is a token in the vocabulary, and $P(X_i = w \mid X_{1:i-1})$ is the conditional probability of the i-th token being w given the preceding tokens $X_{1:i-1}$. This expression calculates the Shannon entropy for the probability distribution of the i-th token.

2.3 Restoring Robustness via Adaptive Transformation

Under more stochastic decoding strategies, which lead to increased overlap in distributions, a natural approach is to restore the original distributions to ensure distinguishability between human-generated and AI-generated texts. We next discuss the influential temperature sampling and repetition penalty. For descriptive convenience, we unify high temperature and high penalty as increasing text stochasticity. Furthermore, when dealing with unknown texts, discerning specific decoding strategies is not feasible; hence, a common descriptive point is required to facilitate subsequent transformations.

Given that this transformation targets unknown texts, including (1) human-generated, (2) AI-generated under normal conditions, and (3) highly stochastic AI-generated texts from complex decoding strategies, applying a uniform transformation would merely result in a distribution shift, failing to achieve distinguishability. The ideal scenario involves: (a) restoring the highly stochastic AI texts to resemble normal AI texts while minimally impacting human-generated and normal AI texts. This is because human-generated and normal AI texts possess inherent distinguishability, and our objective is simply to restore this property; (b) employing a more aggressive restoration strategy for increasingly stochastic texts.

Consequently, we sought to identify characteristics that fulfill the two aforementioned criteria. After evaluating various features indicative of text randomness, as shown in Appendix A, we identified Distinct-2, Trigram Repeat Ratio, and Character Entropy as satisfying these requirements, as shown in Figure 4. The rationale for selecting these specific features lies in their complementary ability to characterize decoding stochasticity. Distinct-2 captures the lexical diversity that typically inflates under high-temperature sampling; Trigram Repeat Ratio acts as a sensitive proxy for the repetitive

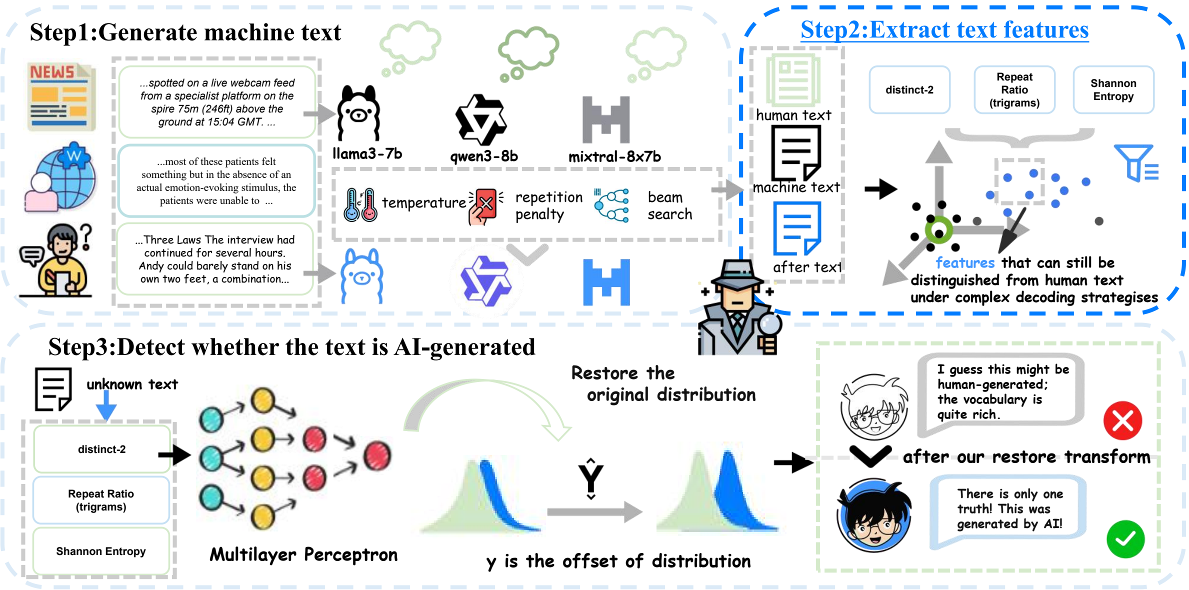

Figure 3: The overview of our total work pipeline.The extraction of text features is crucial for our learned adaptive transformation.

loops common in deterministic decoding; and Character Entropy provides a fine-grained measure of the overall distributional flatness. Together, they form a comprehensive signature of the underlying decoding strategy.

The mathematical definitions of these three features are as follows:

Distinct-2: Given a tokenized text $T = [t_1, t_2, \ldots, t_N]$, Distinct-2 is defined as the ratio of the number of unique bigrams to the total number of bigrams:

$$\text{Distinct-2} = \frac{|\{(t_i, t_{i+1}) \mid 1 \leq i \leq N-1\}|}{N-1} \quad (5)$$

Repeat Ratio (for trigrams): Repeat Ratio is the proportion of repeated n-grams (those appearing more than once) among all n-grams. For trigrams:

$$\text{Repeat Ratio} = \frac{\sum_{g \in G} \mathbb{I}[\text{count}(g) > 1] \cdot \text{count}(g)}{N-2} \quad (6)$$

where G is the set of all trigrams, count(g) is the number of occurrences of trigram g, and N is the total number of tokens.

Char Entropy (Shannon Entropy): The character-level Shannon entropy is defined as:

$$H = -\sum_{c \in C} p(c) \log_2 p(c) \quad (7)$$

where C is the set of unique characters in the text, and $p(c)$ is the probability of character c.

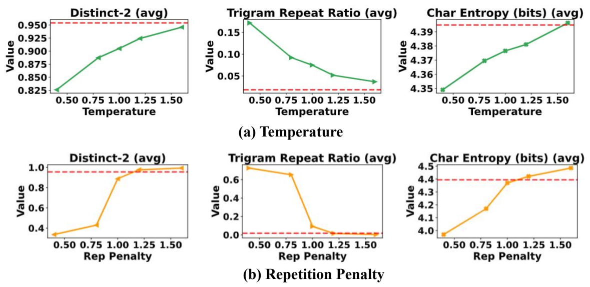

Figure 4: Distinct-2, Trigram Repeat Ratio, and Character Entropy—these three features still maintain good discriminability from human-generated text, even under random decoding strategies. The red dashed line represents the features of human-generated text.

Based on these three features mentioned above, we employ an **Adaptive Distribution Rectifier** to predict the offset required to restore the original distribution, as shown in Equation 8. The rectifier takes the three-dimensional feature vector $\mathbf{x}$ as input and outputs a scalar value $\hat{y}$, which is then used to parameterize the transformation $\mathcal{T}_{restore}$ in Equation 9

Adaptive Rectification:

$$\hat{y} = \mathcal{F}_\phi(\mathbf{x}) \quad (8)$$

After obtaining the output $\hat{y}$, we then apply it to the transformation in Equation 9 to restore the original value. This transformation separates the previously overlapping distributions, thereby increasing the distinguishability between human and machine-generated text and improving both the accuracy and robustness of the detector.

Distribution Transformation:

$$d' = \mathcal{T}_{restore}(d, \hat{y}) = d + \hat{y} \tag{9}$$

where d is the original feature value,which is $d(x, p_\theta, q_\varphi)$in Equation 1, $\hat{y}$ is the offset predicted by the adaptive rectifier in Equation 8, and $\mathcal{T}$ denotes the transformation function.

In subsequent experiments, we will further enhance Fast-DetectGPT by applying this transformation strategy. The transformation $\mathcal{T}_{restore}$ is defined in Equation 9. An overview of the approach is provided in Figure 3.

3 Experiments

3.1 Settings

Datasets Following the approach of DetectGPT (Mitchell et al., 2023), we utilize three datasets to cover a wide range of domains: the SQuAD dataset (Rajpurkar et al., 2016) for Wikipedia contexts, the Writing dataset (Fan et al., 2018) for story generation, and the XSum dataset (Narayan et al., 2018) for news summarization. From each dataset, we randomly sample 150 human-written texts as authentic human-generated samples. For each sample, we provide the first 30 words to the source models with different decoding strategies, prompting them to generate AI-written samples of nearly identical length.

Models and Decoding We select llama3-7b (Dubey et al., 2024), qwen3-8b (Yang et al., 2025), and mixtral-8x7b (Jiang et al., 2024) as the source models for generating texts with complex decoding strategies. For decoding, we control a single variable and focus on the effects of repetition penalty, temperature sampling, beam search, and greedy decoding. The baseline decoding parameters were set to temperature = 0.8 and repetition penalty = 1. The effects of topp and topk have already been discussed in DetectGPT (Mitchell et al., 2023) and found to be less significant,so we didn't put them in the experimental results.

Baselines For logit-based models, we select several state-of-the-art zero-shot detectors, including likelihood, entropy, rank (Gehrmann et al., 2019), LRR, NPR (Su et al., 2023), DNA-GPT (Yang et al., 2023), DetectGPT (Mitchell et al., 2023), and Fast-DetectGPT (Bao et al., 2023). For training-based models, we use RoBERTa-base and RoBERTa-large (Liu et al., 2019), both trained on large-scale parameters.

Metrics We adopt the Area Under the Curve (AUC), a metric commonly used (Mitchell et al., 2023; Bao et al., 2023; Chen et al., 2025) in detector evaluation, which reflects the average performance of a detector across different thresholds and provides a comprehensive measure of its capability. The mathematical formulation is as follows:

$$\text{AUC} = \int_0^1 \text{TPR}(x)\, d\text{FPR}(x) \tag{10}$$

where TPR and FPR denote the true positive rate and false positive rate, respectively.

To reflect the varying proportions of AI-generated text in real-world scenarios, we further introduce a composite metric, denoted as real_score, which combines the AUCs under different decoding strategies. Specifically, we define:

$$\text{real\_score} = a \cdot \text{AUC}_n + (1 - a) \cdot \text{AUC}_i \tag{11}$$

where $a \in [0, 1]$, AUC_n represents the AUC under the normal decoding strategy, and AUC_i denotes the AUC under the intricate decoding strategy. This metric reflects the overall performance of the detector as the proportion of complex decoding strategies changes.

3.2 Results

Performance of Methods under Decoding Strategies: As shown in Table 2 and Figure 5, we present the performance of a wide range of popular detection methods under different decoding strategies. We use different markers to indicate key findings: (1) Facing complex decoding strategies: Pink font denotes the best performance among all strategies for a given method, while light blue font indicates the worst. It can be observed that detectors generally perform better under greedy decoding, whereas their performance tends to degrade under repetition penalty strategies. (2) Differences among methods: No single method consistently achieves optimal performance across all scenarios. Notably, our proposed method maintains relatively robust performance under various decoding strategies compared to other methods, and still achieves excellent results under the baseline condition. (3)It is worth noting that the larger-scale RoBERTa-large, trained on a broader dataset, exhibits better

Type	Method	baseline	strategies			temperature			repetition penalty		
		baseline	greedy	beam3	beam5	temp0.4	temp1.0	temp1.2	rep0.4	rep0.8	rep1.2
Logit-based	Likelihood	0.920	0.974	0.974	0.976	0.966	0.854	0.738	0.986	0.985	0.610
	Entropy	0.347	0.262	0.151	0.119	0.301	0.373	0.415	0.012	0.061	0.316
	Rank	0.706	0.736	0.719	0.722	0.732	0.685	0.656	0.735	0.758	0.653
	LogRank	0.923	0.977	0.974	0.974	0.970	0.858	0.748	0.985	0.985	0.653
	LRR	0.894	0.954	0.932	0.910	0.949	0.831	0.753	0.690	0.868	0.754
	NPR	0.839	0.890	0.869	0.846	0.890	0.778	0.688	0.558	0.677	0.677
	DNA-GPT	0.927	0.975	0.972	0.969	0.972	0.852	0.729	0.985	0.990	0.558
	DetectGPT	0.811	0.812	0.819	0.802	0.830	0.751	0.666	0.429	0.546	0.600
	Fast-DetectGPT	0.972	0.988	0.972	0.965	0.988	0.910	0.764	0.916	0.974	0.474
Training-based	RoBERTa-base	0.864	0.928	0.891	0.880	0.923	0.797	0.747	0.970	0.963	0.470
	RoBERTa-large	0.836	0.905	0.860	0.850	0.897	0.791	0.767	0.988	0.981	0.496
Logit-based	$\mathcal{T}$_restore	0.974	0.989	0.978	0.975	0.989	0.909	0.766	0.912	0.985	0.480

Table 2: A comparison of the performance of various methods under different decoding strategies. Viewed horizontally, the highest and lowest values for a given method are marked in pink and light blue, respectively. Viewed vertically, the best-performing method and the second-best-performing method under each decoding strategy are highlighted with pink and yellow cells, respectively.

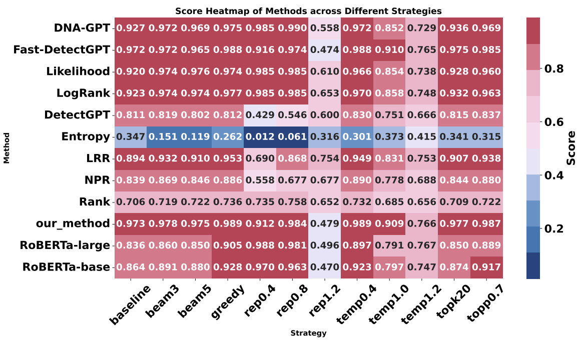

Figure 5: A heatmap visualizing the differences across various methods, strategies, and their respective combinations.

robustness when facing complex decoding strategies. However, on normal text, RoBERTa-large's performance is inferior to RoBERTa-base's. This also provides an intriguing insight: larger models do not necessarily lead to better performance in normal circumstances, despite their commendable robustness.

Performance in Real-World Scenarios: Although our discussion focuses on the robustness of detectors under complex scenarios, it is undeniable that in real-world environments, detectors are mostly exposed to normal text. Therefore, we should not pursue robustness under complex decoding strategies at the expense of performance on standard cases. Using the real_score metric introduced in the Metrics section, Figure 6 illustrates how the performance of each detector changes. We observe that: (1) The slope of the line reflects the anti-interference ability under complex decoding strategies—the larger the slope, the weaker the robustness; (2) The Rank and RoBERTa-large methods even show improved performance under complex decoding strategies; (3) Our method demonstrates superior performance in almost all proportions of real-world scenarios.

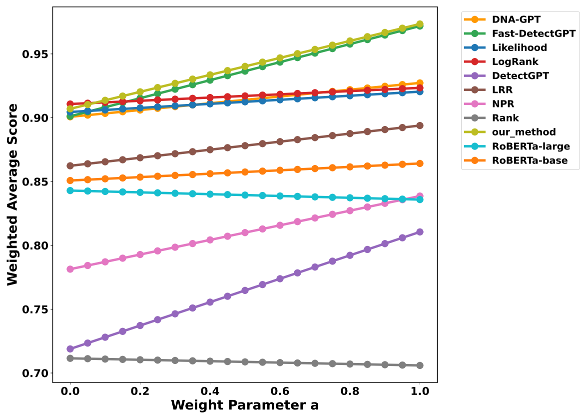

Figure 6: Figure illustrates the performance of various methods in real-world scenarios as the variable a changes, where a represents the proportion of random decoding strategies. A steeper slope indicates poorer robustness to complex decoding strategies.

Differences across Models and Datasets: As shown in Table 3, we find that the overall detection performance of these methods is not balanced across different models and datasets, as detailed in Appendix 3. The percentage indicates by how much a performance is better or worse than the average level of all models across all datasets. Specifically, (1) The AUC difference across models reaches up to 0.133, which indicates a substantial gap. This suggests that the probability of generated text being detected can vary greatly depending on the model used. (2) Texts generated by Mixtral-8x7b consistently achieve the lowest AUC across all datasets, possibly due to its mixture-of-experts architecture leading to greater diversity. (3) Qwen3-8b exhibits significant variation across datasets, while Llama3-7b shows minimal differences, reflecting the stability and diversity of its generated texts.

Dataset	Models	Llama3-7b	Qwen3-8b	Mixtral-8x7b
SQuAD	Average_AUC	0.846	0.731	0.713
	Percentage	+7.47%	-7.16%	-9.44%
Writing	Average_AUC	0.843	0.824	0.767
	Percentage	+7.08%	+4.67%	-2.58%
XSum	Average_AUC	0.842	0.805	0.715
	Percentage	+6.95%	+2.25%	-9.19%

Table 3: Average AUC of different generation models across datasets. Percentages denote deviations from the global mean performance.

4 Related Work

AI Text Detector Current AI-generated text detectors are primarily categorized into logit-based and training-based approaches. Among these, the more mature logit-based detectors, which analyze the internal probabilities and statistics of language models, initially focused on methods based on likelihood, entropy, and rank (Gehrmann et al., 2019). Subsequently, these evolved into more advanced methods such as logrank, NPR, LRR (Su et al., 2023), DNA-GPT (Yang et al., 2023), and Detect-GPT (Mitchell et al., 2023). Fast-DetectGPT (Bao et al., 2023) improved upon DetectGPT (Mitchell et al., 2023) by achieving more efficient sampling and notably faster inference. Training-based detectors, conversely, predominantly rely on large-scale data training to learn distinguishing patterns, with representative works including RoBERTa-base and RoBERTa-large (Liu et al., 2019). RoBERTa-large is a version of RoBERTa-base with a larger parameter count. These represent earlier efforts, and due to the high computational cost of training, along with the continuous retraining needed as Large Language Models evolve, current research predominantly focuses on logit-based detectors.

Decoding Strategies Decoding strategies govern LLM token selection, balancing quality and diversity. **Temperature sampling** scales logits to flatten distributions for creativity or sharpen them for determinism. To exclude unreliable tokens, **Top-k** and **Top-p sampling** (Shi et al., 2019) dynamically truncate the vocabulary. Additionally, **repetition penalty** (Keskar et al., 2019) discourages reiteration to increase lexical variance. In contrast, **Beam search** (Lemons et al., 2022) and greedy decoding prioritize likelihood maximization, often yielding rigid patterns. Despite their role in humanizing AI text, the specific impact of these strategies on the statistical signatures used by detectors remains insufficiently explored.

5 Conclusion

In this work, we systematically revealed a critical, previously under-explored, vulnerability in AI-generated text detectors: their profound susceptibility to diverse token-level decoding strategies. We demonstrated that minor adjustments to these strategies can severely degrade the performance of excellent detectors. Our contributions are threefold: first, we presented the first systematic token-level study on how decoding strategies compromise detector robustness; second, we conducted an in-depth analysis characterizing the distributional shifts induced within detectors by decoding strategies; and third, based on this understanding, we proposed $\mathcal{T}_{restore}$. This novel restorative transformation effectively captures the distinctive signatures of various decoding strategies and restores the detectors' internal-state distributions, thereby improving detection robustness. Extensive experiments confirmed that $\mathcal{T}_{restore}$ consistently enhances the performance of excellent detectors across challenging generation conditions, advancing the development of a more trustworthy AI ecosystem.

Limitations

Exploring various Large Language Models across multiple datasets and diverse decoding strategies incurs a prohibitively high computational cost, exhibiting a near-cubic growth in complexity. Consequently, we are unable to exhaustively cover all decoding strategies, particularly the myriad combinations of temperature and repetition penalties. Furthermore, our proposed method does not consistently outperform all existing approaches across the entire spectrum of decoding strategies. Thus, designing a more efficient distribution restoration method remains a promising avenue for future research.

References

Alim Al Ayub Ahmed, Ayman Aljabouh, Praveen Kumar Donepudi, and Myung Suh Choi. 2021. Detecting fake news using machine learning: A systematic literature review. *arXiv preprint arXiv:2102.04458*.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

O Blog. 2022. Chatgpt: Optimizing language models for dialogue.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiaqi Chen, Xiaoye Zhu, Tianyang Liu, Ying Chen, Chen Xinhui, Yiwen Yuan, Chak Tou Leong, Zuchao Li, Long Tang, Lei Zhang, and 1 others. 2025. Imitate before detect: Aligning machine stylistic preference for machine-revised text detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23559–23567.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

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

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. Tweepfake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Will Douglas Heaven. 2023. Chatgpt is going to change education, not destroy it. *MIT Technology review*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36:27469–27500.

Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. 2023. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*, pages 3637–3647.

Sofia Lemons, Carlos Linares López, Robert C Holte, and Wheeler Ruml. 2022. Beam search: faster and monotonic. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, pages 222–230.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.

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*.

Vijini Liyanage and Davide Buscaldi. 2023. Detecting artificially generated academic text: The importance of mimicking human utilization of large language models. In *International Conference on Applications of Natural Language to Information Systems*, pages 558–565. Springer.

Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. 2019. How decoding strategies affect the verifiability of generated text. *arXiv preprint arXiv:1911.03587*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International conference on machine learning*, pages 24950–24962. PMLR.

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *arXiv preprint arXiv:2301.13852*.

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

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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

Shaohuai Shi, Xiaowen Chu, Ka Chun Cheung, and Simon See. 2019. Understanding top-k sparsification in distributed deep learning. *arXiv preprint arXiv:1911.08772*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, and 1 others. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Max Wolff and Stuart Wolff. 2020. Attacking neural text detectors. *arXiv preprint arXiv:2002.11768*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023. Dnagpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *arXiv preprint arXiv:2305.17359*.

Qihui Zhang, Chujie Gao, Dongping Chen, Yue Huang, Yixin Huang, Zhenyang Sun, Shilin Zhang, Weiye Li, Zhengyan Fu, Yao Wan, and 1 others. 2024. Llm-as-a-coauthor: Can mixed human-written and machine-generated text be detected? *arXiv preprint arXiv:2401.05952*.

A Text features under complex decoding strategies

Figure 7 illustrates the variation of various text feature metrics as a function of temperature. Figure 8 presents the variation of these metrics under different repetition penalties. Distinct-2, Trigram Repeat Ratio, and Character Entropy consistently maintain good discriminability from human-generated text across varying temperatures and repetition penalties.The following shows how each metric changes when facing complex decoding strategies, with the red dashed line representing the corresponding values for human-generated text.

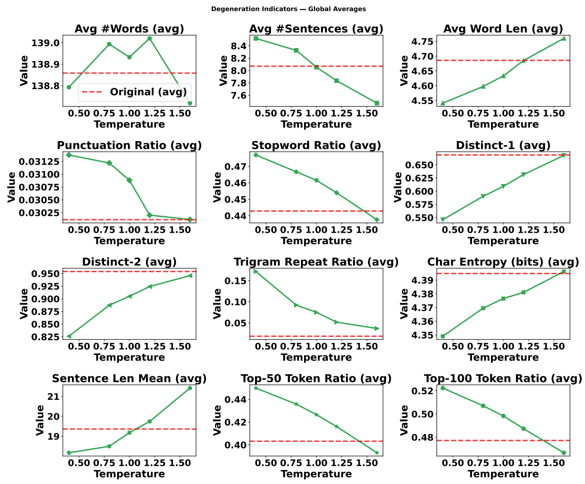

Figure 7: variation of various text feature metrics as a function of temperature

B Performance differences among various dataset-model combinations

Figure 9 displays a heatmap illustrating the performance of different combinations of datasets and generative models across various methods. This heatmap reveals substantial differences among these choices.

C Examples of text generated with complex decoding strategies

Figure 10 presents an illustrative example from the LLaMA-7B model on the writing dataset. The

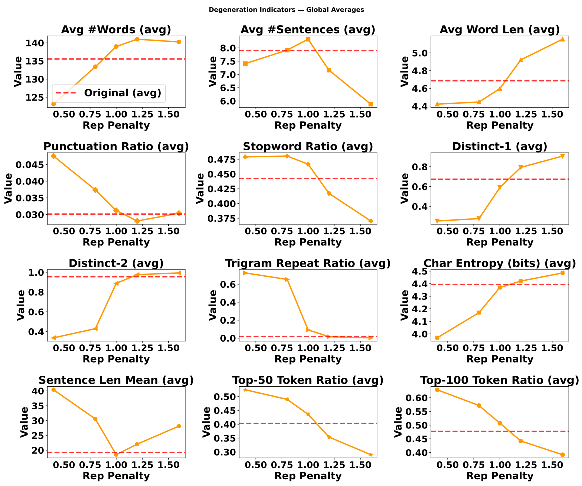

Figure 8: variation of these metrics under different repetition penalties

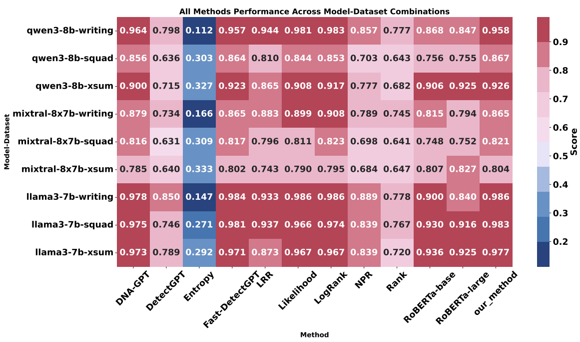

Figure 9: Heatmaps of various dataset and models

green-highlighted sections represent the common prefix among the three outputs. The red-highlighted sections denote emotionally rich and vivid descriptions, while the blue-highlighted sections indicate more conventional descriptions. From this example, it can be observed that while both baseline and high-temperature generations tend towards scene descriptions, human-generated text leans towards plot progression. Furthermore, high-temperature generations exhibit a richer vocabulary.

Human_text: "The Three Laws The interview had continued for several hours. Andy could barely stand on his own two feet, a combination of fatigue and anxiety, when Terrance Goodwin returned to the office. The HR manager held in his hands a stack of papers. Sam found the will to rise as he was expected to do in these instances. \"Please sit,\" Goodwin commanded. Andy did as he was asked. The interview books suggested he follow any orders or suggestions from the company. \"We've run your application and interview responses through the offices and we have the results right here. Congratulations Mr. Ingall, you've been accepted!\" Goodwin held out his hand. Andy rose from his chair and grabbed the inviting limb, shaking it enthusiastically. \"Thank you, Mr. Goodwin,\" he said. \"I won't let you down.\" The HR manager stiffened his arm. Andy withdrew his hand rapidly, slinking back into his chair. \"It's not me you'd be letting",

Baseline_ai_generated: "The Three Laws The interview had continued for several hours. Andy could barely stand on his own two feet, a combination of fatigue and anxiety, when Terrance Goodwin returned to the room. The three of them were in a small, drab office, the walls painted a dull grey. A single window let in a sliver of sunlight, casting a faint glow over the room. Andy's eyes felt heavy, as if weighted down by the gravity of the situation. He glanced at the two men seated across from him, both with serious faces. \"So, Andy, let me sum up what we've discussed,\" Terrance said, his voice calm and measured. \"You're telling me that you have a... device, that you believe can change the course of human history.\" Andy nodded, trying to muster some enthusiasm. \"That's right. It's a machine that can manipulate the fundamental forces of nature.\" Terr",

Temperature1.2_ai_generated: "The Three Laws The interview had continued for several hours. Andy could barely stand on his own two feet, a combination of fatigue and anxiety, when Terrance Goodwin returned to the room. The young agent, fresh out of the academy and still wet behind the ears, seemed almost excited to be questioning him. Andy couldn't understand why he was still here. \"Thank you, Mr. Reed. That will be all for today. We'll be in touch soon.\" Andy's eyes followed the agent as he left the room, the door closing behind him like a tomb. Only then did he allow himself to collapse onto the cold, metal chair. He put his elbows on the table, his face in his hands, and let the despair wash over him. What had he done to deserve this? A small voice in his head, one that had grown silent and still, began to stir. It was the voice of his",

Figure 10: An illustrative example from the LLaMA-7B model on the writing dataset