When AI Settles Down: Late-Stage Stability as a Signature of AI-Generated Text Detection

Anonymous ACL submission

Abstract

Zero-shot detection methods for AI-generated text typically aggregate token-level statistics across entire sequences, overlooking the temporal dynamics inherent to autoregressive generation. We analyze over 120k text samples and reveal *Late-Stage Volatility Decay*: AI-generated text exhibits rapidly stabilizing log probability fluctuations as generation progresses, while human writing maintains higher variability throughout. This divergence peaks in the second half of sequences, where AI-generated text shows 24–32% lower volatility. Based on this finding, we propose two simple features: *Derivative Dispersion* and *Local Volatility*, which computed exclusively from late-stage statistics. Without perturbation sampling or additional model access, our method achieves state-of-the-art performance on EvoBench and MAGE benchmarks and demonstrates strong complementarity with existing global methods.

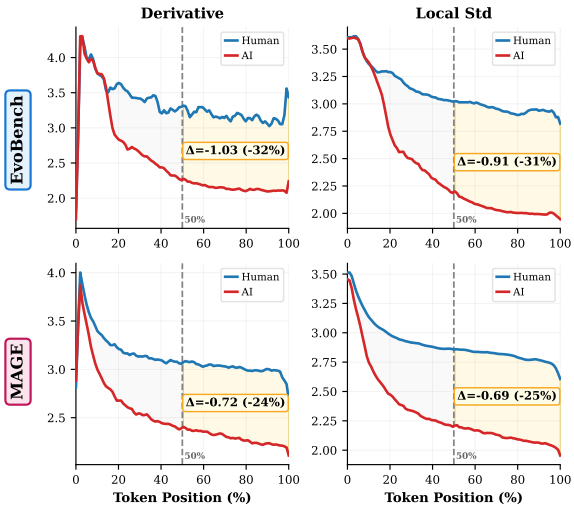

Figure 1: Temporal dynamics of **log probability** features for human vs. AI text on EvoBench (top) and MAGE (bottom). Left: Absolute derivative (change rate); Right: Local standard deviation (volatility). Shaded regions show the human-AI gap, with Δ indicating mean difference (AI − Human) and relative percentage. Yellow regions (50–100%) highlight larger divergence in the second half.

1 Introduction

The widespread adoption of large language models (LLMs) has raised significant concerns about the potential misuse of AI-generated text, making accurate detection of AI-generated content an increasingly important task (Yang et al., 2024b; Wu et al., 2025). Recent zero-shot detection methods, such as DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2024), and LLR (Su et al., 2023) have achieved promising results by analyzing the probabilistic characteristics for different AI-generated and human written texts. These approaches typically compute global statistics, such as average model log-probability, rank, curvature, or sampling discrepancy over the entire text sequence.

Despite their effectiveness, these methods share a common implicit assumption: *token-level statistics are treated as stationary and exchangeable across the sequence.* That is, discriminative signals are assumed to be uniformly distributed, so sequence-wide aggregation suffices. This assumption does not fully leverage characters of the autoregressive nature of LLM generation. Text is not produced all at once; instead, it is generated token by token, with each prediction conditioned on an ever-growing context. As a result, the statistical properties of generated tokens may evolve dynamically over time.

Recent empirical studies have observed that detection performance improves with increasing text length, with longer texts generally being easier to detect (Li et al., 2024; Xu et al., 2024b; Bao et al., 2024). However, this phenomenon is typically attributed to longer texts simply "containing more information." What remains largely unexplored is *how discriminative signals are distributed across different stages of the generation process*,

and whether later tokens exhibit qualitatively different behavior from earlier ones.

To address this issues, we conduct a large-scale temporal analysis of over 120K human-written and AI-generated texts from the EvoBench (Yu et al., 2025) and MAGE (Li et al., 2024) benchmarks. Using multiple surrogate models, we examine how various token-level statistics including negative log-probability, rank, entropy, and sampling discrepancy – evolve across normalized text positions. Crucially, we go beyond raw values and analyze two second-order temporal dynamics: the *derivative*, which captures how rapidly a statistic changes between consecutive tokens, and the *local standard deviation*, which measures regional volatility within a sliding window.

Our analysis reveals that the temporal dynamics of negative log-probability provide particularly discriminative signals. As illustrated in Figure 1, we observe that both the derivative and local standard deviation of negative log-probability decrease more rapidly for AI-generated text than for human text, with the gap widening noticeably in the second half of the sequence (50%–100% positions). Quantitatively, on EvoBench, the derivative of negative log-probability for AI text is 32% lower than that of human text ($\Delta = -1.03$), and local standard deviation is reduced by 31% ($\Delta = -0.91$); on MAGE, similar reductions of 24% and 25% are observed. In other words, AI-generated tokens in the latter portion of text exhibit smaller and more stable probability fluctuations as the model's predictions become increasingly consistent as context accumulates. We term this pattern **Late-Stage Volatility Decay**. This phenomenon reflects the inherent behavior of autoregressive generation: as more context becomes available, the model's prediction distribution sharpens, leading to reduced variability in token-level statistics. Human writing, by contrast, continues to introduce unexpected lexical choices and maintains higher volatility throughout.

Based on this observation, we propose two simple yet effective temporal features computed from token negative log-probability to capture this late-stage dynamics: (1) *derivative dispersion*, the standard deviation of the absolute first derivative in the second half, quantifying the intensity of probability fluctuations; and (2) *local volatility*, calculated as the mean of local standard deviation within a sliding window over the second half, this metric measures local stability, capturing how AI text tends to maintain smoother, more consistent probability transitions as generation progresses.

Since our temporal cues are largely orthogonal to prior signals, a simple fusion with Fast-DetectGPT (Bao et al., 2024) yields consistent gains by combining *global distributional* and *local temporal* evidence. Experiments on MAGE (Li et al., 2024) and EvoBench (Yu et al., 2025) show that our approach excels on longer texts while remaining competitive on shorter ones, and that fusion further improves robustness across diverse generators. Our contributions are summarized as follows:

- We systematically reveal the *Late-Stage Volatility Decay* phenomenon in AI-generated text—a distinctive temporal signature arising from the autoregressive generation process.
- We propose a simple yet effective method that operationalizes this phenomenon through second-half temporal features, requiring no additional training or model access beyond a surrogate LLM.
- We achieve state-of-the-art detection performance across multiple benchmarks, demonstrating the value of temporal analysis in machine-generated text detection.

2 Temporal Dynamics of Zero-Shot Detection Metrics

To understand whether discriminative signals vary across text positions, we empirically analyze how these statistics evolve over generation to identify which temporal patterns best distinguish human and AI-generated text.

2.1 Analysis Setup

Datasets and Models. We analyze over 120k text samples from two established benchmarks: EvoBench (Yu et al., 2025) and MAGE (Li et al., 2024). Both datasets contain parallel human-written and AI-generated texts across diverse domains and generation settings. We use Llama-3-8B-Instruct (Grattafiori et al., 2024) as the primary surrogate model for computing token-level statistics. Results with alternative surrogate models GPT-J-6B (Wang and Komatsuzaki, 2021) and Falcon-7B (Almazrouei et al., 2023) are provided in Appendix K. Additional experiments on frontier models including reasoning-enhanced systems are presented in Appendix L.

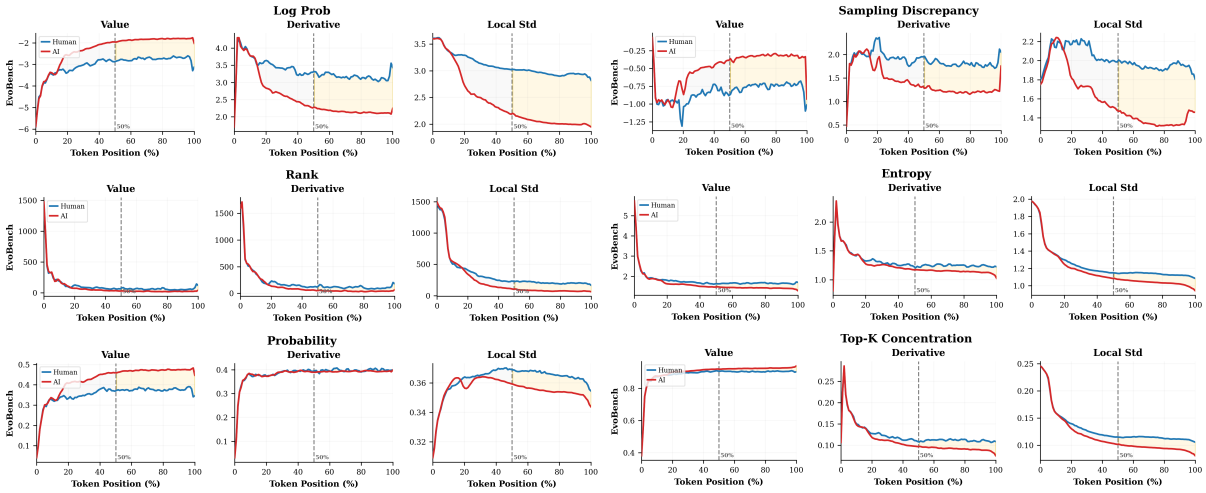

Figure 2: Temporal dynamics of six base metrics on EvoBench. Each row shows a metric's raw value (left), absolute derivative (middle), and local standard deviation (right) across normalized token positions (0–100%). Log probability and Sampling Discrepancy (alias Conditional Probability Curvature) display clear human-AI separation that widens in the second half, while entropy and top-k concentration show minimal divergence.

Zero-Shot Detection Metrics. We examine seven token-level statistics commonly used in zero-shot detection: (1) *log probability* $\log P(x_t|x_{<t})$, reflecting model confidence at each position; (2) *Sampling Discrepancy* (Bao et al., 2024), referred to as conditional probability curvature in the original work; (3) *rank* of the actual token among all vocabulary items; (4) *entropy* of the predicted distribution; (5) *token probability* $P(x_t|x_{<t})$ the raw conditional probability without log transformation; and (6) *top-k concentration*, the cumulative probability of the top-10 tokens.

Temporal Features. To capture the dynamics of these metrics, we compute two second-order features at each position: the *absolute first derivative* $|m_t - m_{t-1}|$, measuring the rate of change between consecutive tokens, and the *local standard deviation* within a sliding window of 20 tokens, quantifying regional volatility. For cross-sample comparison, we normalize token positions to percentiles (0–100%), enabling aggregation across texts of varying lengths.

2.2 Comparative Metric Sensitivity

We first investigate which base metrics exhibit meaningful temporal differences between human and AI-generated text. Figure 2 presents the temporal evolution of all six metrics on EvoBench, showing the raw values (left), absolute derivatives (middle), and local standard deviations (right) across normalized text positions.

Not All Metrics Capture Temporal Dynamics. We first observe that temporal discriminability varies dramatically across metrics. While *log probability* and *sampling discrepancy* show clear separation between human and AI curves, particularly in their derivative and local standard deviation, metrics such as *entropy*, *token probability*, and *top-k concentration* exhibit nearly overlapping trajectories throughout the sequence.

To quantify these differences, we measure the slope ratio: how much steeper AI's temporal trend is compared to human. As shown in Table 1, log probability exhibits the strongest temporal dynamics, with slope ratios of 2.4 for derivative and 2.6 for local standard deviation on EvoBench, and 2.4 for derivative and 1.8 for local standard deviation on MAGE. Sampling discrepancy shows comparable trends, with ratios of 2.2 (derivative) and 2.5 (local) on EvoBench, and 2.3 (derivative) and 2.2 (local) on MAGE, though it requires additional sampling overhead. In contrast, entropy, token probability, and top-k concentration show slope ratios close to 1.0, indicating minimal temporal differentiation. Based on these findings, we focus subsequent analysis on log probability, which provides the strongest temporal signal without additional computation.

2.3 Late-Stage Volatility Decay

We begin by examining the temporal dynamics of log probability, which emerges as the most informative metric throughout our analysis. Figure 1

Metric	EvoBench		MAGE	
	Deriv.	Local	Deriv.	Local
Log Probability	2.4	2.6	2.4	1.8
Sampling Discrepancy	2.2	2.5	2.3	2.2
Rank	1.2	1.2	1.0	1.1
Entropy	1.4	1.3	1.4	1.2
Token Probability	0.9	–	–	–
Top-k Concentration	1.4	1.3	0.9	1.0

Table 1: Slope ratio across metrics, measuring how many times steeper AI's temporal trend is compared to human. Higher values indicate stronger temporal dynamics. "–" indicates inconsistent trend directions between human and AI.

Dataset	Feature	H-decay	A-decay	Ratio	Gap%
EvoBench	Derivative	-9.6%	-27.1%	2.4	-32%
	Local Std	-8.9%	-26.5%	2.6	-31%
MAGE	Derivative	-7.5%	-16.7%	1.9	-24%
	Local Std	-7.3%	-16.5%	1.9	-25%

Table 2: Late-stage volatility decay of log probability features. *H-decay*/*A-decay*: percentage decrease from first to second half for human/AI-generated text. *Ratio*: how many times faster AI decays than human. *Gap%*: relative human-AI difference in the second half. AI-generated text consistently decays 1.9–2.6 faster across datasets.

visualizes the derivative and local standard deviation of log probability across normalized token positions for human-written and AI-generated text.

Two consistent trends are immediately apparent. First, for both human and AI-generated, volatility decreases as generation progresses, reflecting increasing contextual constraint. Second, and more importantly, AI-generated text stabilizes substantially faster than human text. The decline in both derivative magnitude and local variability is markedly steeper for AI-generated sequences, and the gap between the two curves widens progressively over time.

This divergence becomes most pronounced in the second half of the sequence (50%–100% positions). In this region, AI-generated text exhibits significantly smaller fluctuations in token-level log probability, while human text maintains higher volatility. We refer to this systematic asymmetry as *Late-Stage Volatility Decay*.

Quantitative Characters. Table 2 quantifies the late-stage volatility decay from two complementary perspectives. The *decay rate* measures how fast each class stabilizes from the first to second half. The *second-half gap* measures the relative difference between human and AI-generated text in later positions, directly reflecting discriminability.

On EvoBench, human text decays by only 9.6% from the first to second half in terms of derivative, while AI-generated text decays by 27.1%, a rate 2.4 times faster. This asymmetric decay widens the human-AI distinction: in the second half, the relative difference reaches 32%. The local standard deviation exhibits the same pattern, where 2.6 times faster AI decay leads to a 31% gap in the second half.

These patterns generalize across datasets. On MAGE, AI-generated text decays approximately 1.9 times faster than human text, resulting in 24–25% second-half gaps for both features. While absolute magnitudes vary between datasets, the qualitative pattern remains consistent.

Interpretation. This phenomenon reflects the autoregressive nature of LLM generation. As more context accumulates, the model's prediction distribution becomes increasingly constrained, leading to more confident outputs. Consequently, token surprise $s_t = -\log P(x_t|x_{<t})$, which measures how unexpected each token is, exhibits smaller fluctuations in later positions (Basani and Chen, 2025; Hou et al., 2025). Both the rate of change and regional variability of surprise diminish as the model's certainty grows. Human writing does not follow this optimization process. Writers introduce unexpected word choices and stylistic variations throughout their text, maintaining higher surprise volatility even in later portions. This fundamental difference manifests as the late-stage volatility decay pattern and motivates our focus on second-half statistics for detection.

2.4 Summary of Findings

Our empirical analysis demonstrates that AI-generated text exhibits accelerated stabilization in later stages of generation, leading to significantly reduced volatility in token-level surprise. This reveals that discriminative signals are not uniformly distributed across positions, but instead concentrate in the second half of sequences due to this late-stage stabilization. As a result, features computed from the second half capture substantially stronger human–AI divergence than full-sequence averages, which dilute informative late-stage signals with less discriminative early tokens. This insight directly motivates the design of our detection method.

3 Method

Motivated by our empirical analysis in Section 2, we build a simple, training-free method that explicitly exploits this late-stage temporal asymmetry. Rather than aggregating statistics over the entire sequence, we focus on the second half of the text, where the human–AI divergence in token-level dynamics is maximal.

3.1 Late-Stage Stability Metrics

Following Bao et al. (2024), we adopt a surrogate LLM for calculating token-level probabilities. Given a text $x = (t_1, \ldots, t_n)$, we query the surrogate model to obtain $p_i = P(t_i \mid t_{<i})$ and define token surprise $s_i = -\log p_i$. Based on the Late-Stage Volatility Decay phenomenon, we focus exclusively on the *second half* of the sequence to characterize late-stage stability. Let $\mathcal{H}_2 = \{i \mid i > \lfloor n/2 \rfloor\}$ denote the set of token positions in the second half. Motivated by the two temporal features examined in Section 2, we define corresponding detection features that directly quantify their dispersion in the second half.

Derivative Dispersion (DD). This feature captures the variability of surprise changes between consecutive tokens. We first compute the absolute first-order difference of surprise,

$$d_i = |s_i - s_{i-1}|, \quad i = 2, \ldots, n, \tag{1}$$

which measures how rapidly surprise fluctuates from one token to the next. We then quantify the dispersion of these change rates in the second half:

$$\mathrm{DD} = \mathrm{Std}\big(\{d_i\}_{i \in \mathcal{H}_2}\big). \tag{2}$$

AI-generated text, with its increasingly stable predictions, exhibits lower DD values as the change rates become more uniform in later positions.

Local Volatility (LV). To capture regional stability patterns that may not be reflected in pointwise statistics, we compute a sliding-window standard deviation around each position:

$$\ell_i = \mathrm{Std}\big(s_{i-w/2}, \ldots, s_{i+w/2}\big), \tag{3}$$

with window size w (default w=20). This measures how much surprise varies within a local neighborhood. We then average these local volatility values over the second half:

$$\mathrm{LV} = \mathrm{Mean}\big(\{\ell_i\}_{i \in \mathcal{H}_2}\big). \tag{4}$$

This feature is particularly sensitive to the smooth, consistent probability transitions characteristic of AI-generated text in later positions.

3.2 Combined Detection Metric

Both DD and LV features measure different aspects of volatility and yield lower values for AI-generated text due to late-stage stabilization. We negate and sum them to form our Temporal Stability Detection (TSD) score:

$$S_{\mathrm{TSD}} = -(\mathrm{DD} + \mathrm{LV}). \tag{5}$$

Higher S_{TSD} indicates greater likelihood of AI-generated text. Classification is performed by thresholding: $\hat{y} = I[S_{\mathrm{TSD}} > \tau]$, where τ is determined on validation data.

Length-Adaptive Extension. The default $\mathcal{H}_2$ uses the midpoint as the start of the late-stage window. When input length varies substantially across a deployment, this fixed split may be suboptimal: very short inputs leave too few tokens in the second half for stable estimation. We provide a simple training-free router that adjusts the tail ratio based on input length, expanding the analysis window for shorter inputs and retaining the deep tail for longer ones. The router preserves the zero-shot nature of TSD and adds no learned parameters; details and ablations are in Appendix C. All main-paper experiments use the default 50% midpoint.

3.3 Fusion with Global Metrics

Our temporal features capture *when* statistical patterns emerge (late-stage dynamics), which is orthogonal to existing methods that focus on *what* distributional anomalies exist (sequence-wide aggregation). This complementarity suggests a natural fusion strategy: combining temporal features with global statistical methods to leverage both perspectives.

We demonstrate this with Fast-DetectGPT (Bao et al., 2024), a representative global method that computes sampling discrepancy across the entire sequence. We adopt a simple additive fusion to combine both signals, preserving the zero-shot setting without introducing learned parameters:

$$S_{\mathrm{fusion}} = S_{\mathrm{TSD}} + S_{\mathrm{global}}. \tag{6}$$

The additive form yields gains comparable to multiplicative, max, and weighted alternatives (Appendix G.4), which we interpret as evidence that

the two signals are largely orthogonal, so the specific combination mechanism matters less than the fact of combining them.

By explicitly modeling late-stage volatility, our method departs from the common assumption of stationarity in token-level statistics. The resulting features are simple, interpretable, and computationally efficient, yet capture a fundamental temporal signature of machine-generated text.

4 Experiments

4.1 Experimental Setup

Evaluation Benchmarks. We conduct experiments on two complementary benchmarks. **EvoBench** (Yu et al., 2025) specifically targets the evolving nature of LLMs, comprising over 32,000 pairs across seven model families (GPT-4, GPT-4o, Claude, Gemini, LLaMA-2, LLaMA-3, Qwen) with 29 distinct model versions. EvoBench captures both version updates within model families and variations from fine-tuning or quantization, providing a rigorous test for detection robustness as source models evolve. **MAGE** (Li et al., 2024) provides broad coverage with over 30,000 test pairs spanning eight model families (including OpenAI GPT series, LLaMA, OPT, FLAN-T5, and others), representing diverse generation settings across multiple domains.

Baselines. We compare with representative zero-shot detectors from five categories: token-level statistics, including Likelihood (Solaiman et al., 2019), Entropy (Gehrmann et al., 2019), Rank (Gehrmann et al., 2019), and Log-Rank (Mitchell et al., 2023); relative scoring, including LLR (Su et al., 2023); perturbation-based detection, including Fast-Detect (Bao et al., 2024); sequential methods, including Lastde (Xu et al., 2024b), FourierGPT (Xu et al., 2024a), and Diveye (Basani and Chen, 2025); and UCE (Hou et al., 2025). All baselines use the same surrogate model for fair comparison.

Implementation Details. We use Llama-3-8B-Instruct as the surrogate model and truncate inputs to 512 tokens. TSD computes DD and LV on the second half of each sequence, with window size w=20 for LV. TSD+ denotes additive fusion with Fast-DetectGPT (Bao et al., 2024). Following prior work (Mitchell et al., 2023; Bao et al., 2024), we report AUROC as the primary metric.

4.2 Main Results

Table 3 reports detection performance on EvoBench and MAGE. We show per-family results for EvoBench and average results for MAGE, with full MAGE results in Appendix A.1.

TSD achieves the best standalone performance, reaching 83.36% AUROC on EvoBench and 71.56% on MAGE, outperforming all baselines including Fast-DetectGPT. This is achieved with only two second-half temporal statistics, without perturbation sampling or frequency-domain analysis. The gains are especially clear on recent and capable generators. On GPT-4o and GPT-4, TSD improves over Fast-DetectGPT by 9.66% and 7.35%, respectively, suggesting that late-stage volatility remains discriminative even when global distributional anomalies become less pronounced. Compared with other temporal methods, TSD also performs better than Lastde (76.66%) and Diveye (74.40%) on EvoBench, indicating the benefit of explicitly targeting late-stage stabilization.

Combining TSD with Fast-DetectGPT further improves performance. TSD+ reaches 85.37% on EvoBench and 75.20% on MAGE, showing that temporal stability and global discrepancy provide complementary signals. The gain is particularly useful on MAGE, where generators are more heterogeneous. In Appendix G.4, we further explore alternative fusion strategies and show that improved fusion can further enhance performance, suggesting that the reported TSD+ results are a simple yet effective instantiation rather than the upper bound of temporal-global integration. Bootstrap-resampled confidence intervals in Appendix F confirm that TSD significantly outperforms Fast-DetectGPT on both benchmarks ($p < 0.01$).

4.3 Ablation Study

Position Selection. To validate the importance of late-stage features, we compare three position selection strategies: using only the first half, the full sequence, and only the second half of text. As shown in Table 4, second-half features substantially outperform alternatives across all metrics and datasets. On EvoBench, TSD computed from the second half achieves 83.36%, compared to 77.00% for the full sequence and only 63.59% for the first half. The performance gap between second-half and first-half features exceeds 20 points on EvoBench, directly confirming that late-stage volatility decay provides the primary discriminative signal. Using the full

Category	Method	EvoBench								MAGE
		LLaMA-3	LLaMA-2	Claude	GPT-4o	Gemini	GPT-4	Qwen	Avg	Avg
Global	Likelihood	88.84	61.68	79.70	77.80	78.89	77.80	77.40	77.44	56.81
	Entropy	28.85	57.58	33.61	35.43	38.34	38.10	30.34	37.46	51.90
	Rank	68.49	58.38	64.50	64.64	64.46	63.43	55.06	62.71	58.99
	Log-Rank	86.47	59.99	77.37	76.80	75.99	74.87	**76.10**	75.37	55.91
	LLR	66.46	49.61	61.73	60.57	59.12	58.43	63.08	59.86	51.78
	Fast-Detect	94.17	80.27	81.57	74.89	80.48	75.58	71.96	79.85	69.69
Temporal	Lastde	90.80	80.54	76.75	71.34	76.50	71.51	69.20	76.66	70.71
	FourierGPT	63.99	57.23	58.20	55.52	56.69	55.16	64.39	58.74	60.34
	Diveye	84.65	73.46	74.21	73.32	74.06	72.28	68.85	74.40	69.60
	UCE	75.62	61.35	68.72	62.04	61.28	59.38	63.63	64.57	54.17
Ours	DD	91.12	80.77	82.56	80.57	79.86	78.48	71.78	80.73	71.72
	LV	92.57	79.79	82.51	84.88	84.64	83.58	74.55	83.22	71.26
	TSD	92.33	80.91	85.20	84.55	84.06	82.93	73.58	83.36	71.56
	TSD+	**94.50**	**83.42**	**87.12**	**85.68**	**87.40**	**85.00**	74.48	**85.37**	**75.20**

Table 3: Detection performance (AUROC %) on EvoBench and MAGE. DD: Derivative Dispersion; LV: Local Volatility; TSD: DD + LV; TSD+: TSD fused with Fast-DetectGPT. Best results are **bolded**, second best are underlined. Full MAGE results in Appendix A.1; bootstrap 95% confidence intervals in Appendix F.

Position	DD		LV		TSD	
	Evo	MAGE	Evo	MAGE	Evo	MAGE
First Half	59.91	66.54	62.37	67.26	63.59	68.02
Full Seq	72.64	70.31	75.82	70.39	77.00	70.49
Second Half	**80.73**	**71.71**	**83.21**	**71.25**	**83.36**	**71.56**

Table 4: Ablation on position selection.

Base Metric	EvoBench	MAGE
Entropy	53.52	55.24
Probability	61.69	55.37
Top-10 Concentration	53.06	55.66
Rank	76.06	62.98
Sampling Discrepancy	77.66	68.97
Log Probability (Ours)	**83.36**	**71.56**

Table 5: TSD performance with different base metrics.

sequence dilutes this signal with less informative early-stage statistics, resulting in intermediate performance. A finer-grained sweep over seven position ranges (Appendix G.1) shows monotonic improvement from early to mid-late positions and a drop when the window is restricted to the last third or quarter, where too few tokens remain for reliable estimation. Additional experiments in Appendix I further confirm that the relative midpoint (50%) consistently outperforms fixed absolute positions across datasets.

Base Metric Selection. We evaluate the effectiveness of different base metrics for computing temporal features, applying the same TSD computation to each metric.

As shown in Table 5, log probability substantially outperforms all alternatives, achieving 83.36% on EvoBench compared to 77.66% for sampling discrepancy and 76.06% for rank. Entropy, probability, and top-10 concentration yield near-random performance below 62%. This result aligns with our analysis in Section 2: log probability directly measures token surprise, which exhibits the clearest late-stage volatility decay pattern due to the autoregressive generation mechanism. A mechanistic explanation for why these alternative metrics show weaker temporal divergence is provided in Appendix H. Per-base-metric DD and LV breakdowns are in Appendix G.2, and a window-size ablation for LV is in Appendix G.3.

4.4 Robustness Analysis

We further evaluate robustness along three dimensions beyond the standard EvoBench/MAGE setting, with full results reported in the appendix.

Input length stability. Like all zero-shot detectors, our method depends on having enough tokens for reliable estimation, an issue widely observed in this line of work and explicitly acknowledged by commercial systems (e.g., minimum-length requirements of 100–300 words). A length-binned comparison across all methods (Appendix B) shows that this degradation pattern is shared by every zero-shot baseline—Fast-DetectGPT, Lastde, and Diveye all lose 28–35 pp from their best to their worst bin—and that TSD+ in fact achieves the strongest absolute performance in the most chal-

lenging short-text regime (68.99% in the 0–80 token bin). A simple length-adaptive variant (Appendix C) further closes the gap on short inputs without learned parameters. We therefore view length as a property of the detection problem itself rather than a specific limitation of our temporal design.

Adversarial robustness. On RAID (Dugan et al., 2024), which includes whitespace, case, synonym, misspelling, and paraphrase attacks, TSD achieves the best average AUROC of 0.792 and ranks first on four of five attacks. It outperforms Fast-DetectGPT by 20.0 points on average, suggesting that second-order temporal statistics are less affected by local surface perturbations than curvature-based global scores. Full results are in Appendix D.

Multilingual generalization. On the multilingual M4 subset (Wang et al., 2024), covering eight languages, TSD achieves the best average AUROC of 0.904, outperforming Fast-DetectGPT at 0.838. Notably, TSD remains strong on Chinese, where Fast-DetectGPT degrades substantially, indicating that late-stage volatility is relatively robust to tokenization and language variation. Full results are in Appendix E.

5 Related Work

We review zero-shot detection for LLM-generated text, broadly grouping prior work into two streams: *global statistical methods* that aggregate distributional cues over the whole text, and *sequential/temporal methods* that exploit how token-level probability signals evolve along the generation trajectory.

Global Statistical Detection Methods. Zero-shot detection leverages surrogate LLMs to extract distributional artifacts without requiring task-specific training data. Early approaches focused on absolute token statistics, employing metrics such as entropy, perplexity, and rank histograms to identify machine-generated anomalies (Gehrmann et al., 2019; Solaiman et al., 2019; Ippolito et al., 2020). To improve robustness, perturbation-based methods emerged, measuring the local curvature of the log-likelihood landscape around the candidate text, as exemplified by DetectGPT (Mitchell et al., 2023) and its more efficient variant Fast-DetectGPT (Bao et al., 2024) and Glimpse (Bao et al., 2025). Concurrently, relative probability approaches have been developed to normalize scores through cross-model comparisons or likelihood ratios, including DetectLLM (Su et al., 2023), Binoculars (Hans et al., 2024), DNA-GPT (Yang et al., 2024a), and intrinsic dimension estimation (Tulchinskii et al., 2023). Despite their diversity, these methods primarily rely on *global aggregation*, which average statistics over the entire sequence and treat token positions as exchangeable, overlooking the critical temporal dynamics of the generation process.

Sequential and Temporal Analysis. While global statistics provide strong baselines, recent work has shifted toward analyzing the dynamic evolution of probability signals. In the frequency domain, Spectrum (Xu et al., 2024a) and MGT-Prism (Liu et al., 2025) reveal that AI-generated text exhibits consistent spectral artifacts in likelihood curves. In the time domain, Lastde (Xu et al., 2024b) mines discriminative local subsequences from probability vectors, and DivEye (Basani and Chen, 2025) quantifies the rhythmic diversity of surprisal fluctuations. Theoretical studies have further hypothesized that recursive generation leads to uncertainty contraction over time (Hou et al., 2025). While these methods may implicitly capture some late-stage signals, for instance, low-frequency components could encode long-range patterns, they do not explicitly analyze the temporal dynamics of the second half, nor do they characterize the specific features we identify. Our work provides the first systematic analysis revealing late-stage volatility decay, where AI-generated text exhibits monotonically decreasing variability as context accumulates.

6 Conclusion

We investigated *Late-Stage Volatility Decay*, a temporal signature of autoregressive generation wherein AI-generated text exhibits progressively stabilizing token-level statistics while human writing maintains higher variability throughout. We propose simple yet effective features: Derivative Dispersion and Local Volatility for AI-generated text detection. Despite their simplicity, these temporal features achieve state-of-the-art performance on standard benchmarks across different model families, and demonstrate strong complementarity with global methods, particularly excelling on longer texts where late-stage dynamics fully manifest. Our findings suggest that exploiting the temporal structure of autoregressive generation offers a promising direction for robust detection as language models continue to evolve.

Limitations

Our work has several directions for future extension. First, this paper focuses on zero-shot detection with a fixed surrogate LLM, following common practice in prior work. While this setting enables fair comparison and avoids task-specific training, future work may further examine how different surrogate model choices influence temporal signatures across domains and languages. Second, our analysis is conducted on widely used text-generation benchmarks and representative robustness settings. Extending late-stage temporal analysis to broader real-world scenarios, such as mixed human-AI writing, interactive editing, and domain-specialized generation, would further clarify how temporal stability behaves under more diverse usage patterns. Finally, we adopt a lightweight fusion strategy to demonstrate the complementarity between temporal and global signals. More sophisticated integration strategies may further improve robustness and adaptability in practical deployments.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, and 1 others. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Guangsheng Bao, Yanbin Zhao, Juncai He, and Yue Zhang. 2025. Glimpse: Enabling white-box methods to use proprietary models for zero-shot llm-generated text detection. In *The Thirteenth International Conference on Learning Representations*.

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*.

Advik Raj Basani and Pin-Yu Chen. 2025. Diversity boosts ai-generated text detection. *arXiv preprint arXiv:2509.18880*.

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.

Jiachen Fu, Chun-Le Guo, and Chongyi Li. 2025. Detectanyllm: Towards generalizable and robust detection of machine-generated text across domains and models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 11229–11238.

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

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*.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting LLMs with binoculars: zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning*.

Junji Hou, Junzhou Zhao, Shuo Zhang, and Pinghui Wang. 2025. A theoretical analysis of detecting large model-generated time series. *arXiv preprint arXiv:2511.07104*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

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.

Shengchao Liu, Xiaoming Liu, Chengzhengxu Li, Zhaohan Zhang, Guoxin Ma, Yu Lan, and Shuai Xiao. 2025. Mgt-prism: Enhancing domain generalization for machine-generated text detection via spectral alignment. *arXiv preprint arXiv:2508.13768*.

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.

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 Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12395–12412, Singapore. Association for Computational Linguistics.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. 2023. Intrinsic dimension estimation for robust detection of AI-generated texts. In *Advances in Neural Information Processing Systems*, volume 36, pages 39257–39276.

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, and 1 others. 2024. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, pages 1–66.

Yang Xu, Yu Wang, Hao An, Zhichen Liu, and Yongyuan Li. 2024a. Detecting subtle differences between human and model languages using spectrum of relative likelihood. *arXiv preprint arXiv:2406.19874*.

Yihuai Xu, Yongwei Wang, Yifei Bi, Huangsen Cao, Zhouhan Lin, Yu Zhao, and Fei Wu. 2024b. Training-free llm-generated text detection by mining token probability sequences. *arXiv preprint arXiv:2410.06072*.

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

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

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

A MAGE Results

A.1 Detailed Results

Table 6 presents the complete detection performance on the MAGE benchmark across all eight generator families.

The MAGE benchmark reveals complementary strengths across detection approaches, with different methods excelling on specific generator families due to architectural and distributional mismatches with the surrogate model.

Our temporal features achieve the strongest performance on generators that are closer to the Llama-3 surrogate in their decoding behavior. Specifically, LV attains 91.79% on OpenAI, 94.41% on GLM, and 95.45% on LLaMA, outperforming all baselines in these categories. Even the simple DD feature remains competitive (86.75%, 91.94%, and 90.76%), suggesting that derivative-based temporal dispersion captures reliable generation dynamics when the surrogate can effectively characterize token-level uncertainty.

The Human-Paraphrase category is particularly challenging because humans revise AI outputs to evade global distributional detectors. Here, TSD+ reaches 72.03%, outperforming Fast-DetectGPT (61.92%) by over 10 points. A plausible explanation is that human edits can dilute global cues while still leaving residual late-stage temporal irregularities inherited from the original model output; fusing temporal and global signals mitigates this mismatch.

For generator families that diverge substantially from decoder-only Transformers, global baselines can become more reliable. Fast-DetectGPT achieves the best performance on FLAN-T5 (76.29%) and BigScience (72.40%). Importantly, FLAN-T5 samples in MAGE are also typically shorter, which limits the available "late-stage" token budget and makes temporal cues less stable. In such cases, Fast-DetectGPT provides a complementary, less length-sensitive signal that compensates for the reduced temporal evidence, lifting TSD from 35.17% to 55.92% on FLAN-T5.

Overall, while no single method dominates every family, TSD+ achieves the best average AUROC

Category	Method	MAGE								
		OpenAI	GLM	Human-Para	FLAN	LLaMA	BigScience	OPT	EleutherAI	Avg
Global	Likelihood	89.12	91.30	60.61	30.98	92.96	37.51	31.00	21.02	56.81
	Entropy	23.08	17.98	62.83	66.65	20.30	66.22	76.39	81.76	51.90
	Rank	73.74	83.12	53.58	34.93	81.74	45.49	52.75	46.63	58.99
	Log-Rank	87.05	90.28	53.81	31.56	91.69	37.38	32.39	23.10	55.91
	LLR	72.90	74.24	36.21	36.17	78.09	39.97	40.17	36.52	51.78
	Fast-Detect	76.40	80.41	61.92	**76.29**	67.35	**72.40**	64.61	58.12	69.69
Temporal	Lastde	74.47	83.87	62.96	66.35	72.01	64.63	72.35	69.07	70.71
	FourierGPT	59.31	84.88	68.32	4.72	84.39	39.60	57.55	63.97	60.34
	Diveye	87.89	92.61	62.10	34.65	93.80	52.00	67.60	66.11	69.60
	UCE	18.81	20.02	70.60	67.05	23.48	62.86	**82.81**	**87.73**	54.17
Ours	DD	86.75	91.94	61.98	39.60	90.76	55.71	74.84	72.16	71.72
	LV	**91.79**	**94.41**	67.21	34.61	**95.45**	53.38	69.01	64.20	71.26
	TSD	90.84	94.01	65.42	35.17	94.17	53.78	71.79	67.34	71.56
	TSD+	89.13	93.19	**72.03**	55.92	88.34	64.88	72.74	65.36	**75.20**

Table 6: Full detection performance (AUROC %) on MAGE benchmark across all generator families. Human-Para refers to human-paraphrased AI text. DD: Derivative Dispersion; LV: Local Volatility; TSD: Temporal Stability Detection; TSD+: TSD fused with Fast-DetectGPT. Best results are **bolded**, second best are underlined.

(75.20%) by avoiding catastrophic failures and remaining robust across heterogeneous generators. When temporal features are strong (e.g., LLaMA), they substantially improve over global baselines; when temporal evidence is weak (notably for short or architecturally dissimilar generations), the Fast-DetectGPT component acts as a safety net. This complementarity supports our hypothesis that temporal dynamics and global distributional discrepancies capture orthogonal aspects of AI-generated text. A per-family length–performance breakdown is reported jointly with the EvoBench view in Appendix B.1.

B Length-Binned Evaluation Across Methods

Section 4 of the main paper notes a positive correlation between sequence length and TSD performance. A natural question is whether this length sensitivity is a TSD-specific weakness, or a property shared by zero-shot AI-text detection in general. To answer this, we partition the combined EvoBench + MAGE evaluation data into five token-length bins (0–80, 80–150, 150–250, 250–350, $\geq$350 tokens) and evaluate every method under identical conditions and using the same surrogate model (LLaMA-3-8B-Instruct).

Table 7 highlights three findings.

(1) Length sensitivity is field-wide, not TSD-specific. Every zero-shot detector exhibits a substantial drop in the 0–80 bin: Fast-DetectGPT loses 28.9 points relative to its $\geq$350 score; Lastde loses 31.4; Diveye loses 33.3. This is consistent with the well-documented fact that human and machine distributions overlap more in short contexts, and with the minimum-length requirements enforced by commercial systems (e.g., Turnitin: 300 words; GPTZero: $\geq$200 words; Originality.ai: 100 words). A complementary per-family view across both EvoBench and MAGE is reported in Appendix B.1, and shows the same pattern—indicating that the positive length–AUROC correlation is a property of the detection problem itself rather than an artifact of our temporal design.

(2) TSD+ is the best-performing method on short text. In the most challenging 0–80 bin, TSD+ reaches 68.99, outperforming Fast-DetectGPT (65.65, -3.34 pp), Lastde (63.61, -5.38 pp), and Diveye (61.23, -7.76 pp). DD alone (68.59) is already competitive with Fast-DetectGPT despite using no perturbation sampling.

(3) Relative degradation is comparable or smaller for TSD. Going from the $\geq$350 bin to the 0–80 bin, TSD degrades by 29.6% relative to its peak, comparable to Fast-DetectGPT (30.6%) and smaller than Diveye (35.2%) and Lastde (33.0%). The temporal framework is therefore not a uniquely fragile design on short text.

Together with the adaptive variant in Appendix C, these results reframe the short-text question: rather than being a weakness of our method, length sensitivity is a shared limitation that TSD navigates better than existing alternatives.

Method	0–80	80–150	150–250	250–350	≥350
Likelihood	63.59	67.60	79.36	68.18	63.16
Entropy	52.15	52.36	62.79	53.13	51.29
Rank	54.91	61.23	69.88	73.33	70.78
Log-Rank	62.15	66.66	77.75	66.25	61.74
LLR	53.40	56.13	64.25	56.78	52.88
Fast-DetectGPT	65.65	83.93	88.08	91.31	94.53
Lastde	63.61	85.99	88.06	91.95	94.98
UCE	54.44	56.08	79.91	61.62	54.24
FourierGPT	57.04	64.93	69.42	59.89	70.33
Diveye	61.23	81.65	90.18	91.50	94.51
DD (Ours)	68.59	87.03	91.66	93.90	95.48
LV (Ours)	67.71	86.09	**92.99**	92.78	94.10
TSD (Ours)	67.33	86.88	93.15	94.23	95.70
TSD+ (Ours)	**68.99**	**87.56**	91.99	**94.66**	**96.01**

Table 7: AUROC (%) by token-length bin on the combined EvoBench + MAGE evaluation data. All methods share the same LLaMA-3-8B-Instruct surrogate. Best per column **bolded**; second-best underlined.

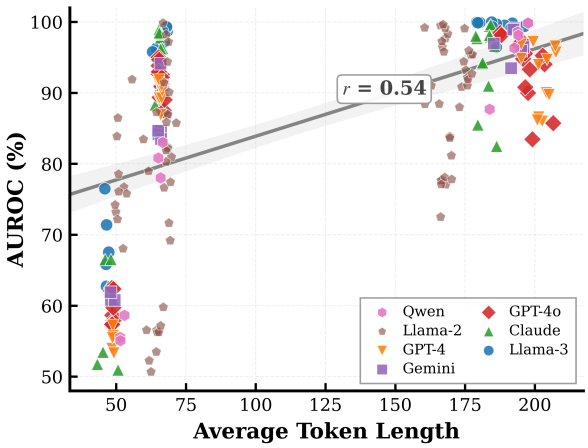

Figure 3: TSD performance (AUROC) versus average token length across seven EvoBench source families. Each point is one family; the dashed line is a linear regression fit (Pearson $r = 0.54$).

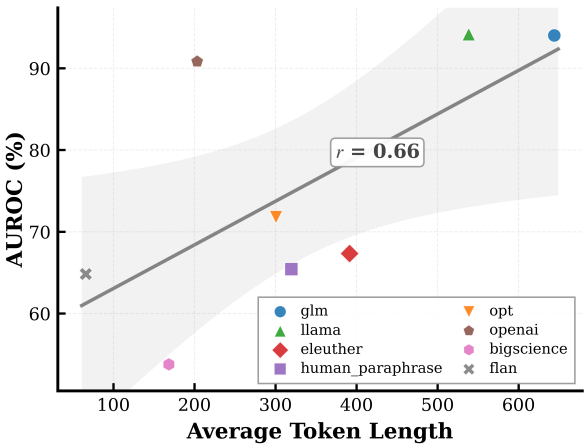

Figure 4: TSD performance (AUROC) versus average token length across eight MAGE generator families. Each point is one family; the dashed line is a linear regression fit (Pearson $r = 0.66$).

B.1 Length Sensitivity Across Generator Families

To complement the cross-method length-binned analysis in Appendix B (which pools data into token-length bins and compares all methods), we now report a per-family view: how TSD performance varies across the natural length distribution of each source. Figure 3 plots TSD AUROC against the average token length of every family in EvoBench, and Figure 4 shows the corresponding view on MAGE.

The two benchmarks show a consistent picture. On EvoBench, we observe a moderate positive correlation between average length and AUROC (Pearson $r = 0.54$): TSD reliably exceeds 82% on families whose outputs exceed 150 tokens on average, while the cases below 70% all correspond to families whose average length falls under 100 tokens. On MAGE, the same trend is even more pronounced ($r = 0.66$): families with shorter outputs (notably FLAN-T5) exhibit weaker temporal separability, which partially explains why TSD underperforms on FLAN-T5 in Table 6, while LLaMA, OpenAI, and GLM families—whose outputs are long enough to fully expose the late-stage decay—reach the highest TSD scores.

We emphasize that this per-family trend should be read jointly with the cross-method length-binned comparison in Table 7 rather than in isolation. That comparison shows that *every* zero-shot baseline degrades substantially on short inputs—Fast-DetectGPT, Lastde, and Diveye lose between 28.9 and 33.3 points from their best to their worst bin—which is consistent with the well-documented fact

that human and machine distributions overlap more in short contexts. Concretely, in the most challenging 0–80 token bin, TSD+ still achieves the highest absolute AUROC (68.99%) among all evaluated methods, and TSD's relative degradation across bins is smaller than that of Diveye and Lastde. Taken together with Figures 3 and 4, length should therefore be viewed as a property of the zero-shot detection problem itself, not as a TSD-specific limitation. This length sensitivity also motivates our fusion strategy: Fast-DetectGPT contributes a complementary global discrepancy signal that remains effective even when the available temporal evidence is limited, improving robustness across length-diverse generators. As shown in Appendix C, a simple training-free length-adaptive router further narrows the remaining gap on short inputs.

C Length-Adaptive Segmentation

The standard TSD computes statistics over the second half of the input ($r = 0.5$, where r denotes the *tail ratio*: the fraction of the sequence used as the late-stage window). On very short inputs, the second half contains too few tokens for reliable estimation. Motivated by this observation, we explore a training-free length-aware router that adjusts the tail ratio as a function of input length:

$$r(n) = \begin{cases} 0.7, & n < 80, \\ 0.6, & 80 \leq n \leq 250, \\ 0.5, & n > 250, \end{cases} \tag{7}$$

where n is the number of tokens. The intuition is that shorter texts have fewer tokens in their second half, so expanding the analysis window captures more of the available temporal pattern; longer texts focus on the deep tail where the volatility-decay signal is strongest. The router is deterministic and adds no learned parameters, preserving the zero-shot nature of TSD.

Table 8 shows that the adaptive router yields consistent improvements across all bins, with the largest gain on the shortest texts (+2.79 pp on 0–80). Two implications follow. First, the temporal framework is not inherently bounded by the 50% midpoint: a simple length-aware rule already recovers meaningful headroom on short text. Second, this leaves space for more sophisticated future extensions such as learned or continuous-position weightings.

D Adversarial Robustness on RAID

We evaluate adversarial robustness on the RAID benchmark (Dugan et al., 2024), which contains parallel human / AI examples perturbed by a set of attacks targeting different surface levels. We use five attack types: whitespace manipulation, upper/lower case perturbation, synonym substitution, perplexity-based misspelling, and paraphrase. Each split contains 3,547 labeled pairs. All methods use the same surrogate model (LLaMA-3-8B-Instruct) and inference configuration.

Observations. Table 9 summarizes the results. TSD attains the best average AUROC (0.792), outperforming every baseline. On four of the five attacks (whitespace, upper/lower, synonym, misspelling), TSD is the top performer. On paraphrase—where global distributional properties are most heavily altered—TSD (0.713) remains close to the best baseline (LLR, 0.739), while perturbation-based Fast-DetectGPT drops to 0.628.

Several patterns are worth highlighting. Curvature-style methods (Fast-DetectGPT, Lastde, Diveye) lose substantial ground under surface-level perturbations because their signals operate on absolute log-probability values, which are easily disturbed by local edits. In contrast, TSD's second-order temporal features (derivative dispersion and local volatility) measure how surprise *fluctuates* in a region; isolated token-level edits do not easily disrupt the aggregate volatility pattern across the second half.

White-box attacks targeting the second half. An informed adversary could in principle attempt to mask LSVD by injecting high-variability tokens specifically in the second half of the sequence. We note three points. First, this requires white-box knowledge of the detector. Second, injecting artificial variability while maintaining fluency is non-trivial and may introduce its own detectable artifacts. Third, most zero-shot detectors share comparable white-box vulnerabilities under matched threat models. We leave a systematic study of targeted attacks on temporal features to future work; the limitation is acknowledged explicitly in the main paper's Limitations.

E Multilingual Generalization on M4

We further evaluate whether late-stage volatility decay generalizes beyond English. We use a multilingual subset of the M4 benchmark (Wang et al.,

Method	0–80	80–150	150–250	250–350	≥350
TSD (standard)	67.33	86.88	93.15	94.23	95.70
TSD Adaptive	**70.12**	**88.86**	**94.92**	**95.32**	**97.18**
Δ	+2.79	+1.98	+1.77	+1.09	+1.48

Table 8: AUROC (%) of standard TSD vs. the length-adaptive variant on the combined EvoBench + MAGE data. The adaptive variant improves performance in every bin, with the largest gain on the shortest texts.

Method	Whitespace	Upper/Lower	Synonym	Misspelling	Paraphrase	Avg
Likelihood	0.727	0.735	0.825	0.727	0.643	0.731
Entropy	0.739	0.723	0.789	0.715	0.687	0.731
Rank	0.747	0.729	0.791	0.717	0.683	0.733
Log-Rank	0.752	0.757	0.843	0.747	0.684	0.757
LLR	0.765	0.766	0.839	0.749	**0.739**	<u>0.772</u>
Fast-DetectGPT	0.582	0.570	0.663	0.519	0.628	0.592
Lastde	0.626	0.634	0.720	0.586	0.667	0.647
FourierGPT	0.615	0.617	0.548	0.596	0.654	0.606
Diveye	0.626	0.617	0.634	0.653	0.554	0.617
UCE	0.752	0.753	0.813	0.745	<u>0.719</u>	0.756
TSD (Ours)	**0.786**	**0.795**	**0.855**	**0.810**	0.713	**0.792**

Table 9: AUROC under five adversarial attacks on RAID. Best per column **bolded**; second-best <u>underlined</u>.

2024) covering eight typologically diverse languages: Arabic (AR), Bulgarian (BG), Chinese (ZH), English (EN), German (DE), Indonesian (ID), Russian (RU), and Urdu (UR). Each language contains 120 human and 120 machine samples (1,920 total pairs). All methods use the same LLaMA-3-8B-Instruct surrogate.

Observations. Table 10 reports per-language results. TSD achieves the best overall average (0.904), outperforming Fast-DetectGPT (0.838), Lastde (0.831), and all baselines except being marginally above Diveye (0.887) on average. Crucially, TSD is robust across typologically distant language families: Semitic (Arabic 0.928), Slavic (Bulgarian 0.910), Sino-Tibetan (Chinese 0.925), Germanic (English 0.977, German 0.827), Austronesian (Indonesian 0.989), and Indo-Aryan (Urdu 0.993). This supports the view that late-stage volatility decay is a language-agnostic consequence of autoregressive generation rather than an English-specific artifact.

Chinese. A particularly notable observation is that Fast-DetectGPT collapses on Chinese (0.526), while TSD remains strong (0.925). One plausible explanation is that perturbation-based methods are more sensitive to tokenization properties of non-Latin scripts (e.g., BPE behavior on CJK characters), while temporal volatility features operate on the relative dynamics of the generation trajectory and are less affected.

Russian. Russian is the most challenging language for nearly every method (most below 0.70). This is likely a consequence of limited Cyrillic data in the LLaMA-3 pretraining mix, which weakens the surrogate's calibration for both human and machine Russian text. TSD's score (0.686) is the best among all evaluated methods on this language. We expect further improvement with a more language-appropriate surrogate.

F Statistical Significance

To quantify the reliability of the main results, we perform bootstrap resampling with 1,000 iterations and report 95% confidence intervals (Table 11). TSD's improvement over Fast-DetectGPT is statistically significant on both benchmarks: the 95% CIs are non-overlapping on EvoBench, and a paired bootstrap test yields $p < 0.01$ on both EvoBench and MAGE.

G Extended Ablations

G.1 Finer-grained Position Selection

Table 12 extends the position-selection ablation in the main paper to seven position ranges, providing a finer view of where the late-stage discriminative signal lies.

Two observations emerge. First, performance increases monotonically as the window shifts toward the late stage, peaking at the second half (50–100%). Second, going beyond the midpoint (e.g.,

Method	AR	BG	ZH	EN	DE	ID	RU	UR	Avg
Likelihood	0.870	**0.919**	**0.948**	0.964	0.735	0.967	0.627	**0.998**	0.879
Entropy	0.600	0.846	0.941	0.888	0.556	0.827	0.556	0.950	0.770
Log-Rank	0.864	0.912	0.952	0.966	0.719	0.964	0.628	**0.998**	0.875
LLR	0.796	0.823	0.955	0.948	0.597	0.862	0.608	0.933	0.815
Fast-DetectGPT	0.914	0.909	0.526	0.969	0.835	0.957	0.598	**0.998**	0.838
Lastde	**0.956**	0.847	0.594	0.949	0.837	0.941	0.546	0.974	0.831
FourierGPT	0.576	0.732	0.627	0.652	0.634	0.556	0.560	0.868	0.651
Diveye	0.916	0.880	0.917	0.972	**0.838**	0.971	0.684	0.920	0.887
UCE	0.649	0.759	0.937	0.852	0.614	0.834	0.536	0.891	0.759
TSD (Ours)	0.928	0.910	0.925	**0.977**	0.827	**0.989**	**0.686**	0.993	**0.904**

Table 10: AUROC across 8 languages on the M4 subset. Best per column **bolded**; second-best underlined.

Method	EvoBench	MAGE
Fast-DetectGPT	79.85 ± 0.48	69.69 ± 0.61
Lastde	76.66 ± 0.55	70.71 ± 0.64
Diveye	74.40 ± 0.52	69.60 ± 0.59
TSD (Ours)	83.36 ± 0.41	71.56 ± 0.56
TSD+ (Ours)	$\mathbf{85.37 \pm 0.38}$	$\mathbf{75.20 \pm 0.52}$

Table 11: Bootstrap 95% confidence intervals (AUROC %) over 1,000 resamples.

last third or last quarter) hurts performance because too few tokens remain to estimate the volatility features reliably. The 50% midpoint is therefore a sweet spot between signal strength (concentrating on the most discriminative late-stage region) and estimation stability (retaining sufficient tokens).

G.2 DD and LV per Base Metric

Table 13 reports DD, LV, and their combination (TSD) computed on top of each candidate base metric.

Three patterns are notable. (i) Log probability dominates across all configurations. (ii) LV consistently outperforms DD on top base metrics, suggesting that regional volatility carries slightly more discriminative information than pointwise derivative changes. (iii) DD and LV are complementary: TSD = DD + LV improves over either alone for every base metric.

G.3 Window Size

The Local Volatility (LV) feature depends on the sliding-window size w. Table 14 reports performance as w varies from 5 to 50, with $w = 20$ used as the default in all main-paper experiments.

Performance is stable across a wide range ($w \in [10, 30]$), peaking at $w = 20$. Too small a window yields noisy estimates of local volatility; too large a window over-smooths the temporal signal. The method is therefore not overly sensitive to this hyperparameter.

G.4 Fusion Strategy

The default fusion in TSD+ is a simple score addition with Fast-DetectGPT. Table 15 compares several alternative fusion strategies, including a length-adaptive variant that uses the adaptive TSD base from Appendix C.

Two findings are notable. First, standard fusion strategies (addition, multiplication, max, weighted) yield similar improvements with no single approach clearly dominating. This insensitivity actually supports our orthogonality claim: when two signals are largely complementary, the specific combination mechanism matters less than the fact of combining them. We chose score addition as the default because it is the simplest and preserves the zero-shot setting (no learned weights). Second, the length-adaptive variant achieves the best overall results, confirming that a more principled, length-aware fusion can improve over naive combination. This is because adaptive segmentation provides a stronger temporal base signal, particularly on shorter texts where the standard 50% midpoint is suboptimal.

Is there evidence of negative interference? A potential concern is whether fusing TSD with Fast-DetectGPT could cause negative interference if the two signals are partially redundant. The length-binned results in Table 7 provide a direct empirical test: TSD+ improves over both individual components in every length bin, including the 0–80 bin where the temporal signal is weakest. If partial redundancy were causing negative interference, we would expect degradation in at least one regime. The consistency across all bins provides direct evidence that the two signals are largely orthogonal: Fast-DetectGPT captures global distributional anomalies (*what* patterns exist), while TSD cap-

Position Range	DD		LV		TSD	
	Evo	MAGE	Evo	MAGE	Evo	MAGE
First 25% (0–25%)	55.82	63.19	57.45	64.08	57.90	64.51
First Half (0–50%)	59.91	66.54	62.37	67.26	63.59	68.02
Middle (25–75%)	71.25	69.13	73.88	69.47	74.62	69.70
Full Sequence	72.64	70.31	75.82	70.39	77.00	70.49
Second Half (50–100%)	**80.73**	**71.71**	**83.21**	**71.25**	**83.36**	**71.56**
Last Third (67–100%)	79.85	71.02	82.47	70.64	82.08	70.83
Last Quarter (75–100%)	77.12	69.84	79.53	69.26	78.86	69.55

Table 12: Finer-grained position-selection ablation (AUROC %). Performance increases monotonically from early to mid-late positions and peaks at the second half. Restricting to the last third or quarter hurts because too few tokens remain for reliable estimation.

Base Metric	DD		LV		TSD	
	Evo	MAGE	Evo	MAGE	Evo	MAGE
Log Probability	**80.73**	**71.72**	**83.21**	**71.26**	**83.36**	**71.56**
Sampling Discrep.	74.48	66.53	76.19	67.82	77.66	68.97
Rank	72.95	60.47	74.76	62.05	76.06	62.98
Entropy	51.48	53.51	52.96	54.77	53.52	55.24
Probability	59.52	53.82	60.89	54.48	61.69	55.37
Top-10 Concentration	50.94	54.05	52.41	55.02	53.06	55.66

Table 13: AUROC (%) of DD, LV, and TSD over each candidate base metric. Log probability dominates on both DD and LV. LV is consistently slightly stronger than DD, suggesting that regional volatility is more discriminative than pointwise derivative changes. Their combination (TSD) further improves over either alone, confirming complementarity.

Window Size w	EvoBench	MAGE
5	80.51	69.78
10	82.14	70.93
20 *(default)*	**83.36**	**71.56**
30	82.97	71.18
50	81.82	70.45

Table 14: TSD performance under varying window size w. Performance is stable in the $w \in [10, 30]$ range and peaks at $w = 20$.

Fusion Strategy	EvoBench	MAGE
TSD only	83.36	71.56
Score Addition (default TSD+)	85.37	75.20
Score Multiplication	84.18	74.08
Max Fusion	83.76	73.52
Weighted ($\alpha = 0.5$)	85.42	75.15
Weighted (val-tuned α)	85.92	75.83
Length-Adaptive TSD + Fast-Detect	**86.45**	**76.38**

Table 15: Comparison of fusion strategies (AUROC %). All standard fusion approaches yield comparable gains, supporting the orthogonality of TSD and Fast-DetectGPT signals. The length-adaptive variant achieves the best results overall.

tures temporal dynamics (*when* patterns emerge).

H Why Alternative Metrics Show Weaker Temporal Divergence

Section 2 reports that the strength of late-stage temporal divergence varies dramatically across base metrics. Here we provide a mechanistic explanation for each metric in our Section 2 comparison.

Log probability (slope ratio 2.4/2.6 on EvoBench). Token surprise $s_t = -\log P(x_t \mid x_{<t})$ directly measures how well the surrogate predicts the actually observed token. During autoregressive generation, the model's predictions for AI-generated text become increasingly aligned with the actual next token as context accumulates. This alignment strengthens monotonically, causing surprise to stabilize. Human text does not follow this optimization trajectory, so its surprise remains volatile. Log probability is the most direct measure of this prediction–observation alignment, which is why it exhibits the strongest temporal dynamics.

Sampling discrepancy (slope ratio 2.2/2.5). This metric measures the curvature of the conditional probability by comparing the observed token against a re-sampled distribution (Bao et al., 2024). It captures a related signal—as AI text stabilizes, the discrepancy between the observed and the expected token also stabilizes—yielding comparable dynamics but requiring additional sampling overhead and access to the surrogate's logits twice per position.

Rank (slope ratio 1.2/1.0). Rank undergoes a lossy discretization: many tokens with very different probabilities may share similar ranks, especially in the long tail. This compression discards fine-grained probability differences in precisely the region where human–AI temporal divergence is most pronounced, resulting in near-identical temporal trends (ratio ≈ 1.0 on MAGE).

Entropy (slope ratio 1.4/1.3). Entropy measures the spread of the entire predicted distribution, not the match between prediction and the specific observed token. Crucially, entropy can remain high even when the model correctly predicts the next token, as long as other vocabulary items retain non-negligible probability. Because both human and AI contexts lead the surrogate to produce qualitatively similar distributional shapes, entropy trajectories largely overlap.

Token probability and top-k concentration (slope ratio ≈ 1.0). These operate in probability space rather than log-probability space. The log transformation amplifies differences in low-probability regions—precisely where human text tends to place unexpected tokens. In linear probability space, these tail differences are compressed, making temporal divergence nearly invisible. Top-k concentration further discards all information outside the top-10 tokens, losing the very tail behavior that most distinguishes human from AI generation dynamics.

Summary. The strength of temporal divergence depends on how directly a metric reflects the prediction–observation alignment that tightens during autoregressive generation. Log probability captures this most faithfully. Other metrics either measure a different quantity (entropy, top-k), undergo lossy transformations (rank), or compress the informative signal (probability, top-k). This analysis explains the Section 2 ranking and suggests that, when designing new temporal features, base metrics should be evaluated on their ability to track this alignment process.

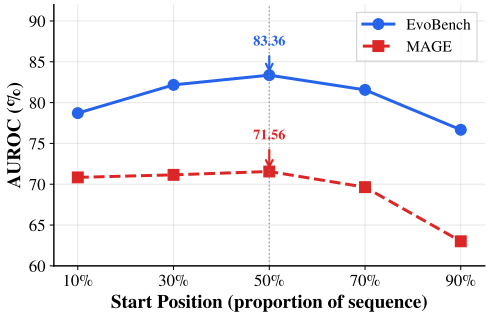

Figure 5: TSD performance versus relative start position. The midpoint (50%) achieves optimal performance on both benchmarks.

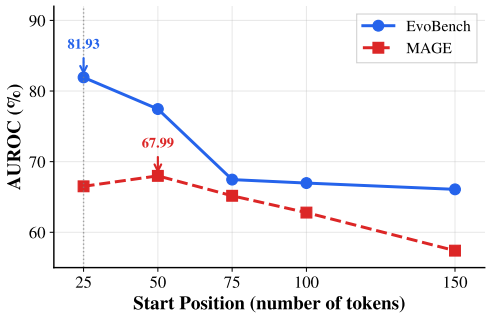

Figure 6: TSD performance versus absolute start position (number of tokens). Performance varies inconsistently across datasets, with EvoBench favoring earlier positions while MAGE shows less sensitivity.

I Impact of Start Position

While Section 3 uses the midpoint (50%) as the default start position for computing temporal features, we investigate whether alternative positions yield better performance. We conduct experiments using both relative positions (proportion of sequence) and absolute positions (fixed number of tokens).

Relative Position. Figure 5 shows TSD performance as a function of relative start position, ranging from 10% to 90% of the sequence. The results reveal an inverted-U pattern on both datasets. Starting too early (10%–30%) includes tokens where the late-stage volatility decay has not yet fully manifested, diluting the discriminative signal. Starting too late (70%–90%) excludes too many tokens, reducing statistical reliability—particularly evident on MAGE where performance drops sharply from 71.56% at 50% to 63.01% at 90%. The midpoint achieves optimal performance on both benchmarks:

83.36% on EvoBench and 71.56% on MAGE.

Absolute Position. We also evaluate using fixed token counts as start positions, shown in Figure 6. While starting at 25 tokens yields high performance on EvoBench (81.93%), this approach shows inconsistent behavior across datasets—MAGE peaks at 50 tokens (67.99%) rather than 25 tokens (66.51%). Moreover, absolute positions fail to adapt to varying sequence lengths: short sequences may have insufficient tokens beyond the start position, while long sequences underutilize available context. In contrast, the relative 50% position consistently achieves strong performance across both datasets regardless of sequence length distribution.

Based on these findings, we adopt the relative midpoint (50%) as the default start position, which provides robust performance across diverse datasets without requiring length-specific tuning.

J Complete Temporal Dynamics Visualization

Figure 7 presents the complete temporal dynamics visualization on the MAGE benchmark, complementing the EvoBench results in the main paper. The patterns are consistent across datasets: AI-generated text exhibits steeper decay in both derivative and local standard deviation of log probability, with the gap between human and AI text widening substantially in the second half of sequences.

K Surrogate Model Independence

A natural question arises: does the late-stage volatility decay phenomenon depend on the specific choice of surrogate model? To investigate this, we replicate our analysis using two additional surrogate models from different architectural families: GPT-J-6B (Wang and Komatsuzaki, 2021) and Falcon-7B (Almazrouei et al., 2023). These models differ substantially from Llama-3-8B-Instruct in training data, architecture details, and model scale, providing a rigorous test of generalization.

K.1 Temporal Dynamics Across Surrogate Models

Table 16 presents the temporal dynamics of log probability features extracted using each surrogate model on EvoBench. The late-stage volatility decay phenomenon persists across all three surrogate models with remarkably consistent patterns.

All three surrogate models show: (1) substantial negative gaps in the second half, ranging from −19% to −32%; (2) AI text decaying 1.8–2.6 times faster than human text. While Llama-3-8B produces the strongest signal (likely due to its closer architectural alignment with recent generators), GPT-J-6B and Falcon-7B still capture meaningful temporal patterns. This consistency suggests that late-stage volatility decay is a fundamental property of autoregressive generation rather than an artifact of specific surrogate-generator pairings.

Surrogate	Feature	EvoBench		MAGE	
		Gap%	Decay	Gap%	Decay
Llama-3-8B	Derivative	−32	2.4	−24	1.9
	Local Std	−31	2.6	−25	1.9
GPT-J-6B	Derivative	−22	1.8	−20	1.8
	Local Std	−22	2.0	−21	1.9
Falcon-7B	Derivative	−22	1.9	−19	1.8
	Local Std	−23	2.1	−20	1.8

Table 16: Temporal dynamics of log probability features across different surrogate models. *Gap%*: relative human-AI difference in the second half. *Decay*: ratio of AI to human decay rate. All surrogate models exhibit consistent late-stage volatility decay patterns.

Surrogate	Feature	H-decay	A-decay	Ratio	Gap%
EvoBench					
Llama-3-8B	Derivative	−9.6%	−27.1%	2.4	−32%
	Local Std	−8.9%	−26.5%	2.6	−31%
GPT-J-6B	Derivative	−9.8%	−20.1%	1.8	−22%
	Local Std	−9.1%	−20.3%	2.0	−22%
Falcon-7B	Derivative	−10.0%	−21.1%	1.9	−22%
	Local Std	−9.3%	−21.9%	2.1	−23%
MAGE					
Llama-3-8B	Derivative	−7.5%	−16.7%	1.9	−24%
	Local Std	−7.3%	−16.5%	1.9	−25%
GPT-J-6B	Derivative	−5.0%	−10.9%	1.8	−20%
	Local Std	−4.9%	−10.9%	1.9	−21%
Falcon-7B	Derivative	−5.6%	−11.4%	1.8	−19%
	Local Std	−5.3%	−11.0%	1.8	−20%

Table 17: Late-stage volatility decay details across surrogate models. *H-decay*/*A-decay*: percentage decrease from first to second half for human/AI text. The decay ratio remains consistently above 1.8 regardless of surrogate model choice.

K.2 Late-Stage Volatility Decay Details

Table 17 provides detailed late-stage statistics for each surrogate model, showing the percentage decay from first to second half for both human and AI text.

Across all configurations, AI text exhibits sub-

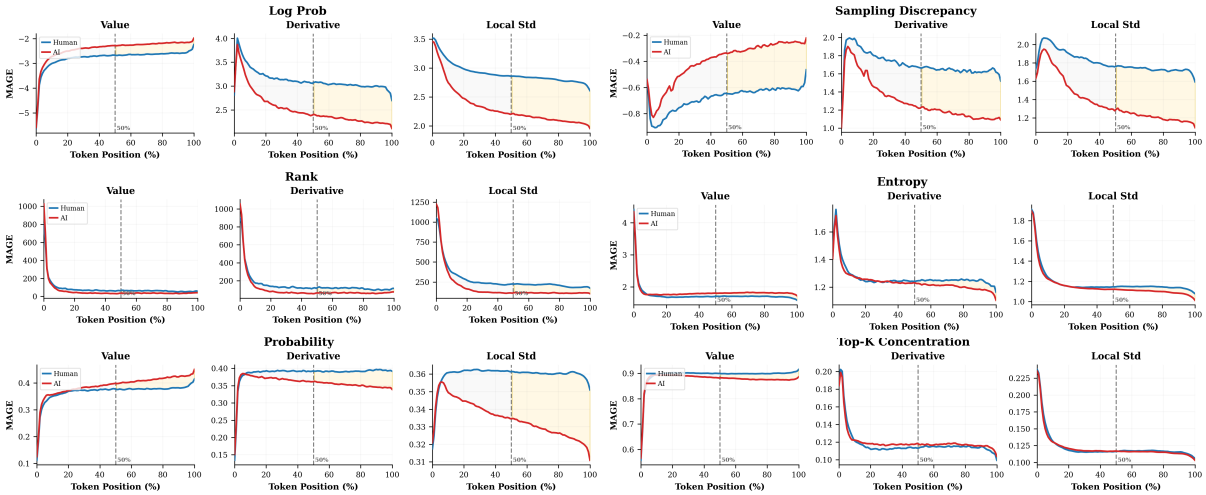

Figure 7: Temporal dynamics of seven token-level metrics on MAGE benchmark. Each metric shows three views: raw value trajectory, derivative (rate of change), and local standard deviation (volatility). The second half of sequences (highlighted in yellow) consistently exhibits larger human-AI gaps in derivative and local standard deviation features, particularly for log probability and sampling discrepancy.

stantially faster decay than human text. The decay ratios range from 1.8 to 2.6, with human text showing modest decay (5–10%) while AI text decays more dramatically (11–27%). This asymmetry is precisely what enables our detection method: the autoregressive generation process causes AI text to stabilize toward the end of sequences, while human text maintains more consistent variability throughout.

K.3 Visualization Across Surrogate Models

Figures 8 and 9 visualize the temporal dynamics using GPT-J-6B and Falcon-7B as surrogate models, respectively. Both figures exhibit the same qualitative patterns observed with Llama-3-8B: log probability shows clear separation in derivative and local standard deviation features, with the gap widening in the second half. Sampling discrepancy similarly maintains strong discriminative power across surrogate models.

The consistency across three diverse surrogate models—spanning different architectures (GPT-style, Llama-style, Falcon-style), scales (6B to 8B parameters), and training procedures—strongly supports our theoretical claim that late-stage volatility decay is an intrinsic property of autoregressive text generation. This surrogate-independence has practical implications: our detection method can leverage whichever surrogate model is most convenient or computationally efficient, without sacrificing the fundamental discriminative signal.

L Generalization to Frontier Models

To further validate the universality of late-stage volatility decay, we extend our analysis to frontier models released in late 2024 and early 2025, with particular attention to reasoning-enhanced models. We collect text samples from MIRDGE (Fu et al., 2025), a recent benchmark containing parallel human-AI pairs across 17 state-of-the-art generators.

Table 18 presents the late-stage volatility decay statistics across 13 frontier models. The phenomenon persists universally, with decay ratios ranging from 1.0 to 3.0. Critically, we find that late-stage volatility decay extends to reasoning-enhanced models (including GPT-o3-mini, DeepSeek-R1, and QwQ-Plus), which represent a distinct generation paradigm involving extended chain-of-thought processes. While these reasoning models exhibit somewhat attenuated decay ratios at 1.0–2.3 compared to standard models at 1.8–3.0, they nonetheless display the fundamental pattern: AI-generated text stabilizes more than human text toward sequence end. The extended reasoning chains may introduce additional variability that partially masks the decay signal, yet the underlying phenomenon remains detectable.

Figure 10 visualizes the temporal dynamics of a representative subset of models. Across all model families, AI-generated text consistently exhibits a steeper decay in both the derivative and the local standard deviation, with the separation between AI and human text becoming more pronounced in the

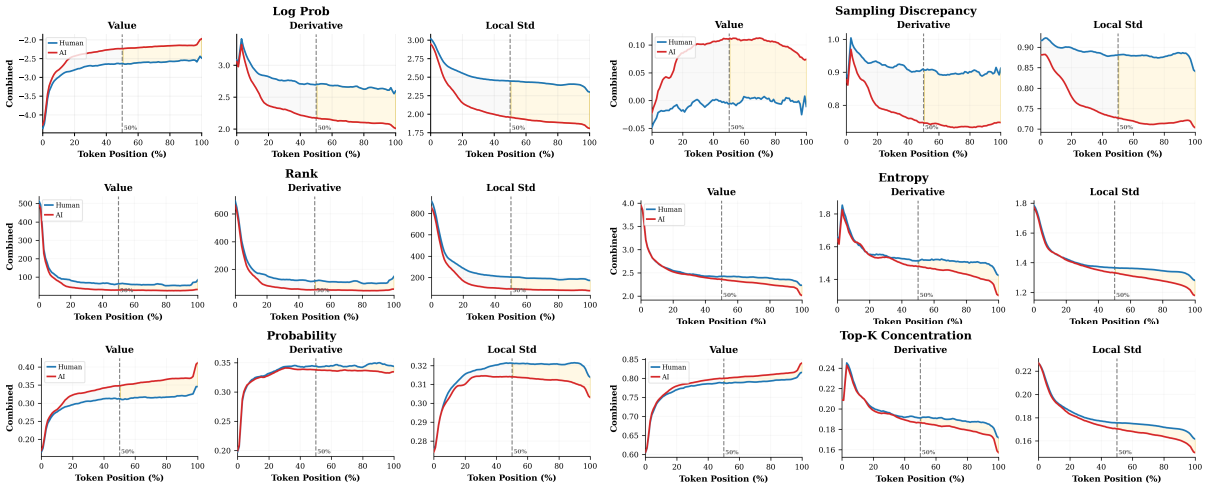

Figure 8: Temporal dynamics on EvoBench using GPT-J-6B as surrogate model. The late-stage volatility decay pattern persists: AI text (red) exhibits steeper decay in derivative and local standard deviation features compared to human text (blue), with pronounced separation in the second half (highlighted in yellow).

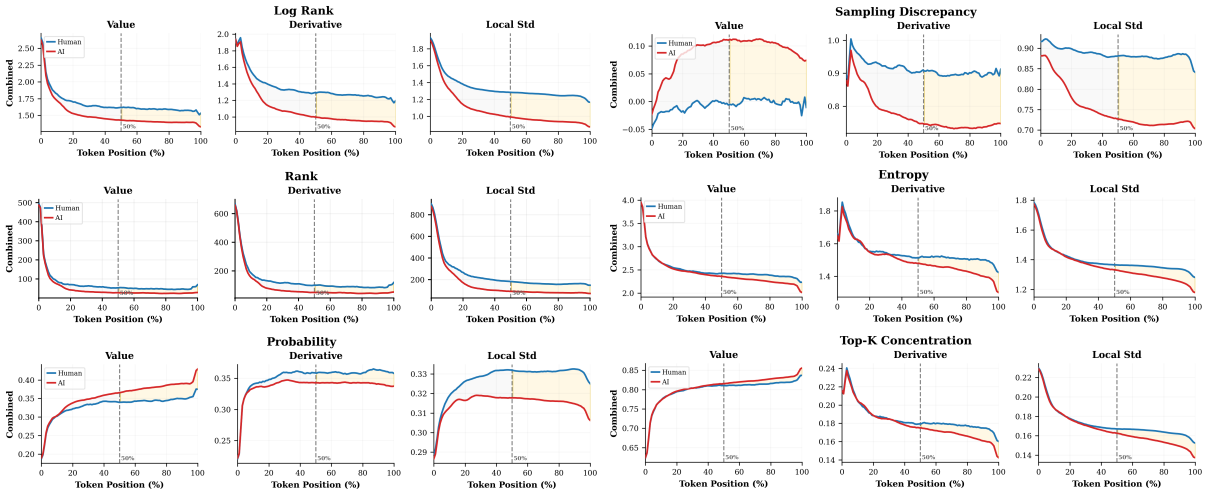

Figure 9: Temporal dynamics on EvoBench using Falcon-7B as surrogate model. Despite different model architecture and training data, the temporal patterns remain consistent with Llama-3-8B and GPT-J-6B results, confirming the surrogate-independent nature of late-stage volatility decay.

latter half of the sequence. This trend persists regardless of architectural variations or the presence of explicit reasoning mechanisms.

The consistency of late-stage volatility decay across this diverse set of frontier models—spanning different organizations, architectures, scales, and training paradigms—provides strong evidence that this phenomenon is an intrinsic property of autoregressive language generation rather than an artifact of specific model designs.

M Additional Statements

Potential Risks. Our method is designed for AI-generated text detection research and should be used responsibly. Like all detection methods, it may produce false positives or negatives and should not be the sole basis for consequential decisions.

Model Size and Computational Budget. Our experiments rely on Llama-3-8B-Instruct as the surrogate model for computing token-level statistics. All inference was conducted on one NVIDIA H100 GPU. The choice of surrogate model affects detection performance, and we acknowledge that results may vary with different model sizes or architectures.

Experimental Reproducibility. Our approach performs zero-shot detection without requiring training, which ensures stable and deterministic results. All reported point-estimate results are from

Family	Generator	Derivative				Local Std			
		H-decay	A-decay	Ratio	Gap%	H-decay	A-decay	Ratio	Gap%
OpenAI	GPT-4o	−10.3%	−26.2%	2.5	−27	−10.4%	−28.3%	2.7	−29
	GPT-4o-mini	−10.5%	−21.8%	2.1	−24	−10.5%	−23.4%	2.2	−26
	GPT-o3-mini†	−10.4%	−22.5%	2.2	−23	−10.6%	−24.3%	2.3	−25
Anthropic	Claude-3.5-Haiku	−10.5%	−24.2%	2.3	−24	−10.8%	−26.8%	2.5	−28
	Claude-3.7-Sonnet	−10.5%	−24.8%	2.4	−24	−10.6%	−26.9%	2.5	−27
Google	Gemini-2.0-Flash	−10.6%	−28.7%	2.6	−24	−10.5%	−30.6%	2.8	−26
	Gemini-2.0-Flash-Lite	−10.6%	−28.8%	2.6	−23	−10.7%	−30.2%	2.8	−26
DeepSeek	DeepSeek-V3	−11.0%	−24.4%	1.8	−31	−10.6%	−25.3%	1.9	−32
	DeepSeek-R1†	−10.0%	−10.9%	1.0	−5	−10.2%	−15.1%	1.4	−11
Meta	Llama-3.1-8B-Instruct	−11.0%	−29.2%	2.2	−43	−10.7%	−29.2%	2.4	−42
Alibaba	Qwen2.5-7B-Instruct	−11.0%	−29.6%	2.3	−33	−10.8%	−31.1%	2.4	−35
	QwQ-Plus†	−10.3%	−14.6%	1.3	−12	−10.7%	−17.9%	1.5	−18
ByteDance	Doubao-1.5-Pro-32K	−10.0%	−32.2%	3.0	−30	−10.1%	−32.7%	3.0	−30
Average (All)		−10.5%	−24.5%	2.2	−24	−10.6%	−26.3%	2.3	−27
Average (Reasoning†)		−10.2%	−16.0%	1.5	−13	−10.5%	−19.1%	1.7	−18

Table 18: Late-stage volatility decay across frontier models on MIRDGE. *H-decay*/*A-decay*: percentage decrease from first to second half for human/AI text. *Ratio*: AI-to-human decay ratio. *Gap%*: relative human-AI difference in the second half. †Reasoning-enhanced models.

single runs, as the absence of stochastic training procedures eliminates the need for multiple runs. Bootstrap-based 95% confidence intervals for the main results are reported in Appendix F.

Use of AI Assistants. We used AI assistants to aid in manuscript writing and code refinement. These tools helped with language polishing, clarity improvements, and debugging assistance. All core research ideas, experimental design, and analytical conclusions are the authors' own work.

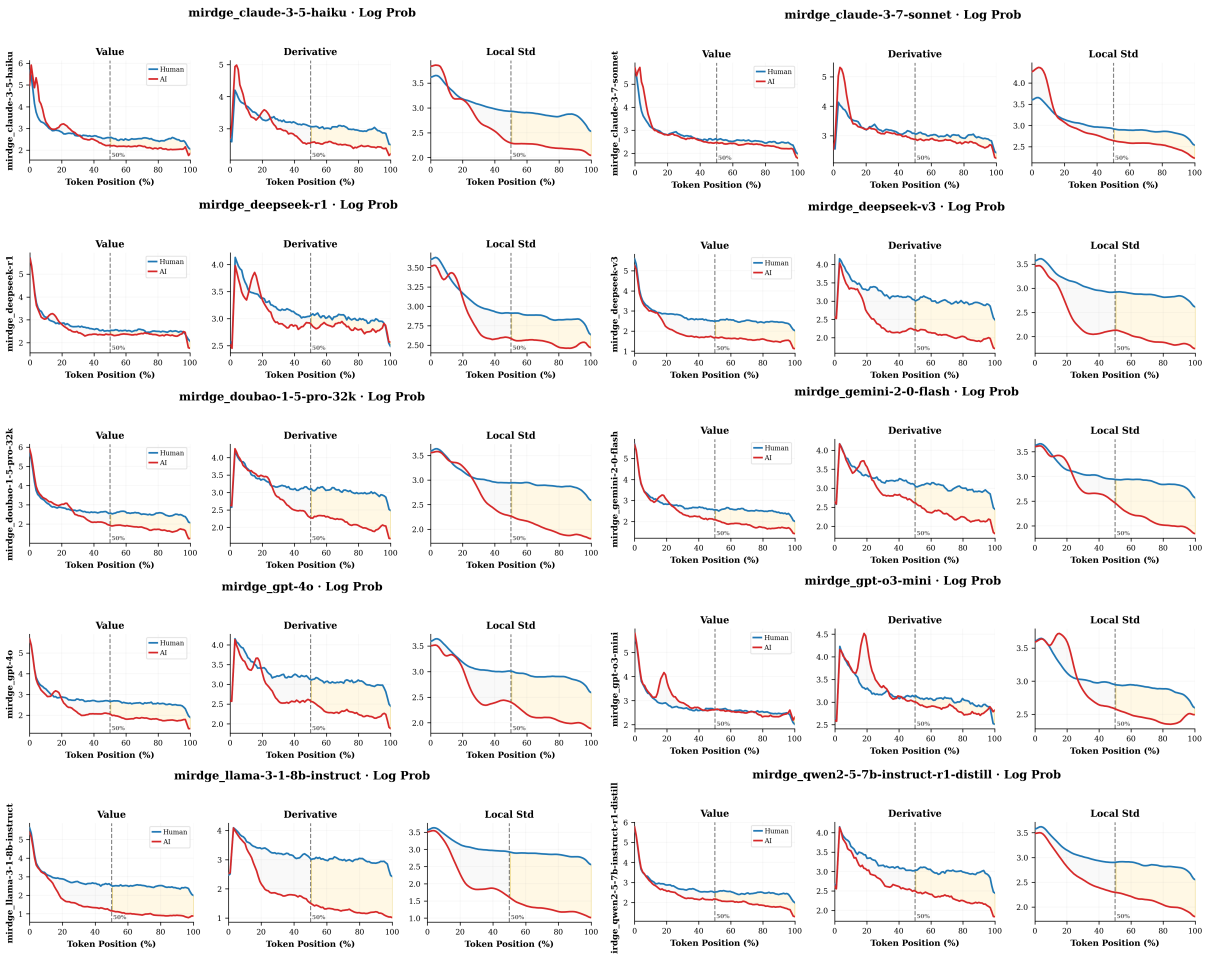

Figure 10: Temporal dynamics of log probability across frontier models from MIRDGE. Each row shows a different generator. The late-stage volatility decay pattern persists universally, including for reasoning models such as DeepSeek-R1 and GPT-o3-mini.