When Detection Tools Collide: On the Impact of Watermarking on AI-Generated Text Detectors

Michael-Andrei Panaitescu-Liess[1], Yigitcan Kaya[2], Fiza Mulla[3], Anirudh Satheesh[1], Furong Huang[1]

[1]University of Maryland, College Park, [2]University of California, Santa Barbara, [3]Indepedent Researcher

Correspondence: mpanaite@cs.umd.edu

Abstract

Detecting AI-generated text remains an open problem, typically addressed via two paradigms: watermarking at generation time and post-hoc detection. Watermarking biases generation to embed a secret, detectable signature in the token distribution, requiring deployment by LLM providers. In contrast, post-hoc detectors classify text by learning statistical differences between human and model-generated language, making them suitable for third-party use when generation access is unavailable. Despite their complementary roles, the interaction between these approaches, i.e., specifically, the impact of watermarking on post-hoc detection, remains underexplored. We show that watermarking-induced perturbations, while enabling detection by LLM providers who know the signature, can introduce a distribution shift that degrades post-hoc detectors. When degradation occurs, it worsens with increasing watermark strength, reducing detection AUC from 0.90 to near chance, revealing a fundamental breakdown. Our findings call for incorporating watermarked text into robustness evaluation for post-hoc detectors (akin to adversarial perturbations) and for developing methods that reconcile these complementary approaches.

1 Introduction

The proliferation of AI-generated text has introduced significant challenges for ensuring content authenticity, creating an urgent need for reliable detection methods with applications such as academic integrity, content moderation, and spam detection. In response, there has been substantial work on developing *post-hoc* AI text detectors that leverage statistical patterns, likelihood-based methods, and classifier-based approaches to detect machine-generated content.

Much of this prior work focuses on robustness against adversaries who actively attempt to evade detection by generating text designed to bypass these detectors by applying perturbations such as paraphrasing, synonym substitution, rewriting, and humanization techniques. These approaches aim to alter surface-level features of generated text while preserving its semantic content, weakening the signals used by detection methods.

Although prior work has largely focused on adversarial threats to detectors, in practical deployments, generated text often undergoes benign transformations introduced during or immediately after generation by LLM providers, such as watermarking, formatting, and post-processing. While not adversarial, these transformations can still alter the statistical properties of the text. For example, watermarking modifies token selection during generation by promoting or suppressing subsets of tokens, embedding a secret signature detectable only by the LLM provider. Despite their ubiquity, however, the impact of such non-adversarial interventions, particularly watermarking, on the performance of post-hoc detectors remains largely unexplored.

In this work, we close this gap through an empirical study of a realistic deployment setting where post-hoc detection is applied to text independently generated with benign watermarking, i.e., text containing embedded token-level secret signatures. Here, users lack visibility into the generation process and cannot determine whether watermarking or other generation-time interventions have been applied, as they do not have access to the secret. While watermarking is intended to facilitate provider-side AI text detection, it modifies the underlying token distribution, potentially inducing shifts in statistical signals, such as likelihood and entropy, on which post-hoc detectors often depend.

We found that the non-adversarial shift induced by watermarking creates a realistic and previously unknown robustness problem for post-hoc detectors. Our experiments show that state-of-the-art detectors routinely exhibit substantial performance differences when applied to watermarked versus

non-watermarked AI-generated text. Moreover, the severity of this effect depends on factors not observable to downstream users, including the source LLM, the watermarking scheme, and watermark strength. Taken together, these results suggest that reliably deploying post-hoc detectors in black-box settings, where decisions rely solely on the text, is both challenging and unreliable.

Concretely, we perform systematic experiments across 10 language models and 8 detectors (4 open-source supervised, 3 open-source zero-shot, and 1 black-box commercial). We evaluate two watermarking methods (KGW and Unigram) across multiple strengths and compare against baselines without watermarking as well as targeted attacks such as Adversarial Paraphrasing.

Watermarking can reduce AUROC from strong detection performance (e.g., 90%+) to near- or even sub-chance levels for zero-shot detectors, with drops of up to 74%. Even training-based detectors can degrade, with performance decreases of up to 6%. In several cases, watermarking is as damaging as or more damaging than adversarial paraphrasing, despite not being optimized for evasion.

Our results call for incorporating watermarking as a robustness stress test for AI-text detectors and for developing stronger detection methods that are robust to generation-time logit perturbations.

2 Related Work

AI text detection methods can be broadly categorized into supervised (*training-based*) classifiers and zero-shot statistical approaches. Supervised detectors, such as OpenAI RoBERTa-based models (Solaiman et al., 2019) and RADAR (Hu et al., 2023), are trained on labeled data to distinguish human and machine-generated text. In contrast, zero-shot methods rely on intrinsic properties of language models without requiring labeled detection data.

Early approaches such as GLTR analyze token-level likelihood patterns (Gehrmann et al., 2019). More recent methods exploit richer signals from model probabilities, including curvature-based scoring (Mitchell et al., 2023; Bao et al., 2023), cross-model discrepancies (Hans et al., 2024), entropy-based measures (Chen et al., 2025), token cohesiveness (Ma and Wang, 2024), and context-aware logit consistency (Huang et al., 2025).

Recent work has also considered supervised detection under distribution shift (Li et al., 2024; Liu et al., 2025).

Despite these advances, AI text detectors remain vulnerable to adversarial transformations. In particular, paraphrasing-based attacks that preserve semantic content while modifying surface-level features can significantly degrade detection performance. For example, Adversarial Paraphrasing (Cheng et al., 2025) introduces a detector-guided, training-free framework that generates human-like rewrites optimized for evasion, reducing accuracy across diverse detectors. Similar weaknesses have been observed in earlier work, where simple paraphrasing and rewriting strategies substantially lower detection performance, in some cases to near-random levels (Krishna et al., 2023; Sadasivan et al., 2025).

Watermarking methods embed detectable signals directly into the generation process of large language models to enable identification of machine-generated text. A common approach biases token selection during generation (Kirchenbauer et al., 2023).

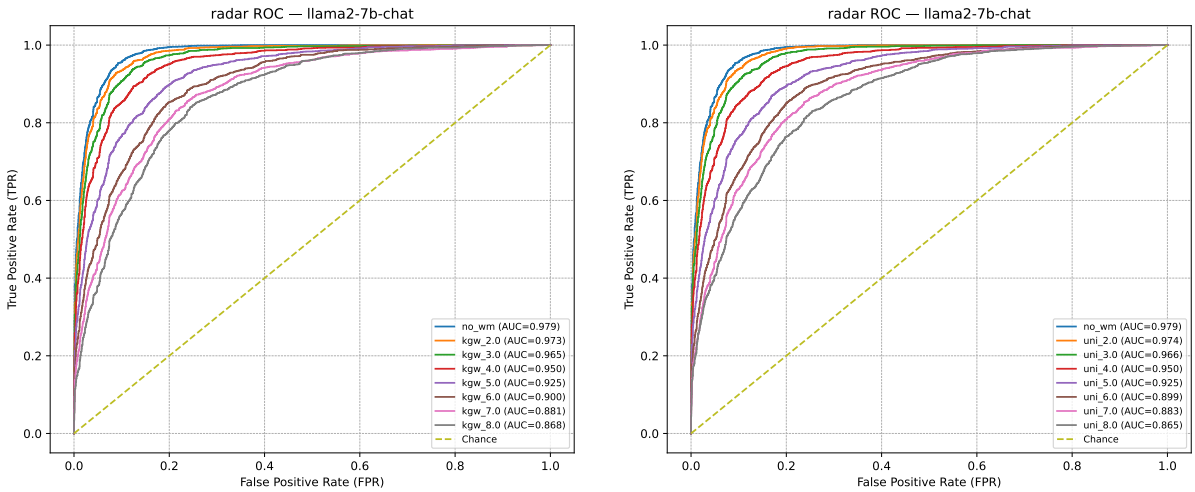
Subsequent work has studied the robustness of such watermarking schemes under editing and paraphrasing, showing that detection remains possible given sufficient text length (Kirchenbauer et al.). Other works provide theoretical guarantees on detectability (Zhao et al., 2023).

3 Observation and Experimental Setup

3.1 Watermarking Can Interfere with Independent AI-Text Detection

LLM watermarking is intended to support the identification of machine-generated text by embedding a detectable signal during generation (Kirchenbauer et al., 2023; Zhao et al., 2023). At first glance, this suggests that watermarking and AI-text detection should be naturally aligned. In this work, however, we show a counterintuitive phenomenon: *watermarking can interfere with independent AI-generated text detectors*.

This is paradoxical. Watermarking is itself a detection-oriented mechanism, yet it can sometimes make generated text harder for external detectors to identify. This distinction matters in practice. Watermark verification typically assumes access to the watermarking scheme and key (Kirchenbauer et al., 2023). In realistic deployment settings, however, text is often analyzed by third-party or commercial AI-text detectors that do not have access to the watermark verifier.



(a) KGW watermarking

(b) Unigram watermarking

Figure 1: ROC curves for RADAR under KGW and Unigram watermarking, respectively, against the baseline without watermarking (denoted by "no_wm"). Samples were generated with `Llama2-7b-chat`. We vary the watermark strengths from 2.0 to 8.0.

Our central observation is that watermarking can induce a distribution shift that degrades the performance of these detectors. In other words, although watermarking adds a signal intended for one detection mechanism, it can simultaneously weaken the signals relied upon by other detection systems. As we show later, the degradation caused by watermarking can be substantial, and in some cases is competitive with explicitly adversarial attacks.

3.2 Experimental Setup

Evaluation metrics. Following prior work on AI-text detection and adversarial paraphrasing (Cheng et al., 2025), we evaluate detector performance using Receiver Operating Characteristic (ROC) curves, Area Under the ROC Curve (AUROC), and True Positive Rate at 1% False Positive Rate (TPR@1%FPR).

Dataset. We follow the 2,000-sample subset of the MAGE dataset used in prior work on adversarial paraphrasing (Cheng et al., 2025). This allows direct comparison between the effect of watermarking and the effect of previously proposed attack-based transformations. Unless otherwise noted, all detector evaluations are conducted on this benchmark.

Detectors. We consider eight detectors in total. Four are open-source training-based detectors: MAGE (Li et al., 2024), OpenAI RoBERTa Base and OpenAI RoBERTa Large (Solaiman et al., 2019), RADAR (Hu et al., 2023), three are open-source zero-shot detectors: FastDetectGPT (Bao et al., 2023), GLTR (Gehrmann et al., 2019), Binoculars (Hans et al., 2024) and last one is the commercial Pangram detector (Pangram Labs). This combination covers both supervised and zero-shot detection paradigms, allowing us to study whether the watermarking effect is detector-specific or more general.

Watermarking methods. We evaluate two representative watermarking methods: KGW (Kirchenbauer et al., 2023) and Unigram watermarking (Zhao et al., 2023). We study multiple watermark strengths when relevant, with particular focus on stronger settings that produce clearer shifts in detector behavior.

Generation models. We study 10 language models used to generate samples: `gemma-2-9b` (Team et al., 2024), `gpt-j-6B` (Wang and Komatsuzaki, 2021), `gpt-neo-2.7B` (Black et al., 2021), `gpt2-xl` (Radford et al., 2019), `llama2-7b-chat` (Touvron et al., 2023), `llama2-13b-chat` (Touvron et al., 2023), `llama3.1-8b-instruct` (Grattafiori et al., 2024), `mistral-7b-instruct` (Jiang et al., 2023), `opt-6.7b` (Zhang et al., 2022), `pythia-12b` (Biderman et al., 2023). Using a diverse set of generators helps us determine whether the observed interaction between watermarking and detection is robust across model families rather than an artifact of a single generator. In the AI-

generated samples, we follow prior work (Cheng et al., 2025) and give the first 20 words as a prefix for the models to complete.

Comparison to adversarial paraphrasing. To contextualize the watermarking effect, we compare it against Adversarial-Paraphrase (AdvPara), using all four guidance methods used in prior work (Cheng et al., 2025). This comparison is useful because AdvPara is explicitly designed to reduce post-hoc detector performance, whereas watermarking is not.

4 Results

4.1 Watermarking Degrades Detector Performance

We first evaluate the effect of watermarking on the RADAR detector using samples generated by `Llama2-7b-chat`. As shown in Fig. 1, both KGW and Unigram watermarking reduce detection performance, with stronger watermark strengths leading to larger degradation. This trend holds consistently across both watermarking schemes.

We extend this analysis to the remaining six detectors. As shown in Fig. 2, KGW watermarking degrades performance across all detectors. The effect is particularly pronounced for zero-shot detectors (FastDetectGPT, GLTR, Binoculars), where performance drops below chance level (AUROC < 0.5), indicating that watermarked samples are effectively perceived as more human-like by these detectors than the human-written ones. Similar trends are observed for unigram watermarking (Fig. 5 in the appendix).

4.2 Comparison to Adversarial Paraphrasing

We compare watermarking against Adversarial-Paraphrase (AdvPara) using all four guidance methods from prior work and samples generated by `Llama2-7b-chat`. For RADAR (Fig. 3), watermarking at strength 8.0 causes degradation greater than AdvPara. We observe a degradation comparable or greater for other detectors (Fig. 6, appendix), where watermarking often matches or exceeds the impact of adversarial attacks despite not being explicitly optimized to evade detection.

These results highlight that watermarking, although not designed as an attack, can induce distribution shifts that are as damaging as targeted adversarial perturbations.

4.3 Aggregate Results Across Models

Table 1 summarizes results averaged over 10 generation models. We observe that watermarking moderately degrades performance for training-based detectors such as MAGE (e.g., AUROC drops by 2-6%) and RADAR (by 1-5%). In contrast, the impact on zero-shot detectors is severe: FastDetectGPT drops by up to 74%, GLTR by up to 63% under KGW, and Binoculars by up to 73%.

Watermarking can also improve performance of some detectors. For example, OpenAI RoBERTa detectors in some cases (e.g., TPR@1%FPR increases from 5.8% to 8.1% for the base model under KGW). Overall, watermarking can outperform AdvPara in degrading RADAR and the zero-shot detectors by a large margin.

4.4 Text Length Analysis

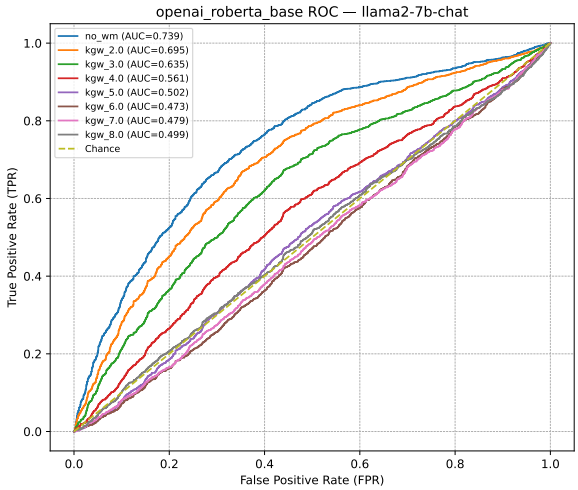
To rule out trivial explanations based on text length, we report word and token statistics in Table 2 (appendix). Watermarked and adversarial samples exhibit relatively similar average lengths to the no-watermark baseline.

4.5 Analysis on a Commercial Detector (Pangram)

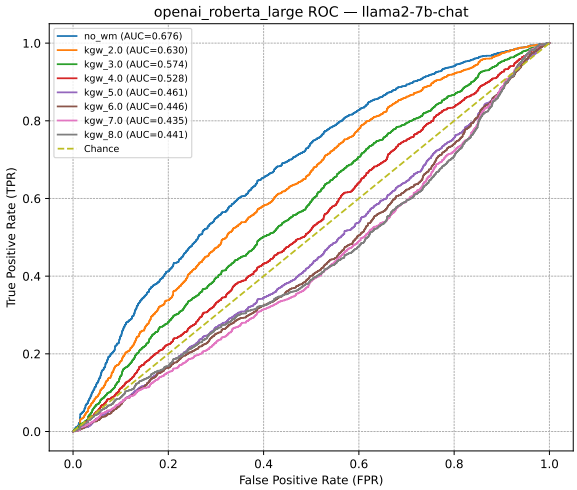
We further evaluate the Pangram detector, a state-of-the-art commercial system. As shown in Fig. 4, watermarking reduces detection performance by 25% on 100 AI-generated samples. Even after grammatical correction, performance remains degraded (11–12% drop), indicating that the effect is not solely due to surface-level grammatical artifacts. A more fine-grained per-sample analysis is provided in Fig. 7 (appendix), which shows that grammar correction only partially recovers the AI-assistance scores.

Additionally, the detector's confidence decreases substantially under watermarking, with 27–30% of samples exhibiting lower confidence scores.

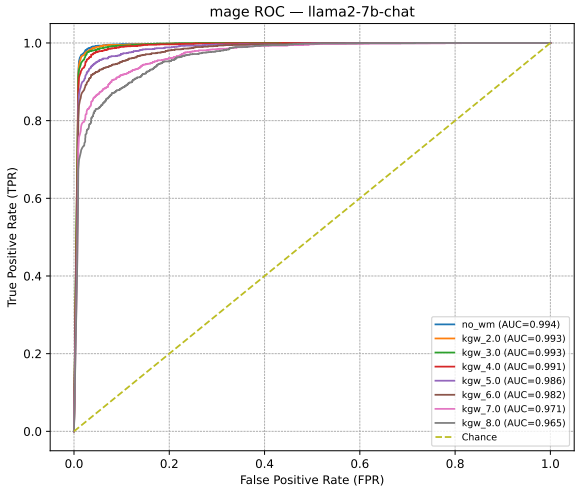
Finally, we illustrate this effect qualitatively in Fig. 8 in the appendix, where a watermarked and grammar-corrected sample is misclassified as human-written, while the original non-watermarked version is correctly identified as AI-generated.



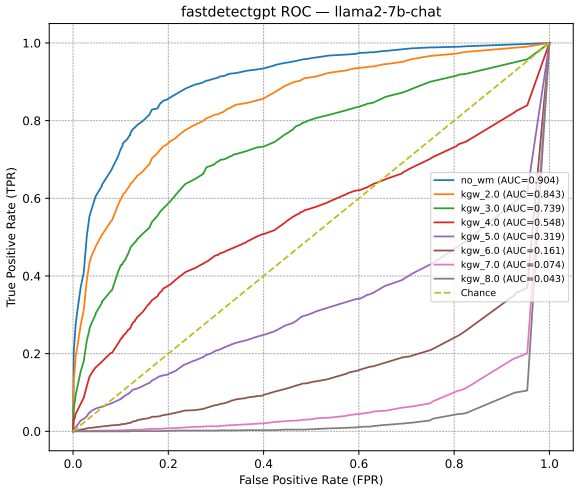
(a) OpenAI RoBERTa Base



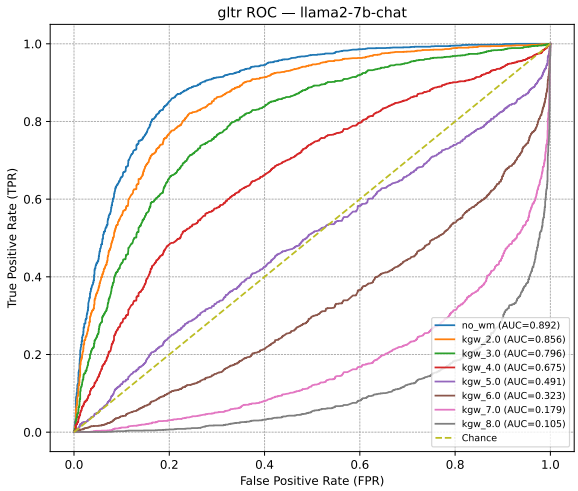
(b) OpenAI RoBERTa Large



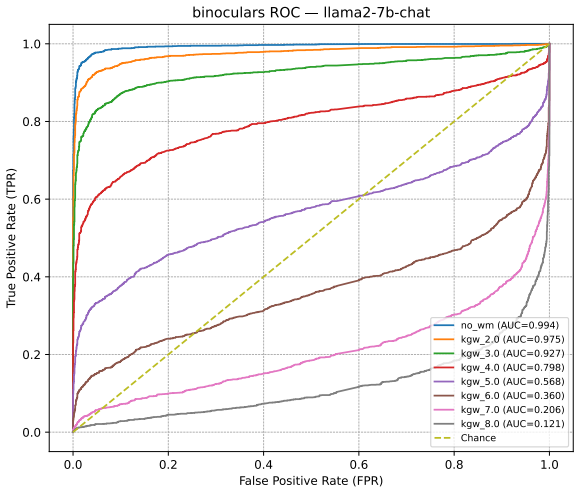
(c) MAGE



(d) FastDetectGPT



(e) GLTR



(f) Binoculars

Figure 2: ROC curves for KGW watermarking evaluated across six detectors: OpenAI RoBERTa Base, OpenAI RoBERTa Large, MAGE, FastDetectGPT, GLTR, and Binoculars. Samples were generated using `Llama2-7b-chat`. Each plot includes the no-watermark baseline and watermark strengths ranging from 2.0 to 8.0.

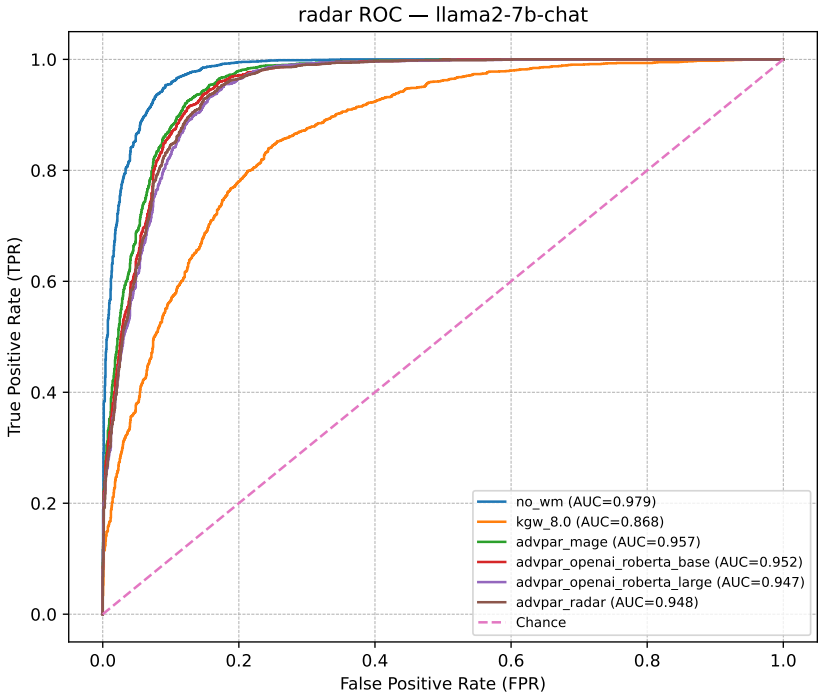
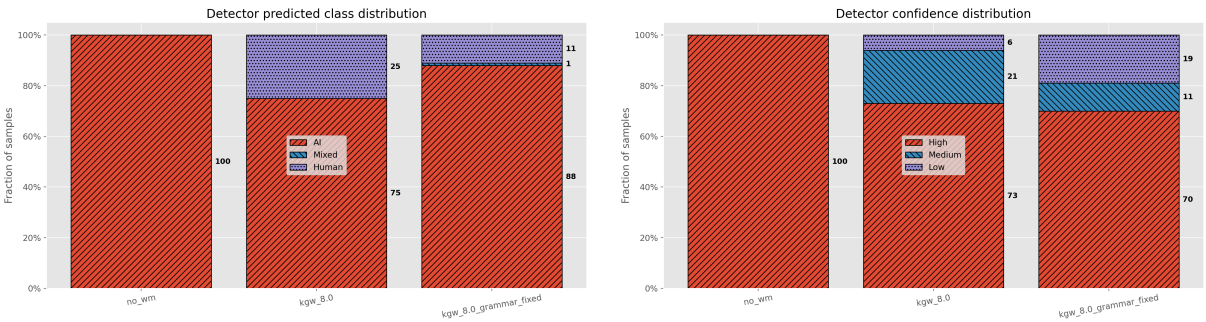


Figure 3: ROC curves under the Adversarial-Paraphrase attack (four variants) evaluated with the RADAR detector. Samples were generated using `Llama2-7b-chat`. We compare the attacks' performances against the no-watermark baseline and KGW watermarking at strength 8.0.



(a) Prediction class distribution

(b) Confidence distribution

Figure 4: Pangram detector outputs across watermarking conditions. Left: prediction class distribution for 100 samples randomly selected from the 2,000-sample dataset used in prior work. Right: confidence distribution of the Pangram detector across the same scenarios. We compare the no-watermark baseline, KGW watermarking (strength 8.0), and grammatically corrected KGW outputs (grammar fixed using `gpt-5-mini`). Results show how watermarking and post-processing affect both predicted classes and detector confidence.

Table 1: Detection performance under adversarial paraphrasing and watermarking. We report average AUROC and TPR at 1% FPR for seven detectors across the no-watermark baseline, four Adversarial-Paraphrase variants (each targeting a specific detector), and KGW and unigram watermarking at strength 8.0. Results are averaged over 10 language models used to generate the samples: `gemma-2-9b`, `gpt-j-6B`, `gpt-neo-2.7B`, `gpt2-xl`, `llama2-13b-chat`, `llama2-7b-chat`, `llama3.1-8b-instruct`, `mistral-7b-instruct`, `opt-6.7b`, and `pythia-12b`. Note that we only include the cases where the detectors on the no-watermark vs. human-written task perform significantly better than random chance (i.e., AUROC > 0.55).

Detector	Setting	Avg AUROC	Avg TPR@1%FPR
mage	no_wm	0.988	0.872
	advpar_mage	0.896	0.346
	advpar_openai_roberta_base	0.945	0.504
	advpar_openai_roberta_large	0.929	0.440
	advpar_radar	0.954	0.538
	kgw_8.0	0.966	0.774
	uni_8.0	0.927	0.561
openai_roberta_base	no_wm	0.732	0.058
	advpar_mage	0.712	0.051
	advpar_openai_roberta_base	0.349	0.001
	advpar_openai_roberta_large	0.483	0.003
	advpar_radar	0.733	0.050
	kgw_8.0	0.753	0.081
	uni_8.0	0.743	0.086
openai_roberta_large	no_wm	0.811	0.121
	advpar_mage	0.805	0.069
	advpar_openai_roberta_base	0.703	0.014
	advpar_openai_roberta_large	0.565	0.004
	advpar_radar	0.803	0.063
	kgw_8.0	0.799	0.267
	uni_8.0	0.793	0.239
radar	no_wm	0.781	0.197
	advpar_mage	0.909	0.227
	advpar_openai_roberta_base	0.900	0.213
	advpar_openai_roberta_large	0.889	0.191
	advpar_radar	0.890	0.186
	kgw_8.0	0.731	0.102
	uni_8.0	0.765	0.113
fastdetectgpt	no_wm	0.935	0.491
	advpar_mage	0.842	0.215
	advpar_openai_roberta_base	0.796	0.117
	advpar_openai_roberta_large	0.774	0.091
	advpar_radar	0.866	0.232
	kgw_8.0	0.187	0.003
	uni_8.0	0.317	0.073
gltr	no_wm	0.771	0.089
	advpar_mage	0.706	0.019
	advpar_openai_roberta_base	0.694	0.009
	advpar_openai_roberta_large	0.672	0.006
	advpar_radar	0.723	0.017
	kgw_8.0	0.133	0.001
	uni_8.0	0.171	0.010
binoculars	no_wm	0.823	0.490
	advpar_mage	0.896	0.487
	advpar_openai_roberta_base	0.878	0.422
	advpar_openai_roberta_large	0.862	0.384
	advpar_radar	0.907	0.509
	kgw_8.0	0.093	0.004
	uni_8.0	0.158	0.012

5 Conclusion

We studied the interaction between LLM watermarking and post-hoc AI-text detection. While watermarking is designed to enable detection by embedding a known signal during generation, we show that it can interfere with independent detectors. Specifically, watermarking induces a distribution shift that degrades detector performance.

These results reveal a fundamental tension between watermarking and post-hoc detection in realistic settings where external detectors are not aware of the watermarking or do not have access to the watermarking key. Our findings suggest that watermarking should be considered when evaluating the robustness of AI-text detectors.

6 Future Work

A natural next step is to develop defenses that improve robustness to watermark-based distribution shifts. One promising direction is to treat watermarking as a form of logit-level perturbation and design detectors that are invariant to such changes. In other words, in future work we plan to explore perturbation-based training or calibration methods that expose detectors to synthetic logit perturbations during training. This may help build detection systems that remain reliable across different generation conditions, including watermarking.

Limitations

Our study has several limitations. First, most of our experiments focus on not very long samples (approximately 140–150 tokens). It remains unclear whether the observed effects persist for significantly longer texts, where detection signals may behave differently.

Second, although watermarking consistently degrades performance for zero-shot detectors, the results for training-based detectors are less uniform across models. Understanding the factors that drive this variability remains an open question.

Third, our analysis is primarily empirical, so it lacks a theoretical characterization of why watermark-based perturbations can degrade external detectors' performance. Developing such a theoretical framework would help explain when and why these effects occur, and could guide the design of more robust detection systems.

References

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*.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International conference on machine learning*, pages 2397–2430. PMLR.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. *Zenodo*.

Zhihui Chen, Kai He, Yucheng Huang, Yunxiao Zhu, and Mengling Feng. 2025. Divscore: Zero-shot detection of llm-generated text in specialized domains. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19242–19264.

Yize Cheng, Vinu Sankar Sadasivan, Mehrdad Saberi, Shoumik Saha, and Soheil Feizi. 2025. Adversarial paraphrasing: A universal attack for humanizing ai-generated text. *arXiv preprint arXiv:2506.07001*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. In *Proceedings of the 57th annual meeting of the association for computational linguistics: system demonstrations*, pages 111–116.

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 and 1 others. 2024. Binoculars: Zero-shot detection of machine-generated text via cross-model discrepancy. *arXiv preprint arXiv:2401.12070*.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in neural information processing systems*, 36:15077–15095.

Zhaoheng Huang, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025. Enhancing llm text detection with retrieved contexts and logits distribution consistency. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9922–9934.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International conference on machine learning*, pages 17061–17084. PMLR.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. In *The Twelfth International Conference on Learning Representations*.

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.

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*.

Shixuan Ma and Quan Wang. 2024. Zero-shot detection of llm-generated text using token cohesiveness. *arXiv preprint arXiv:2409.16914*.

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

Pangram Labs. Pangram ai detector. https://pangram.com/. Accessed: 2026.

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.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2025. Can AI-generated text be reliably detected? stress testing AI text detectors under various attacks. *Transactions on Machine Learning Research*.

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*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

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*.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model.

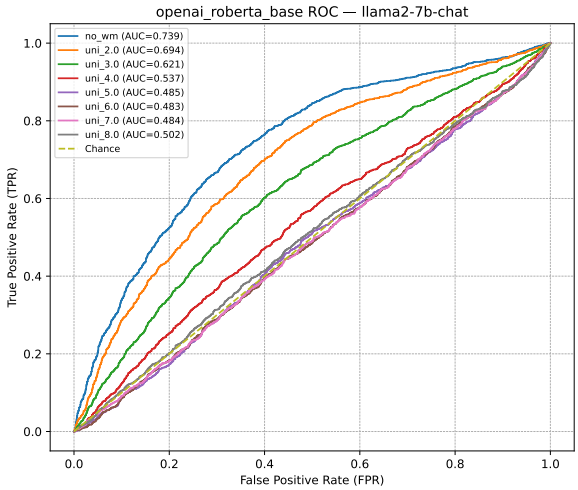
Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*.

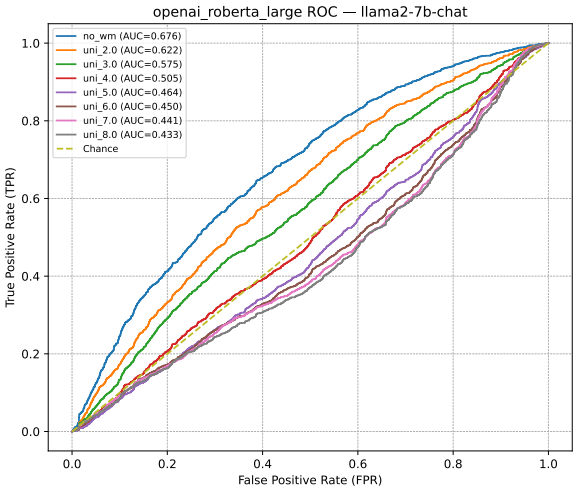
A Appendix

Table 2: Text length statistics across different scenarios. We report the mean and standard deviation of word and token counts for the no-watermark baseline, adversarial paraphrasing variants, KGW and unigram watermarking, and human-written text.

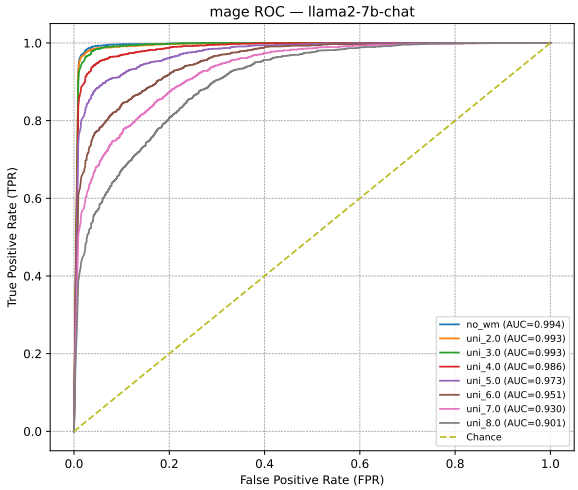
Case	Words (mean±std)	Tokens (mean±std)
no_wm	121.0±12.5	148.2±11.3
advpar_mage	109.9±32.1	137.5±39.0
advpar_openai_roberta_base	110.3±35.4	138.1±43.5
advpar_openai_roberta_large	108.5±28.8	137.2±35.8
advpar_radar	110.9±32.8	138.5±39.6
kgw_8.0	115.7±16.7	151.3±12.2
uni_8.0	112.5±17.6	149.6±13.4
human_text	142.2±29.0	177.9±38.3



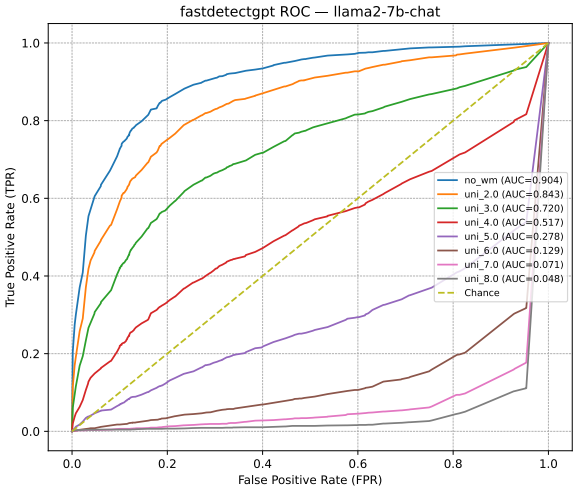
(a) OpenAI RoBERTa Base



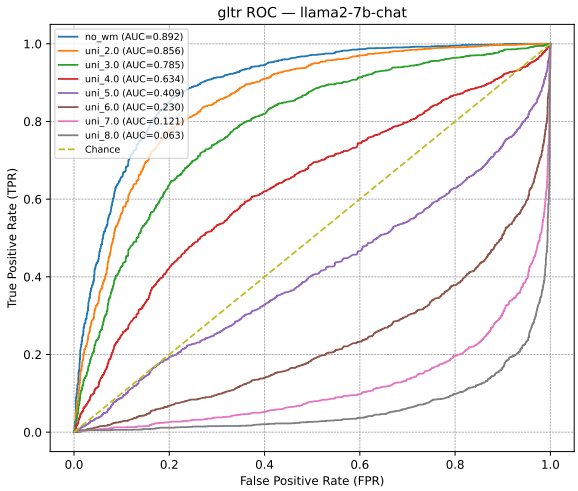
(b) OpenAI RoBERTa Large



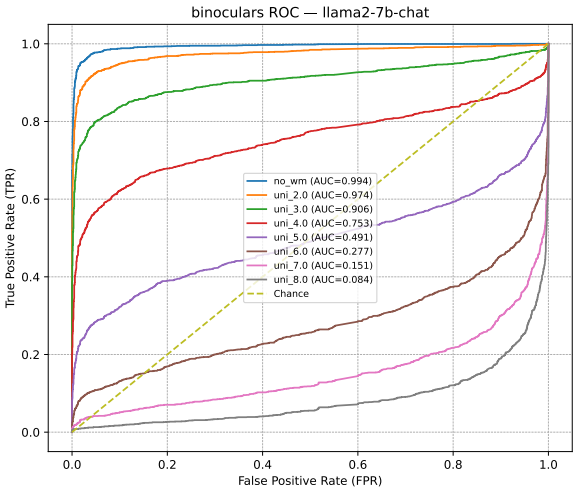
(c) MAGE



(d) FastDetectGPT

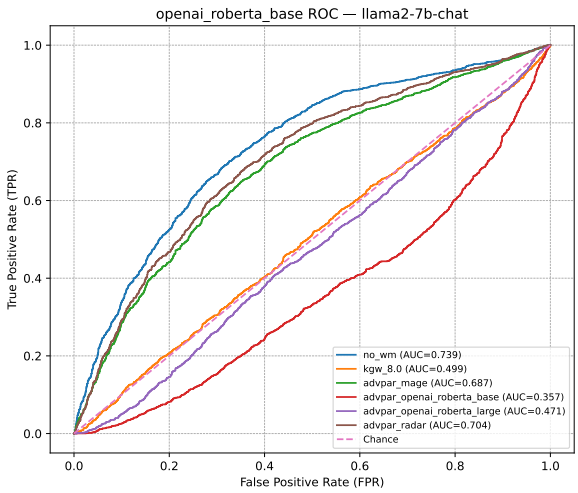


(e) GLTR

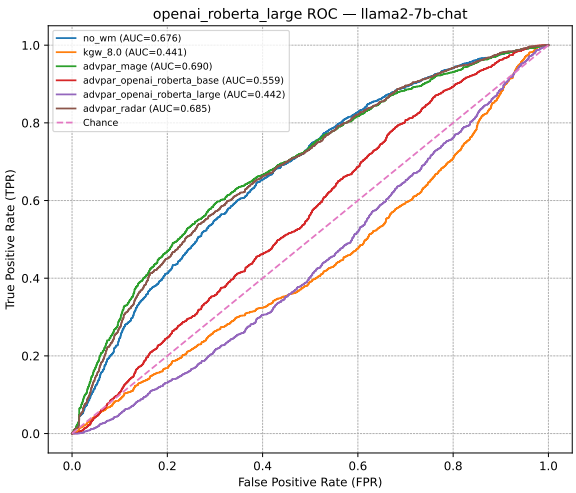


(f) Binoculars

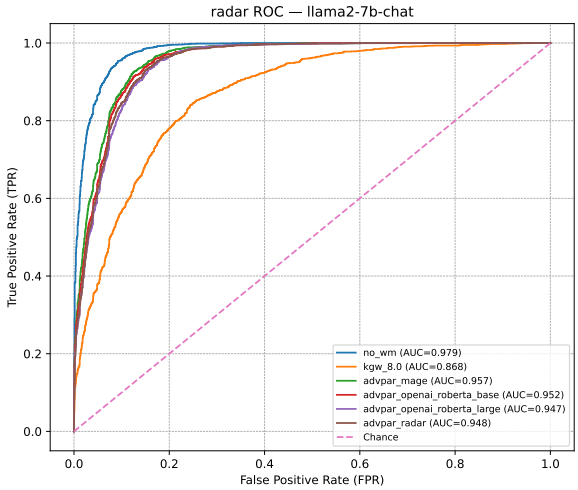
Figure 5: ROC curves for Unigram watermarking evaluated across six detectors: OpenAI RoBERTa Base, OpenAI RoBERTa Large, MAGE, FastDetectGPT, GLTR, and Binoculars. Samples were generated using `Llama2-7b-chat`. Each plot includes the no-watermark baseline and watermark strengths ranging from 2.0 to 8.0.



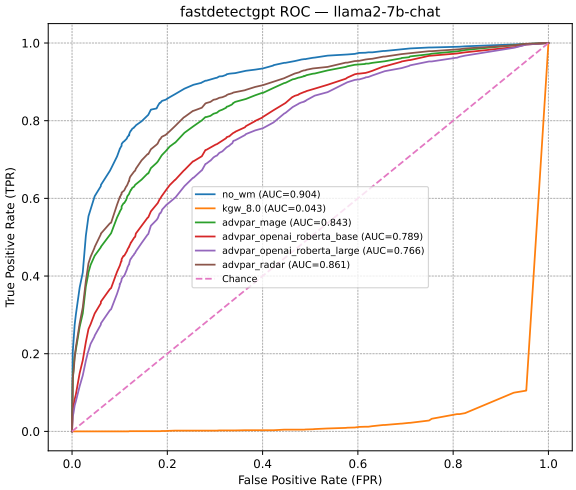
(a) OpenAI RoBERTa Base



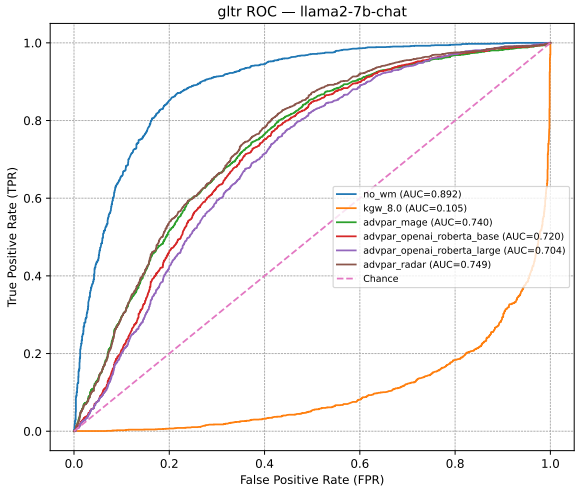
(b) OpenAI RoBERTa Large



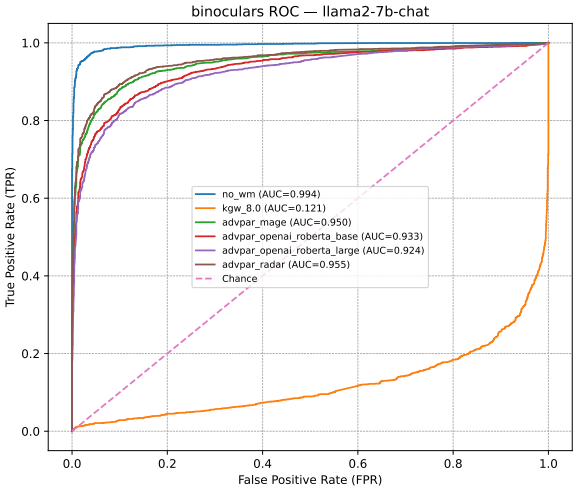
(c) RADAR



(d) FastDetectGPT



(e) GLTR



(f) Binoculars

Figure 6: ROC curves under the Adversarial-Paraphrase attack (four variants) evaluated across six detectors: OpenAI RoBERTa Base, OpenAI RoBERTa Large, RADAR, FastDetectGPT, GLTR, and Binoculars. Samples were generated using `Llama2-7b-chat`. We compare the attacks' performances against the no-watermark baseline and KGW watermarking at strength 8.0.

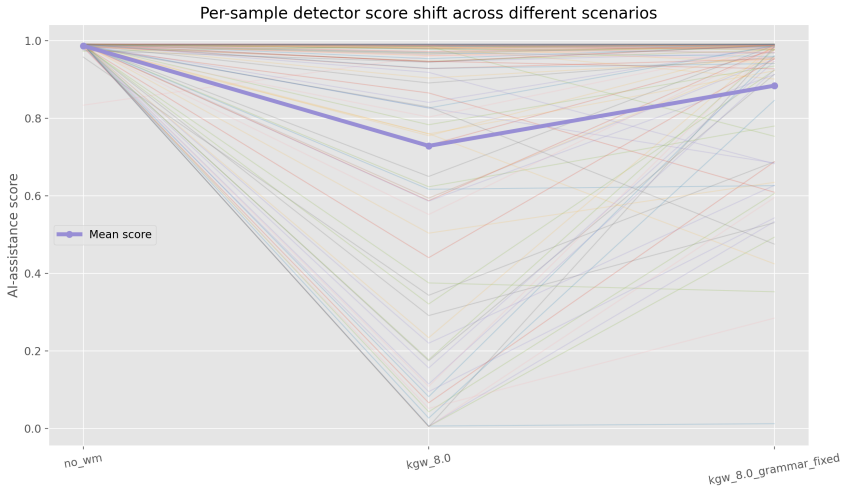
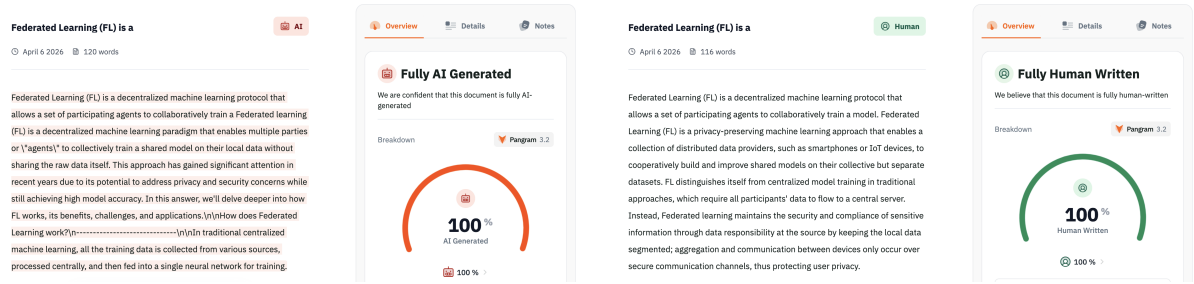


Figure 7: Changes in Pangram AI-assistance scores across watermarking conditions. Each line corresponds to a sample, showing its score under the no-watermark baseline, KGW watermarking (strength 8.0), and KGW with grammatical correction (using `gpt-5-mini`). On average, watermarking reduces the AI-assistance score, indicating weaker detection performance. While grammatical correction partially recovers the score, it does not restore it to the original no-watermark levels. This suggests that the distribution shift induced by watermarking affects detector performance, rather than the effect being attributable solely to grammatical artifacts.



(a) No watermark (predicted as AI-generated)

(b) KGW + grammar correction (predicted as human-written)

Figure 8: Example Pangram detector outputs under different conditions. Left: a non-watermarked sample is correctly identified as AI-generated. Right: the same sample after KGW watermarking (strength 8.0) and grammatical correction (using `gpt-5-mini`) is instead classified as human-written. This illustrates how watermarking combined with post-processing can still alter detector predictions.