Diversity Boosts AI-Generated Text Detection

Advik Raj Basani [1] **Pin-Yu Chen** [2]

Abstract

Detecting AI-generated text is increasingly important to prevent misuse in education, journalism, and social media, where synthetic fluency can obscure misinformation. Existing detectors often rely on likelihood heuristics or black-box classifiers, which struggle with high-quality outputs and lack interpretability. We propose DivEye, a novel detection framework that leverages surprisal-based features to capture fluctuations in lexical and structural unpredictability, a signal more prominent in human-authored text. DivEye outperforms existing zero-shot detectors by up to 33.2%, matches fine-tuned baselines, and boosts existing detectors by up to 18.7% when used as an auxiliary signal. DivEye is robust to paraphrasing and adversarial attacks, generalizes across domains, and offers interpretable insights into rhythmic unpredictability as a key indicator of AI-generated text.

Project Website & Demos: https://diveye.vercel.app/

1. Introduction

Large Language Models (LLMs) are widely used in tasks from personal assistance to content creation (Alahdab, 2024; Meyer et al., 2023; Lund et al., 2023; Hu et al., 2024; Yuan et al., 2022). While their fluency enhances utility, it also enables seamless insertion of AI-generated text into essays, articles, legal briefs, and social media, often without detection (De Giorgio et al., 2025; Papageorgiou et al., 2024; Telenti et al., 2024; Törnberg et al., 2023).

Reliable AI-text detection is vital for combating risks like misinformation, academic dishonesty, professional misconduct, and the suppression of genuine human writing (Abdali et al., 2024; Gameiro et al., 2024; Wu et al., 2025). Traditional supervised detectors (Shukla et al., 2024; Tolstykh et al., 2024; Wang et al., 2024b) rely on labeled datasets but often fail to generalize to unseen models or domains (Doughman et al., 2024; Gameiro et al., 2024), especially as new LLMs emerge. Zero-shot detectors (Bao et al., 2024; Gehrmann et al., 2019; Mitchell et al., 2023; Wang et al., 2024a) address this by leveraging statistical signals or LLMs at inference time, offering scalable, model-agnostic detection critical for maintaining platform integrity.

Contributions. We present DivEye[1], a zero-shot framework that enhances AI-text detection using diversity-based statistical features from token-level surprisal (Wilcox et al., 2025). By capturing distributional irregularities and dynamically enriching existing detectors with diverse features, DivEye improves generalization beyond static classifiers or aggregate metrics.

- *Zero-shot diversity detection:* We propose DivEye, a zero-shot framework that enhances detectors using diversity metrics based on token-level surprisal. Each feature is grounded in known differences between human and machine text, and DivEye boosts black-box detectors without requiring retraining.
- *Language & Model-agnostic detection:* DivEye is a fully zero-shot method that requires no model access or fine-tuning. It relies solely on token probability sequences from an off-the-shelf language model and generalizes across languages and model families.
- *Complementary to existing detectors:* DivEye captures statistical patterns missed by detectors relying on fine-tuned representations or classifiers, and significantly improves robustness when combined, particularly against high-quality and paraphrased adversarial text.
- *Strong generalization across domains and attacks:* Extensive evaluations across three benchmarks and varied testbeds reveal that DivEye not only achieves state-of-the-art accuracy in standard settings but also remains robust when tested on unseen domains and language models.

[1]Birla Institute of Technology and Science, KK Birla Goa Campus, India [2]IBM Research, USA. Correspondence to: Advik Raj Basani <f20221155@goa.bits-pilani.ac.in>.

Published at Data in Generative Models Workshop: The Bad, the Ugly, and the Greats (DIG-BUGS) at ICML 2025, Vancouver, Canada.

[1]The code of our method and experiments is available at https://github.com/IBM/diveye/.

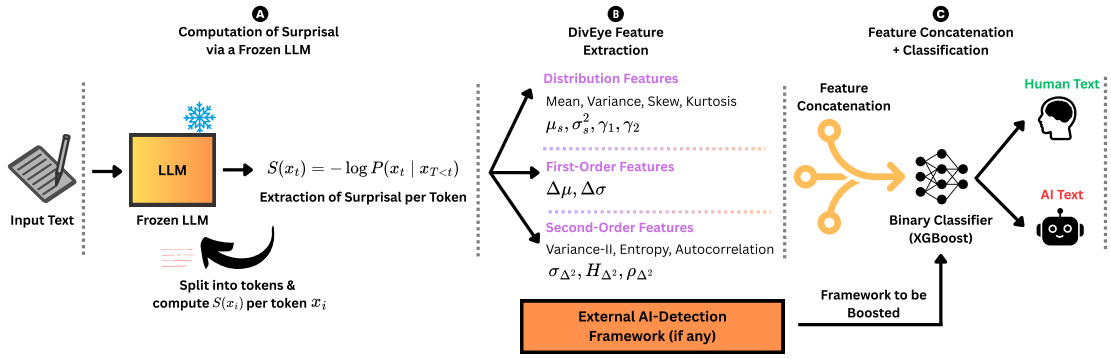


Figure 1: Overview of DivEye. DivEye extracts diversity-based features (see Section 3, Equation (3)) from token-level surprisal patterns. These features can be used in two ways: (1) as a standalone detector, or (2) as an enhancement to existing detectors, improving their performance.

2. Background and Problem Formulation

The rise of LLMs has enabled machine-generated text that closely mimics human writing by approximating the true conditional distribution of natural language, $P_{\text{human}}(x_t \mid x_{<t})$, through training on large human-written corpora (Chen et al., 2024; Lu et al., 2025). The LLM's learned distribution, $P_{\text{LLM}}(x_t \mid x_{<t})$, is used to sequentially generate tokens during inference via sampling (Zhou et al., 2024). Despite their fluency, LLMs imperfectly approximate human language ($P_{\text{LLM}} \neq P_{\text{human}}$) (Ippolito et al., 2020; Jones et al., 2024), and this subtle difference is the crux of AI text detection.

Existing detection methods fall into two categories: watermarking and zero-resource detection. Watermarking (Kirchenbauer et al., 2024; Liang et al., 2024; Liu et al., 2024a) embeds patterns in generated text but requires model access or fine-tuning, limiting use in black-box or adversarial settings. Zero-resource methods need no model knowledge and rely on statistical or learned differences between human and AI text, further divided into statistical and training-based approaches.

Training-based / Fine-tuned detection methods train classifiers, such as fine-tuned transformers on a labeled corpora of human and AI text. While these models can be accurate, they often fail to generalize across domains or against adversarial paraphrasing, especially when trained on specific generators or prompts. **Statistical / Zero-shot detection methods** refers to identifying AI-generated text without task-specific training, either by leveraging LLM probability cues or prompting LLMs directly as detectors. We discuss all related works in more detail in Appendix A.

Despite progress, AI-text detection remains unsolved. We move beyond individual token probabilities (Solaiman et al., 2019) to measure statistical diversity across token sequences, capturing variation in surprise and predictability. This reveals distributional and temporal patterns beyond likelihood metrics, as shown by the class separation in Figure 3.

3. DivEye: Methodologies

3.1. Design Hypothesis

One of the key challenges in detecting AI-generated text (Ghosal et al., 2023; Sadasivan et al., 2025) is that current models, while proficient at producing fluent language, often sacrifice variability and unpredictability for coherence and consistency.

Our hypothesis is that human-written text inherently exhibits greater stylistic diversity and unpredictability than AI-generated text. Humans make creative, spontaneous choices that introduce bursts of surprise, whereas LLMs aim to maximize sequence likelihood (Park & Choi, 2024), leading to more predictable and uniform outputs. We support this hypothesis through both intuition and empirical evidence (see Remark A).

3.2. Mathematical underpinning of DivEye

DivEye computes higher-order statistical features over surprisal sequences, capturing structural signals beyond aggregate likelihood.

Surprisal. Human language balances consistency with creative bursts, introducing novel expressions and stylistic variation. This diversity can be quantified using surprisal (Kuribayashi et al., 2025), the negative log-probability of a token given its context $S(x_t) = -\log P(x_t \mid x_1, x_2, \ldots, x_{t-1})$. For a sequence $X = x_1, \ldots, x_n$, surprisal offers a principled measure of local unpredictability based on model log-probabilities.

Rather than examining individual token surprisals in isolation, we summarize their behavior through aggregate metrics. The mean surprisal (μ_S) serves as a coarse indicator of how "expected" a text is on average: Lower values suggest closer conformity to the model's distribution, whereas

higher values signal greater unpredictability. Moreover, human writing also exhibits fluctuations in predictability due to stylistic shifts, topic changes, or bursts of creativity, motivating the use of surprisal variance (σ_S^2) alongside the mean. Formally:

$$\mu_S = \frac{1}{n}\sum_{t=1}^{n} S(x_t); \quad \sigma_S^2 = \frac{1}{n}\sum_{t=1}^{n}(S(x_t) - \mu_S)^2 \tag{1}$$

Mean and Variance are not sufficient. Mean and variance capture surprisal's central tendency and spread but miss deeper structural signals distinguishing human from AI text. Human writing often shows asymmetric surprisal distributions with bursts of creativity, causing occasional spikes in unpredictability. AI-generated text, optimized for consistency, tends toward more symmetrical distributions centered on high-probability tokens (Ippolito et al., 2020). Skewness (γ_1) measures this asymmetry, positive values indicate rare, surprising tokens typical of human writing, while kurtosis (γ_2) reflects the frequency of extreme deviations, signaling stylistic diversity. These higher-order moments enable `DivEye` to detect subtle irregularities overlooked by methods focusing only on average behavior.

$$\gamma_1 = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{S(x_t)-\mu_S}{\sigma_S}\right)^3; \quad \gamma_2 = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{S(x_t)-\mu_S}{\sigma_S}\right)^4 - 3. \tag{2}$$

Static metrics still miss temporal structure. While static surprisal metrics (mean, variance, skewness, kurtosis) summarize overall unpredictability, they miss how it evolves across a sequence, a key trait separating human from AI text. To model these dynamics, we compute the first-order difference $\Delta S_t = S(x_t) - S(x_{t-1})$, with its mean ($\Delta\mu$) and variance ($\Delta\sigma^2$) capturing stylistic volatility, such as abrupt shifts in topic or tone common in human writing.

We also compute the second-order difference $\Delta^2 S_t = \Delta S_t - \Delta S_{t-1}$ to track fluctuations in the rate of surprisal change. From this, we extract: (1) variance ($\sigma^2_{\Delta^2}$) for erratic transitions; (2) entropy ($\mathcal{H}_{\Delta^2}$) for irregularity; and (3) autocorrelation ($\rho(\Delta^2 S_t)$) for clustering of unpredictability bursts. These metrics uncover rhythmic, non-stationary patterns typical of human text but rare in the smoother, more uniform outputs of LLMs, offering a richer signal for detection. These have been formally defined in Equation (6).

We provide empirical validation of these temporal features and their individual contributions to detection performance in Appendix C.

Combinations. Collectively, `DivEye`, formalized as ($\mathcal{D}$) in Equation (3), encapsulates critical aspects of text generation that distinguish human creativity from algorithmically generated predictability, thereby serving as a robust basis for our detection framework.

$$\mathcal{D} = \{\underbrace{\mu_s, \sigma_s^2, \gamma_1, \gamma_2}_{\text{Distribution}} \oplus \underbrace{\Delta\mu, \Delta\sigma^2}_{\text{1st-Order}} \oplus \underbrace{\sigma^2_{\Delta^2}, H_{\Delta^2}, \rho_{\Delta^2}}_{\text{2nd-Order}}\} \tag{3}$$

Table 1: Performance of zero-shot and open-source fine-tuned methods on RAID. Results are aggregated over 8 domains, 12 models, and 4 decoding strategies. δ denotes the difference in AvgAcc from the benchmark leader.

Frameworks	Type	AvgAcc	δ
Desklib AI (Desklib)	Fine-tuned	94.9%	0%
e5-small-lora (Dugan et al., 2024)	Fine-tuned	93.9%	-1%
`DivEye` (Ours)	**Zero-shot**	**93.63%**	**-1.27%**
Binoculars (Hans et al., 2024)	Zero-shot	79.0%	-15%
SuperAnnotate (SuperAnnotate)	Fine-tuned	70.3%	-24.6%
RADAR (Hu et al., 2023)	Fine-tuned	65.6%	-29.3%
GLTR (Gehrmann et al., 2019)	Zero-shot	59.7%	-35.2%

$\mathcal{D}$ is a 9-dimensional vector of distributional, first-order, and second-order statistics, derived by passing text through an autoregressive LLM. These features feed a binary classifier, optionally combined with existing detector outputs. See Algorithm 1 and Appendix B for details.

`DivEye` as a booster. Existing detectors often fail against high-quality adversarial text that mimics human writing. `DivEye` provides a complementary signal by capturing statistical and temporal patterns of token-level unpredictability, orthogonal to traditional features. We enhance detectors by appending `DivEye` 's feature vector to their outputs and training a lightweight meta-classifier (e.g., XGBoost (Chen & Guestrin, 2016), Random Forest (Breiman, 2001)) on the combined representation. This fusion significantly improves performance on adversarial and out-of-distribution text, without retraining or altering the base model.

4. Experiments

Setup. We evaluate `DivEye` on diverse benchmarks covering adversarial, domain & model-specific settings. Our main tests use the RAID dataset (Dugan et al., 2024), featuring adversarial attacks, and the MAGE benchmark (Li et al., 2024), which spans eight domains and 27 LLMs to assess generalization. `DivEye` is compared against a wide range of baselines, including RADAR (Hu et al., 2023), LogRank (Ghosal et al., 2023), Entropy (Lavergne et al., 2008), FastDetectGPT (Bao et al., 2024), DetectLLM (Su et al., 2023), OpenAI Detector (Solaiman et al., 2019), Binoculars (Hans et al., 2024), RAiDAR (Mao et al., 2024), BiScope (Guo et al., 2024), and others listed on the RAID leaderboard. For implementation, we compute all `DivEye` features using GPT-2 and use a lightweight XGBoost (Chen & Guestrin, 2016) meta-classifier, either standalone (using only `DivEye`) or fused (concatenated with other detectors' features). Following each benchmark's predefined splits, we evaluate using Average Accuracy (AvgAcc), AUROC, and F1 score to capture comprehensive performance.

4.1. Performance of `DivEye`

We evaluate `DivEye` across a wide range of challenging testbeds to assess its robustness and adaptability to both

Table 2: Performance of zero-shot methods on 6 diverse testbeds from MAGE. The OOD settings examine the detection capability on texts from unseen domains or texts generated by new LLMs.

Settings	Methods	HumanAcc	MachineAcc	AvgAcc	AUROC
Testbed 2,3,4: In-distribution detection					
Arbitrary-domains & Model-specific (GPT-J)	LogRank	58.81%	63.94%	61.38%	0.67
	Entropy	76.43%	76.84%	76.64%	0.83
	DetectLLM	66.36%	62.07%	64.21%	0.72
	FastDetectGPT	62.31%	50.49%	56.4%	0.59
	Binoculars	60.11%	65.22%	62.67%	0.69
	BiScope	89.62%	84.86%	87.24%	0.93
	DivEye	90.63%	88.56%	**89.60%**	**0.97**
Fixed-domain (WP) & Arbitrary-models	LogRank	89.61%	56.15%	72.88%	0.76
	Entropy	85.96%	60.4%	73.18%	0.78
	DetectLLM	88.54%	80.77%	84.66%	0.91
	FastDetectGPT	87.25%	54.08%	70.67%	0.76
	Binoculars	80.80%	62.07%	71.44%	0.77
	BiScope	91.78%	95.27%	93.53%	0.94
	DivEye	92.22%	96.88%	**94.55%**	**0.99**
Arbitrary-domains & Arbitrary-models	LogRank	84.91%	44.47%	64.69%	0.68
	Entropy	75.68%	50.04%	62.86%	0.67
	DetectLLM	64.74%	69.02%	66.88%	0.75
	FastDetectGPT	93.65%	41.73%	67.69%	0.7
	Binoculars	76.1%	54.89%	65.49%	0.71
	BiScope	91.54%	58.70%	75.12%	0.86
	DivEye	73.72%	82.57%	**78.15%**	**0.88**
Testbed 5,6,8: Out-of-distribution detection					
Unseen Models (BLOOM-7B)	LogRank	85.84%	19.82%	52.89%	0.52
	Entropy	77.56%	34.74%	56.15%	0.59
	DetectLLM	67.85%	58.5%	63.18%	0.68
	FastDetectGPT	94.57%	13.81%	54.19%	0.54
	Binoculars	76.10%	54.89%	65.50%	0.71
	BiScope	76.72%	50.47%	63.60%	0.72
	DivEye	74.75%	77.06%	**75.91%**	**0.86**
Unseen Domains (WP)	LogRank	88.57%	49.8%	69.19%	0.74
	Entropy	78.5%	58.16%	68.33%	0.74
	DetectLLM	74.15%	71.52%	72.34%	0.79
	FastDetectGPT	95.99%	47.17%	71.58%	0.74
	Binoculars	78.93%	67.8%	73.37%	0.8
	BiScope	80.1%	78.3%	79.2%	0.86
	DivEye	94.64%	84.53%	**89.59%**	**0.97**
Unseen Domains & Unseen Models	LogRank	83.87%	43.95%	63.91%	0.68
	Entropy	74.93%	50.18%	62.55%	0.66
	DetectLLM	63.66%	67.40%	65.53%	0.73
	FastDetectGPT	93.38%	41.50%	67.44%	0.70
	Binoculars	77.85%	69.39%	73.62%	0.81
	BiScope	86%	82.58%	**84.24%**	**0.92**
	DivEye	69.75%	83.22%	76.49%	0.87

domain and model shifts. Table 1 benchmarks DivEye on the RAID dataset (Dugan et al., 2024), which spans diverse models, domains, attacks, and decoding strategies. DivEye surpasses other zero-shot methods by 13.73% and matches the performance of generative detection baselines, demonstrating strong robustness to evasive generation. As shown in Figures 6, 7, and 8, DivEye achieves high AUROCs of 0.98 and 0.93 across domains and generators, along with strong accuracy, confirming its stability and generalization across varied scenarios. Table 2 shows that DivEye outperforms existing zero-shot baselines across six MAGE testbeds, three in-distribution and three out-of-distribution, achieving an average AUROC of 0.92. These results validate DivEye's strong generalization and support the hypothesis in Section 3 that diversity-based features effectively distinguish human and machine-generated text. Appendix D.5 shows that DivEye achieves strong detection accuracy across major models like GPT-3.5-Turbo, GPT-4o, Claude-3-Opus, Sonnet, and Gemini-1.0-Pro (Brown et al., 2020; et al., 2024; Anthropic; et al., 2025), confirming its robustness and adaptability in enhancing both zero-shot and fine-tuned detection frameworks.

4.2. Robustness, Efficiency & Boosting Effectiveness

To evaluate robustness, we assess DivEye across a wide range of adversarial attacks, including paraphrasing from MAGE and adversarial perturbations from RAID. As shown in Table 3, DivEye outperforms strong baselines like the fine-tuned Longformer on MAGE by 10.15% AvgAcc and 0.11 AUROC, and surpasses zero-shot methods on RAID, notably outperforming Binoculars by 11.2%. Additional attack-specific results are detailed in Appendices D.3. Beyond accuracy, DivEye is highly efficient (see Figure 5b), requiring just 0.01 seconds per sample and achieving up to a 2971× speedup over RAiDAR, thanks to its lightweight GPT-2 backbone and fast statistical computations, making it ideal for resource-constrained settings. Furthermore, **we show that DivEye's diversity features significantly boost detection performance when fused with other detectors** like RADAR, Binoculars, DetectLLM, BiScope, and FastDetectGPT. As demonstrated in Table 5, this integration improves AUROC and AvgAcc by over 18.7%, validating that surprisal-based features offer orthogonal and complementary signals to traditional heuristics. We further explore the relative importance of DivEye and the base detector during prediction in Appendix D.4.

4.3. Ablation Studies

Model Backbone Robustness. We evaluate DivEye on MAGE Testbed 4 using different LMs for computing token-level surprisal: GPT2, GPT2-xl (Radford et al., 2019), and Falcon-7B (Almazrouei et al., 2023). As shown in Figure 5a, DivEye achieves strong AUROC scores of 0.88, 0.89, and 0.90 respectively, demonstrating consistent performance across model sizes. Even the smallest model, GPT2, remains competitive, while larger models improve human-AI separation, suggesting they better capture stylistic diversity.

Feature Importance. We analyze the contributions of DivEye 's feature groups, distributional, first-order, and second-order surprisal statistics, via XGBoost importance scores. Second-order features contribute the most (39.4%), followed by distributional (34.2%) and first-order (23.7%), supporting DivEye 's central claim about second-order features: modeling the evolution of surprisal yields stronger detection capabilities than relying solely on static measures. Detailed breakdowns are included in Appendix D.6.

5. Conclusion

We introduce DivEye, a zero-shot, model-agnostic framework for detecting AI-generated text using diversity in token-level surprisal. It is efficient, generalizes well across detectors and datasets, and we discuss its future work & limitations in Appendix G.

Impact Statement

This work adds a model-agnostic, zero-shot, scalable framework (`DivEye`) for AI-generated text detection that is model, domain, and decoding strategy-robust. By relying solely on intrinsic statistical features, our approach remains model-agnostic and does not require fine-tuning or access to LLM internals, making it broadly deployable. We hope this can facilitate responsible AI use by providing a practical tool for synthetic text identification in education, journalism, and the internet at large. At the same time, we stress the importance of cautious interpretation and advocate for its use as a complementary signal within broader content verification pipelines.

References

Aaditya Bhat. Gpt-wiki-intro, 2023. URL https://huggingface.co/datasets/aadityaubhat/GPT-wiki-intro.

Abdali, S., Anarfi, R., Barberan, C., and He, J. Decoding the ai pen: Techniques and challenges in detecting ai-generated text, 2024. URL https://arxiv.org/abs/2403.05750.

Alahdab, F. Potential impact of large language models on academic writing. *BMJ evidence-based Medicine*, 29(3): 201–202, 2024.

Almazrouei, E., Alobeidli, H., Alshamsi, A., Cappelli, A., Cojocaru, R., Debbah, M., Étienne Goffinet, Hesslow, D., Launay, J., Malartic, Q., Mazzotta, D., Noune, B., Pannier, B., and Penedo, G. The falcon series of open language models, 2023. URL https://arxiv.org/abs/2311.16867.

Anthropic. Model Card Addendum: Claude 3.5 Haiku and Upgraded Claude 3.5 Sonnet. https://assets.anthropic.com/m/1cd9d098ac3e6467/original/Claude-3-Model-Card-October-Addendum.pdf.

Arman, M. Poems dataset (nlp). https://www.kaggle.com/datasets/michaelarman/poemsdataset, 2020.

Bamman, D. and Smith, N. A. New alignment methods for discriminative book summarization, 2013.

Bao, G., Zhao, Y., Teng, Z., Yang, L., and Zhang, Y. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature, 2024. URL https://arxiv.org/abs/2310.05130.

Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document transformer, 2020. URL https://arxiv.org/abs/2004.05150.

Bhat, M. M. and Parthasarathy, S. How effectively can machines defend against machine-generated fake news? an empirical study. In Rogers, A., Sedoc, J., and Rumshisky, A. (eds.), *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pp. 48–53, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.insights-1.7. URL https://aclanthology.org/2020.insights-1.7/.

Bień, M., Gilski, M., Maciejewska, M., Taisner, W., Wisniewski, D., and Lawrynowicz, A. RecipeNLG: A cooking recipes dataset for semi-structured text generation. In *Proceedings of the 13th International Conference on Natural Language Generation*, pp. 22–28, Dublin, Ireland, December 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.inlg-1.4.

Boháček, M., Bravanský, M., Trhlík, F., and Moravec, V. Fine-grained czech news article dataset: An interdisciplinary approach to trustworthiness analysis, 2022.

Breiman, L. Random forests. *Machine Learning*, 45 (1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Chakraborty, S., Bedi, A. S., Zhu, S., An, B., Manocha, D., and Huang, F. On the possibilities of ai-generated text detection. 2023. URL https://arxiv.org/abs/2304.04736.

Chen, B., Wang, X., Peng, S., Litschko, R., Korhonen, A., and Plank, B. "seeing the big through the small": Can LLMs approximate human judgment distributions on NLI from a few explanations? In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14396–14419, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.842. URL https://aclanthology.org/2024.findings-emnlp.842/.

Chen, H., Takamura, H., and Nakayama, H. SciXGen: A scientific paper dataset for context-aware text generation. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 1483–1492, Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.128. URL https://aclanthology.org/2021.findings-emnlp.128/.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794. ACM, August 2016. doi: 10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

Chen, Y., Liu, Y., Chen, L., and Zhang, Y. DialogSum: A real-life scenario dialogue summarization dataset. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 5062–5074, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.449. URL https://aclanthology.org/2021.findings-acl.449/.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. Scaling instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

Cohere, N. T. World-class ai, at your command, 2024. URL https://cohere.com/models/command. Accessed: 2024-02-02.

De Giorgio, A., Matrone, G., and Maffei, A. Detecting large language models in exam essays. In *2025 IEEE Engineering Education World Conference (EDUNINE)*, pp. 1–6. IEEE, 2025.

Desklib. DeskLib: AI-Text-Detector-v1.01. https://huggingface.co/desklib/ai-text-detector-v1.01.

Doughman, J., Afzal, O. M., Toyin, H. O., Shehata, S., Nakov, P., and Talat, Z. Exploring the limitations of detecting machine-generated text, 2024. URL https://arxiv.org/abs/2406.11073.

Dugan, L., Hwang, A., Trhlik, F., Ludan, J. M., Zhu, A., Xu, H., Ippolito, D., and Callison-Burch, C. Raid: A shared benchmark for robust evaluation of machine-generated text detectors, 2024. URL https://arxiv.org/abs/2405.07940.

et al., B. W. Bloom: A 176b-parameter open-access multilingual language model, 2023. URL https://arxiv.org/abs/2211.05100.

et al., G. T. Gemini: A family of highly capable multimodal models, 2025. URL https://arxiv.org/abs/2312.11805.

et al., O. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Fan, A., Lewis, M., and Dauphin, Y. Hierarchical neural story generation. In Gurevych, I. and Miyao, Y. (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082/.

Fan, A., Jernite, Y., Perez, E., Grangier, D., Weston, J., and Auli, M. ELI5: Long form question answering. In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL https://aclanthology.org/P19-1346/.

Gagiano, R., Kim, M. M.-H., Zhang, X., and Biggs, J. Robustness analysis of grover for machine-generated news detection. In Rahimi, A., Lane, W., and Zuccon, G. (eds.), *Proceedings of the 19th Annual Workshop of the Australasian Language Technology Association*, pp. 119–127, Online, December 2021. Australasian Language Technology Association. URL https://aclanthology.org/2021.alta-1.12/.

Gameiro, H. D. S., Kucharavy, A., and Dolamic, L. Llm detectors still fall short of real world: Case of llm-generated short news-like posts, 2024. URL https://arxiv.org/abs/2409.03291.

Gehrmann, S., Strobelt, H., and Rush, A. M. Gltr: Statistical detection and visualization of generated text, 2019. URL https://arxiv.org/abs/1906.04043.

Ghosal, S. S., Chakraborty, S., Geiping, J., Huang, F., Manocha, D., and Bedi, A. S. Towards possibilities impossibilities of ai-generated text detection: A survey, 2023. URL https://arxiv.org/abs/2310.15264.

Greene, D. and Cunningham, P. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pp. 377–384. ACM Press, 2006.

Guerrero, J., Liang, G., and Alsmadi, I. A mutation-based text generation for adversarial machine learning applications, 2022. URL https://arxiv.org/abs/2212.11808.

Guo, B., Zhang, X., Wang, Z., Jiang, M., Nie, J., Ding, Y., Yue, J., and Wu, Y. How close is chatgpt to human experts? comparison corpus, evaluation, and detection, 2023. URL https://arxiv.org/abs/2301.07597.

Guo, H., Cheng, S., Jin, X., ZHANG, Z., Zhang, K., Tao, G., Shen, G., and Zhang, X. Biscope: AI-generated text detection by checking memorization of preceding tokens. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=Hew2JSDycr.

Hans, A., Schwarzschild, A., Cherepanova, V., Kazemi, H., Saha, A., Goldblum, M., Geiping, J., and Goldstein, T. Spotting llms with binoculars: Zero-shot detection of machine-generated text, 2024. URL https://arxiv.org/abs/2401.12070.

Hu, J., Gao, H., Yuan, Q., and Shi, G. Dynamic content generation in large language models with real-time constraints. 2024.

Hu, X., Chen, P.-Y., and Ho, T.-Y. Radar: Robust ai-text detection via adversarial learning, 2023. URL https://arxiv.org/abs/2307.03838.

Ippolito, D., Duckworth, D., Callison-Burch, C., and Eck, D. Automatic detection of generated text is easiest when humans are fooled. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1808–1822, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.164. URL https://aclanthology.org/2020.acl-main.164/.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023.

Jin, Q., Dhingra, B., Liu, Z., Cohen, W., and Lu, X. PubMedQA: A dataset for biomedical research question answering. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259/.

Jones, C. R., Trott, S., and Bergen, B. Comparing humans and large language models on an experimental protocol inventory for theory of mind evaluation (epitome). *Transactions of the Association for Computational Linguistics*, 12:803–819, 06 2024. ISSN 2307-387X. doi: 10.1162/tacl_a_00674. URL https://doi.org/10.1162/tacl_a_00674.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A watermark for large language models, 2024. URL https://arxiv.org/abs/2301.10226.

Krishna, K., Song, Y., Karpinska, M., Wieting, J., and Iyyer, M. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense, 2023. URL https://arxiv.org/abs/2303.13408.

Kumarage, T., Garland, J., Bhattacharjee, A., Trapeznikov, K., Ruston, S., and Liu, H. Stylometric detection of ai-generated text in twitter timelines, 2023. URL https://arxiv.org/abs/2303.03697.

Kuribayashi, T., Oseki, Y., Taieb, S. B., Inui, K., and Baldwin, T. Large language models are human-like internally, 2025. URL https://arxiv.org/abs/2502.01615.

Lavergne, T., Urvoy, T., and Yvon, F. Detecting fake content with relative entropy scoring. In *Proceedings of the 2008 International Conference on Uncovering Plagiarism, Authorship and Social Software Misuse - Volume 377*, PAN'08, pp. 27–31, Aachen, DEU, 2008. CEUR-WS.org.

Li, Y., Li, Q., Cui, L., Bi, W., Wang, Z., Wang, L., Yang, L., Shi, S., and Zhang, Y. Mage: Machine-generated text detection in the wild, 2024. URL https://arxiv.org/abs/2305.13242.

Liang, G., Guerrero, J., and Alsmadi, I. Mutation-based adversarial attacks on neural text detectors, 2023a. URL https://arxiv.org/abs/2302.05794.

Liang, W., Yuksekgonul, M., Mao, Y., Wu, E., and Zou, J. Gpt detectors are biased against non-native english writers, 2023b. URL https://arxiv.org/abs/2304.02819.

Liang, Y., Xiao, J., Gan, W., and Yu, P. S. Watermarking techniques for large language models: A survey, 2024. URL https://arxiv.org/abs/2409.00089.

Liu, A., Pan, L., Lu, Y., Li, J., Hu, X., Zhang, X., Wen, L., King, I., Xiong, H., and Yu, P. S. A survey of text watermarking in the era of large language models, 2024a. URL https://arxiv.org/abs/2312.07913.

Liu, Z., Yao, Z., Li, F., and Luo, B. On the detectability of chatgpt content: Benchmarking, methodology, and evaluation through the lens of academic writing, 2024b. URL https://arxiv.org/abs/2306.05524.

Lu, Y., Huang, J., Han, Y., Bei, B., Xie, Y., Wang, D., Wang, J., and He, Q. Llm agents that act like us: Accurate human behavior simulation with real-world data, 2025. URL https://arxiv.org/abs/2503.20749.

Lund, B. D., Wang, T., Mannuru, N. R., Nie, B., Shimray, S., and Wang, Z. Chatgpt and a new academic reality: Artificial intelligence-written research papers and the ethics of the large language models in scholarly publishing. *Journal of the Association for Information Science and Technology*, 74(5):570–581, 2023.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In Lin, D., Matsumoto, Y., and Mihalcea, R. (eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015/.

Mao, C., Vondrick, C., Wang, H., and Yang, J. Raidar: generative ai detection via rewriting, 2024. URL https://arxiv.org/abs/2401.12970.

Meyer, J. G., Urbanowicz, R. J., Martin, P. C., O'Connor, K., Li, R., Peng, P.-C., Bright, T. J., Tatonetti, N., Won, K. J., Gonzalez-Hernandez, G., et al. Chatgpt and large language models in academia: opportunities and challenges. *BioData mining*, 16(1):20, 2023.

Mikros, G., Koursaris, A., Bilianos, D., and Markopoulos, G. Ai-writing detection using an ensemble of transformers and stylometric features. *CEUR Workshop Proceedings*, 3496, September 2023. ISSN 1613-0073. Publisher Copyright: © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).; 2023 Iberian Languages Evaluation Forum, IberLEF 2023 ; Conference date: 26-09-2023.

Mitchell, E., Lee, Y., Khazatsky, A., Manning, C. D., and Finn, C. Detectgpt: Zero-shot machine-generated text detection using probability curvature, 2023. URL https://arxiv.org/abs/2301.11305.

MosaicML, N. T. Introducing mpt-30b: Raising the bar for open-source foundation models, 2023. URL www.mosaicml.com/blog/mpt-30b. Accessed: 2023-06-22.

Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P., and Allen, J. A corpus and cloze evaluation for deeper understanding of commonsense stories. In Knight, K., Nenkova, A., and Rambow, O. (eds.), *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 839–849, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1098. URL https://aclanthology.org/N16-1098/.

Narayan, S., Cohen, S. B., and Lapata, M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Riloff, E., Chiang, D., Hockenmaier, J., and Tsujii, J. (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Papageorgiou, E., Chronis, C., Varlamis, I., and Himeur, Y. A survey on the use of large language models (llms) in fake news. *Future Internet*, 16(8):298, 2024.

Park, B. and Choi, J. Identifying the source of generation for large language models, 2024. URL https://arxiv.org/abs/2407.12846.

Paul, S. and Rakshit, S. arxiv paper abstracts. https://www.kaggle.com/datasets/spsayakpaul/arxiv-paper-abstracts, 2021.

Pu, J., Sarwar, Z., Abdullah, S. M., Rehman, A., Kim, Y., Bhattacharya, P., Javed, M., and Viswanath, B. Deepfake text detection: Limitations and opportunities. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1613–1630, 2023. doi: 10.1109/SP46215.2023.10179387.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised

multitask learners. *OpenAI*, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. Accessed: 2024-11-15.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. SQuAD: 100,000+ questions for machine comprehension of text. In Su, J., Duh, K., and Carreras, X. (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Ren, J., Xu, H., Liu, Y., Cui, Y., Wang, S., Yin, D., and Tang, J. A robust semantics-based watermark for large language model against paraphrasing, 2024. URL https://arxiv.org/abs/2311.08721.

Sadasivan, V. S., Kumar, A., Balasubramanian, S., Wang, W., and Feizi, S. Can ai-generated text be reliably detected?, 2025. URL https://arxiv.org/abs/2303.11156.

Sanh, V., Webson, A., Raffel, C., Bach, S., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Raja, A., Dey, M., Bari, M. S., Xu, C., Thakker, U., Sharma, S. S., Szczechla, E., Kim, T., Chhablani, G., Nayak, N., Datta, D., Chang, J., Jiang, M. T.-J., Wang, H., Manica, M., Shen, S., Yong, Z. X., Pandey, H., Bawden, R., Wang, T., Neeraj, T., Rozen, J., Sharma, A., Santilli, A., Fevry, T., Fries, J. A., Teehan, R., Scao, T. L., Biderman, S., Gao, L., Wolf, T., and Rush, A. M. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

Schabus, D., Skowron, M., and Trapp, M. One million posts: A data set of german online discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, pp. 1241–1244, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450350228. doi: 10.1145/3077136.3080711. URL https://doi.org/10.1145/3077136.3080711.

See, A., Liu, P. J., and Manning, C. D. Get to the point: Summarization with pointer-generator networks. In Barzilay, R. and Kan, M.-Y. (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://aclanthology.org/P17-1099/.

Shukla, S. M., Magoo, C., and Garg, P. Comparing fine tuned-lms for detecting llm-generated text. In *2024 3rd Edition of IEEE Delhi Section Flagship Conference (DELCON)*, pp. 1–8. IEEE, 2024.

Solaiman, I., Brundage, M., Clark, J., Askell, A., Herbert-Voss, A., Wu, J., Radford, A., Krueger, G., Kim, J. W., Kreps, S., McCain, M., Newhouse, A., Blazakis, J., McGuffie, K., and Wang, J. Release strategies and the social impacts of language models, 2019. URL https://arxiv.org/abs/1908.09203.

Su, J., Zhuo, T. Y., Wang, D., and Nakov, P. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text, 2023. URL https://arxiv.org/abs/2306.05540.

SuperAnnotate. SuperAnnotate: AI-Detector. https://huggingface.co/SuperAnnotate/ai-detector.

Tan, C., Niculae, V., Danescu-Niculescu-Mizil, C., and Lee, L. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16. International World Wide Web Conferences Steering Committee, April 2016. doi: 10.1145/2872427.2883081. URL http://dx.doi.org/10.1145/2872427.2883081.

Telenti, A., Auli, M., Hie, B. L., Maher, C., Saria, S., and Ioannidis, J. P. Large language models for science and medicine. *European journal of clinical investigation*, 54 (6):e14183, 2024.

Tolstykh, I., Tsybina, A., Yakubson, S., Gordeev, A., Dokholyan, V., and Kuprashevich, M. Gigacheck: Detecting llm-generated content. *arXiv preprint arXiv:2410.23728*, 2024.

Törnberg, P., Valeeva, D., Uitermark, J., and Bail, C. Simulating social media using large language models to evaluate alternative news feed algorithms. *arXiv preprint arXiv:2310.05984*, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023a. URL https://arxiv.org/abs/2302.13971.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn,

A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Völske, M., Potthast, M., Syed, S., and Stein, B. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4508. URL https://aclanthology.org/W17-4508.

Wang, H., Luo, X., Wang, W., and Yan, X. Bot or human? detecting chatgpt imposters with a single question, 2024a. URL https://arxiv.org/abs/2305.06424.

Wang, R., Chen, H., Zhou, R., Ma, H., Duan, Y., Kang, Y., Yang, S., Fan, B., and Tan, T. Llm-detector: Improving ai-generated chinese text detection with open-source llm instruction tuning. *arXiv preprint arXiv:2402.01158*, 2024b.

Wilcox, E. G., Pimentel, T., Meister, C., Cotterell, R., and Levy, R. P. Testing the predictions of surprisal theory in 11 languages, 2025. URL https://arxiv.org/abs/2307.03667.

Wolff, M. and Wolff, S. Attacking neural text detectors, 2022. URL https://arxiv.org/abs/2002.11768.

Wu, J., Yang, S., Zhan, R., Yuan, Y., Chao, L. S., and Wong, D. F. A survey on llm-generated text detection: Necessity, methods, and future directions. *Comput. Linguistics*, 51(1):275–338, 2025. doi: 10.1162/COLI_A_00549. URL https://doi.org/10.1162/coli_a_00549.

Yuan, A., Coenen, A., Reif, E., and Ippolito, D. Wordcraft: story writing with large language models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, pp. 841–852, 2022.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a machine really finish your sentence? In Korhonen, A., Traum, D., and Màrquez, L. (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., Tam, W. L., Ma, Z., Xue, Y., Zhai, J., Chen, W., Zhang, P., Dong, Y., and Tang, J. Glm-130b: An open bilingual pre-trained model, 2023. URL https://arxiv.org/abs/2210.02414.

Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., Mihaylov, T., Ott, M., Shleifer, S., Shuster, K., Simig, D., Koura, P. S., Sridhar, A., Wang, T., and Zettlemoyer, L. Opt: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Zhou, Z., Ning, X., Hong, K., Fu, T., Xu, J., Li, S., Lou, Y., Wang, L., Yuan, Z., Li, X., Yan, S., Dai, G., Zhang, X.-P., Dong, Y., and Wang, Y. A survey on efficient inference for large language models, 2024. URL https://arxiv.org/abs/2404.14294.

A. Related Work

In recent years, identifying AI-generated text has become an increasingly important challenge, leading to the development of various detection strategies. As outlined in Section 2, prior research (Ghosal et al., 2023; Sadasivan et al., 2025) emphasizes the inherent difficulty of this task, detection becomes more challenging as language models more closely emulate human writing. Nevertheless, Chakraborty et al. 2023 demonstrate that even highly capable models can still be statistically distinguishable under specific conditions, such as when using multiple samples or robust feature sets. This theoretical foundation supports many practical detection approaches that exploit subtle irregularities in LLM outputs.

While `DivEye` demonstrates strong performance across current models and attacks, we acknowledge that existing detectors may struggle against future generative models with more human-like distributions, as highlighted by recent work (Doughman et al., 2024). In particular, baseline failures can be attributed to the narrowing statistical gap between machine and human-generated text, a trend that will likely intensify. Nonetheless, by evaluating on paraphrasing and instruction-tuned variants, we partially simulate such future shifts and show that `DivEye` remains robust under these distributional changes.

Broadly, these approaches can be categorized into watermark-based methods and zero-resource detection.

Watermarking. Watermarking embeds traceable patterns in a model's outputs during training or generation, enabling downstream identification of machine-generated content (Ren et al., 2024; Liu et al., 2024a). While watermarking can be effective in controlled environments, it relies on access to or cooperation from the model's developers, an assumption that frequently fails in real-world or adversarial scenarios. Furthermore, it is inherently unsuitable for practical situations where AI-generated text lacks any embedded watermark. This limitation has led to growing interest in zero-resource detection methods, which make no assumptions about access to the model's internals or training data. Instead, these methods analyze the output text alone, offering a more flexible and broadly applicable approach. Within this space, techniques can be further categorized into fine-tuned methods, which rely on labeled datasets, and zero-shot methods, which generalize to unseen models without task-specific training.

Fine-tuned Detection. Fine-tuned detection methods represent a major strand of zero-resource detection, often leveraging fine-tuned classifiers built atop pre-trained language models (PLMs). A pivotal development was the Grover model, which demonstrated that models trained on text from specific generators can achieve high accuracy on in-distribution data, particularly when integrating Grover-specific layers. This inspired a wave of PLM-based detectors, most notably OpenAI's GPT-2 detector (Solaiman et al., 2019), which uses a RoBERTa classifier trained on GPT-2 outputs. However, such detectors often struggle to generalize across models, especially as newer LLMs introduce more fluent and coherent outputs.

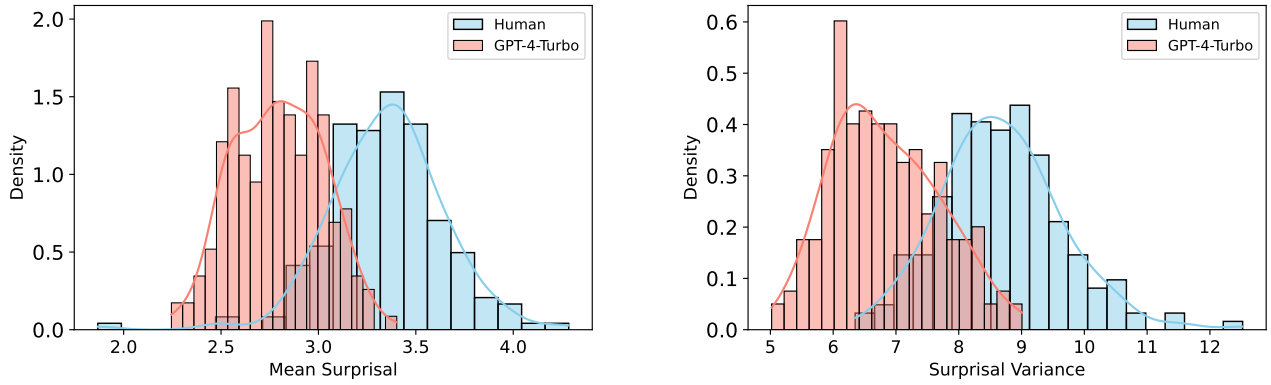
To improve generalization and robustness, recent work has focused on feature augmentation. Stylometric approaches, for instance, introduce handcrafted features that capture writing style discrepancies between humans and machines (Mikros et al., 2023). These include measures of phraseology, punctuation, linguistic diversity, and journalistic standards, which have proven useful for detecting AI-generated tweets and news articles. Additional features such as perplexity statistics, sentiment, and error-based cues like grammatical mistakes further enrich detection pipelines (Kumarage et al., 2023).

Parallel efforts have explored structural features, incorporating models that explicitly account for the factual or contextual structure of text. Techniques such as TriFuseNet combine stylistic and contextual branches with fine-tuned BERT models, while others employ attentive-BiLSTMs to replace standard feedforward layers, enhancing interpretability and robustness (Liu et al., 2024b).

Despite these advancements, fine-tuned detectors still require labeled training data and model-specific tuning of PLMs, which can limit their scalability to novel or proprietary LLMs. Although these detectors perform exceptionally well on data similar to their training sets, they face significant drawbacks, most notably, a tendency to overfit to specific domains and a reliance on retraining for every newly emerging AI model, which is unsustainable in light of the fast-paced evolution of generative technologies. This motivates the development of zero-shot methods, such as `DivEye`, that aim to detect AI-generated text without relying on supervised learning or access to model internals.

Zero-shot Detection. Recent research has focused on zero-shot detection strategies that require no fine-tuning on labeled examples from the target generator. These methods typically leverage statistical cues from PLM's output distributions or repurpose LLMs themselves as detectors.

A prominent class of zero-shot detectors exploits the probability structure of text under language models. RADAR (Hu



(a) Mean Surprisal Distribution

(b) Surprisal Variance Distribution

Figure 2: Distribution of token-level surprisal metrics for human-written vs. GPT-4-Turbo-generated essays. The left plot shows the histogram of mean surprisal per essay, while the right plot shows the histogram of surprisal variance. Human-written texts exhibit higher dispersion and heavier tails in both distributions, suggesting greater linguistic unpredictability and stylistic diversity. In contrast, GPT-4-Turbo outputs are more concentrated and predictable, aligning with the likelihood-maximization objective of language models.

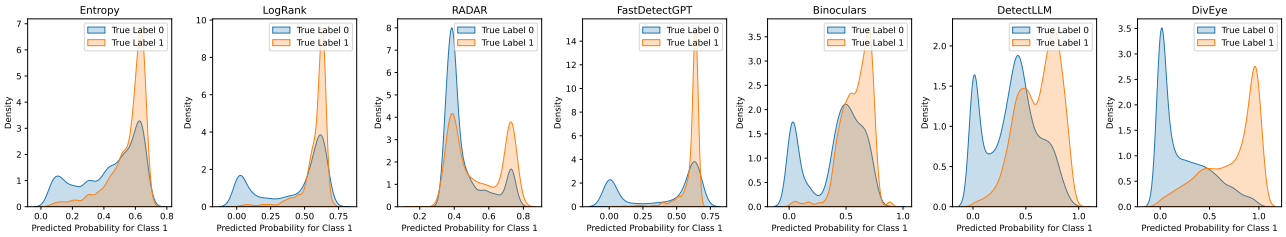


Figure 3: Distributions of predicted class probabilities for diverse AI-text detectors. Trained and evaluated on Testbed 4 of the MAGE benchmark, `DivEye` shows stronger separation between Class 0 (human-written) and Class 1 (AI-generated), indicating greater confidence and discriminative power.

et al., 2023) extends the curvature-based insight of DetectGPT by combining perturbation-based robustness testing with multiple surrogate models, thus improving generalization in LLM families. It applies controlled rewrites to the input and computes the variance in log probabilities across these perturbed samples under the hypothesis that AI-generated text resides in regions of higher curvature and lower local stability. FastDetectGPT (Bao et al., 2024) eliminates the need for explicit perturbations by directly measuring curvature in conditional probabilities, observing that AI text typically exhibits sharper transitions between tokens compared to human writing. These observations are refined in DetectLLM (Su et al., 2023), which introduces the Log-Likelihood Log-Rank Ratio (LRR) and Normalized Perturbed log-Rank (NPR) metrics to quantify the distinguishability of AI-generated content using statistical features derived from token rankings.

Another line of work focuses on token predictability and entropy. LogRank (Ghosal et al., 2023) investigates the use of token rank distributions and demonstrates that log-rank statistics, such as the frequency of top-ranked tokens, are reliable signals of AI authorship. This builds on early work such as entropy-based detection (Lavergne et al., 2008) and GLTR (Gehrmann et al., 2019), which showed that humans tend to use more surprising and diverse tokens, while LLMs often fall back on high-probability continuations.

Moving beyond single-directional statistics, BiScope (Guo et al., 2024) proposes a bi-directional cross-entropy framework that measures how well a model's predicted logits align both with the ground truth next token (forward loss) and with the previous token (backward loss). The key insight is that AI-generated text often exhibits predictable forward progression but weaker backward association due to its autoregressive nature. A shallow classifier trained on the joint distribution of these losses can reliably detect AI text with zero-shot generalization.

Finally, Binoculars (Hans et al., 2024) offers a model-agnostic strategy by comparing the statistical disagreement between two LLMs on the same input. By contrasting the outputs of two diverse LLMs, the method detects anomalies in token

distributions that are characteristic of synthetic text. This ensemble-based disagreement is found to correlate strongly with model-generated samples, providing a powerful signal without the need for training data from either model.

Collectively, these techniques demonstrate that zero-shot detection can be achieved by carefully analyzing how text aligns with the inductive biases and statistical signatures of language models, without any finetuning or access to the original generator. They lay the foundation for our proposed method, `DivEye`, which further capitalizes on diversity-based statistical properties to robustly differentiate AI- and human-written content.

Remark 1: Proof Sketch

Consider a text sequence $X = (x_1, x_2, \ldots, x_n)$ generated either by a human or by a language model M. The language model defines a probability distribution $P_M(X) = \prod_{t=1}^{n} P_M(x_t \mid x_{<t})$ where each token is chosen to maximize overall likelihood.
Humans, however, produce language through a complex, multi-layered cognitive process that balances informativeness, creativity, and contextual appropriateness, rather than strictly maximizing statistical likelihood.
Formally, the surprisal of token x_t under model M is defined as:

$$S_M(x_t) = -\log P_M(x_t \mid x_{<t})$$

Since M is trained to assign high probability to plausible continuations, its outputs tend to minimize surprisal on average, implying that maximum likelihood generation compresses diversity:

$$\mathbb{E}_{X\sim P_M}[S_M(x_t)] \leq \mathbb{E}_{X\sim P_H}[S_M(x_t)]$$

where P_H denotes the distribution of human-generated text.
Similarly, human language exhibits higher variance in surprisal due to spontaneous creative choices, idiomatic expressions, and stylistic variation, causing:

$$\mathrm{Var}_{X\sim P_M}[S_M(x_t)] < \mathrm{Var}_{X\sim P_H}[S_M(x_t)]$$

We validate this theoretical intuition through empirical experiments detailed below, which confirm statistically significant differences in surprisal and diversity metrics between human-written and AI-generated texts.
We collect 200 human-written essays and 200 GPT-4-Turbo-generated essays on comparable topics, provided by BiScope (Guo et al., 2024). For each essay, we computed the token-level surprisal scores using a fixed language model evaluator (GPT-2) and then calculated the mean and variance of these surprisal values per essay. Figure 2a shows the histogram of mean surprisal scores across the two sets, while Figure 2b displays the histogram of surprisal variances. The human-written texts exhibit a noticeably wider spread and heavier tails in both metrics, indicating greater unpredictability and stylistic variability. In contrast, the AI-generated essays cluster around lower mean surprisal and exhibit significantly lower variance. These results empirically confirm our theoretical claim: **human language inherently reflects higher diversity and surprise, whereas AI-generated language, optimized for likelihood, tends toward more predictable and homogeneous patterns.**

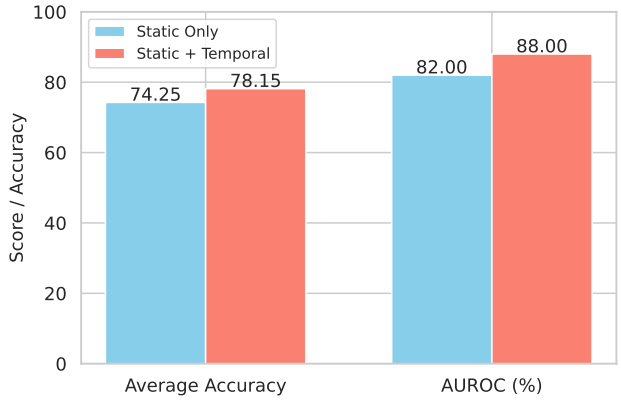


Figure 4: Ablation results on Testbed 4 of the MAGE benchmark showing the impact of temporal surprisal features. Adding temporal dynamics to static surprisal statistics improves both accuracy (from 74.25% to 78.15%) and AUROC (from 0.82 to 0.88), demonstrating their complementary value for robust AI-generated text detection.

Algorithm 1 `DivEye`: Algorithm for Feature Extraction & Training

Require: Text dataset $\mathcal{D} = \{(x_i, \ell_i)\}_{i=1}^{N}$, where x_i is a text input and $\ell_i \in \{0, 1\}$ indicates whether it is human-written ($\ell_i = 1$) or machine-generated ($\ell_i = 0$)
Require: Pretrained auto-regressive language model g_ϕ (e.g., GPT-2)
Require: XGBoost classifier with hyperparameters Θ
Ensure: Trained binary classifier f_θ
Initialize an empty feature matrix $\mathcal{F} \leftarrow [\,]$
for each $(x_i, \ell_i) \in \mathcal{D}$ **do**
    Compute token-level log-likelihoods: $y_i \leftarrow g_\phi(x_i)$
    Convert to token-level surprisals: $s_i \leftarrow -y_i$
    Compute diversity features $\texttt{DivEye}(x_i) \in \mathbb{R}^9$ as described in Equation (3) using s_i
    Append $(\texttt{DivEye}(x_i), \ell_i)$ to $\mathcal{F}$
end for
Train binary classifier f_θ on feature set $\mathcal{F}$ using XGBoost with hyperparameters Θ
return f_θ =0

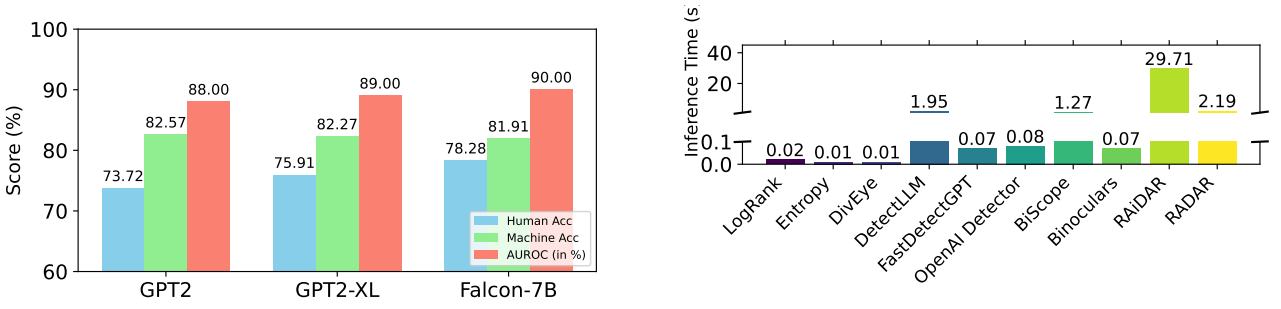
B. Implementation of `DivEye`

We provide a detailed description of our `DivEye` implementation in Algorithm 1. This includes all steps from surprisal computation to feature extraction and final classification. We use an XGBoost classifier for binary classification as a preliminary choice, without extensive comparison to other classifiers, leaving exploration of alternative models for future work. For completeness and reproducibility, we include all additional implementation details, such as hyperparameter configurations, model architectures, and experimental testbeds, in Appendix F and Appendix E.

C. Motivation Behind Temporal Features

While static surprisal statistics such as mean, variance, skewness, and kurtosis provide useful summaries of token-level unpredictability, they overlook the evolution of this unpredictability over time, a dimension critical to distinguishing human and AI-generated text. Human authors naturally embed stylistic variability through temporal fluctuations, such as abrupt topic shifts, tonal changes, and bursts of creativity, which manifest as distinctive temporal dynamics in surprisal sequences.

Intuitively, these temporal features, as listed in Section 3, expose rhythmic and non-stationary patterns characteristic of human creativity and coherence, typically absent in the more uniform output of large language models. These second-order



(a) DivEye using diverse base models.

(b) Inference time comparison of various frameworks.

Figure 5: (a) Performance of DivEye across different base models (GPT-2, GPT-2-XL, Falcon-7B). (b) Inference time (in sec) comparison of various methods.

features can be formally expressed as:

$$\Delta S_t = S(x_t) - S(x_{t-1}), \qquad \Delta\mu = \frac{1}{n-1}\sum_{t=2}^{n}\Delta S_t, \quad \Delta\sigma^2 = \frac{1}{n-1}\sum_{t=2}^{n}(\Delta S_t - \mu_\Delta)^2 \tag{4}$$

$$\Delta^2 S_t = \Delta S_t - \Delta S_{t-1}, \qquad \sigma^2_{\Delta^2} = \frac{1}{n-2}\sum_{t=3}^{n}(\Delta^2 S_t - \mu_{\Delta^2})^2, \quad \mathcal{H}_{\Delta^2} = -\sum_b p_b \log p_b, \tag{5}$$

$$\rho(\Delta^2 S_t) = \frac{\mathbb{E}\left[(\Delta^2 S_t - \mu_{\Delta^2})(\Delta^2 S_{t+1} - \mu_{\Delta^2})\right]}{\sigma^2_{\Delta^2}} \tag{6}$$

where μ_{Δ^2} is the mean of second-order differences, and p_b is the empirical probability of a value falling into bin b after discretizing $\Delta^2 S_t$ for entropy computation.

Furthermore, through an ablation study on Testbed 4 of the MAGE benchmark (Figure 4), we empirically show that augmenting static surprisal features with temporal metrics leads to a measurable improvement in classification accuracy. This highlights the complementary value of temporal dynamics in enhancing the robustness of AI-generated text detection. Moreover, an analysis of feature importance (Appendix D.6) reveals that temporal features collectively contribute more than static features, consistently ranking among the most informative signals for distinguishing between human and AI-generated text.

Overall, these findings motivate the inclusion of temporal surprisal features as integral components of our DivEye framework.

D. Additional Results

In this section, we present additional supporting experiments that demonstrate the generalizability, robustness, and complementary strengths of DivEye through various ablation studies.

D.1. Domain-Specific Performance of DivEye

Figure 7 presents the AUROC performance of seven detection methods evaluated across ten text domains (Testbed 3 of the MAGE benchmark). DivEye consistently achieves the highest AUROC scores in every domain—reaching up to 0.99 in WP, 0.97 in CMV, and 0.95 in SciXGen, outperforming other detectors by a notable margin. This highlights DivEye 's adaptability and robustness in capturing domain-specific writing patterns that other methods frequently miss. These results reinforce the advantage of leveraging surprisal features for more generalizable and context-sensitive detection of AI-generated text.

D.2. Model-Specific Performance of DivEye

Figure 8 compares the AUROC performance of seven detection methods across text on generated by six different large language models (Testbed 5 of the MAGE benchmark). DivEye achieves the highest AUROC scores across all six models, demonstrating strong robustness (0.95 on GLB-130B, 0.89 on GPT-J, 0.85 on GPT-3.5-Turbo). This consistent performance

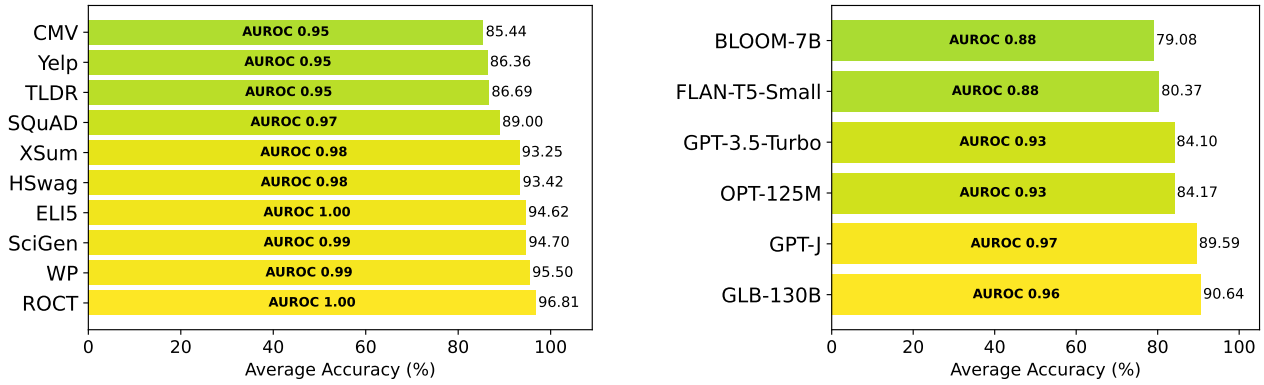


Figure 6: (a) Performance of DivEye across different domains, generated by GPT-J-6B. (b) Performance of DivEye across various generator models. Results are based on the MAGE benchmark.

Table 3: Performance of DivEye and baselines on adversarial benchmarks, MAGE & RAID. The RAID benchmark, which independently tests each model, does not report an AUROC score.

Settings	Methods	AvgAcc	AUROC
[MAGE] Testbed 8: Paraphrasing Attack			
Paraphrased via GPT-3.5-Turbo	Longformer (Beltagy et al., 2020)	69.34%	0.76
	BiScope (Guo et al., 2024)	69.30%	0.81
	DivEye **(Ours)**	**76.49%**	**0.87**
[RAID] Adversarial Attacks			
Paraphrase, Whitespace, Misspelling, Homoglyph, Article Deletion & more	Desklib AI (Desklib)	91.2%	-
	e5-small-lora (Dugan et al., 2024)	85.7%	-
	DivEye **(Ours)**	80.52%	-
	Binoculars (Hans et al., 2024)	69.32%	-
	RADAR (Hu et al., 2023)	63.9%	-
	GLTR (Gehrmann et al., 2019)	51.5%	-

highlights DivEye's effectiveness in capturing temporal surprisal patterns that generalize well across different language model architectures, making it broadly applicable for reliable AI-generated text detection.

D.3. Adversarial Attack Analysis of DivEye

We evaluate DivEye against a wide range of adversarial attacks using the RAID benchmark, reporting average classification accuracies across all attack categories listed in Table 4. DivEye achieves performance on par with the top-performing fine-tuned models reported by the benchmark. While one might argue that LLMs can be manipulated to produce more diverse text, potentially evading detection, our evaluation includes paraphrasing attacks, which are specifically designed to do just that. Notably, it consistently surpasses all zero-shot detectors by a significant margin across every attack type, demonstrating strong robustness against both diverse adversarial attacks.

D.4. Relative Importance of DivEye in a Boosted Model

Figure 9 illustrates the relative feature importance of DivEye when integrated into boosted ensembles with five existing AI detectors: BiScope (Guo et al., 2024), OpenAI Detector (Solaiman et al., 2019), RADAR (Hu et al., 2023), DetectLLM (Su et al., 2023), and Binoculars (Hans et al., 2024). DivEye contributes significantly to the overall model, with particularly high importance when combined with RADAR (91.92%), OpenAI Detector (90.26%), and Binoculars (89.71%). Even in ensembles with more advanced detectors like BiScope, DivEye still adds valuable signal (32.93%). These results affirm the standalone strength of DivEye and its utility in hybrid detection frameworks.

D.5. Results with Different Propriety LLMs

Table 6 reports AUROC scores of DivEye on text generated by five proprietary LLMs, Claude-3 Opus, Claude-3 Sonnet, Gemini 1.0-pro, GPT-3.5 Turbo, and GPT-4 Turbo, using data provided in the BiScope paper (Guo et al., 2024) across five domains. DivEye achieves consistently strong performance on the Normal dataset (e.g., 1.000 on GPT-3.5 Turbo for

Figure 7: AUROC performance profiles of seven AI-detection tools evaluated on text generated by ten diverse domains generated by arbitrary LLMs. Each spider plot corresponds to a specific domain, with radial axes representing the AUROC score (ranging from 0 to 1) and angular axes representing the detection tools: RADAR, Entropy, LogRank, FastDetectGPT, DetectLLM, OpenAI Detector, and `DivEye`.

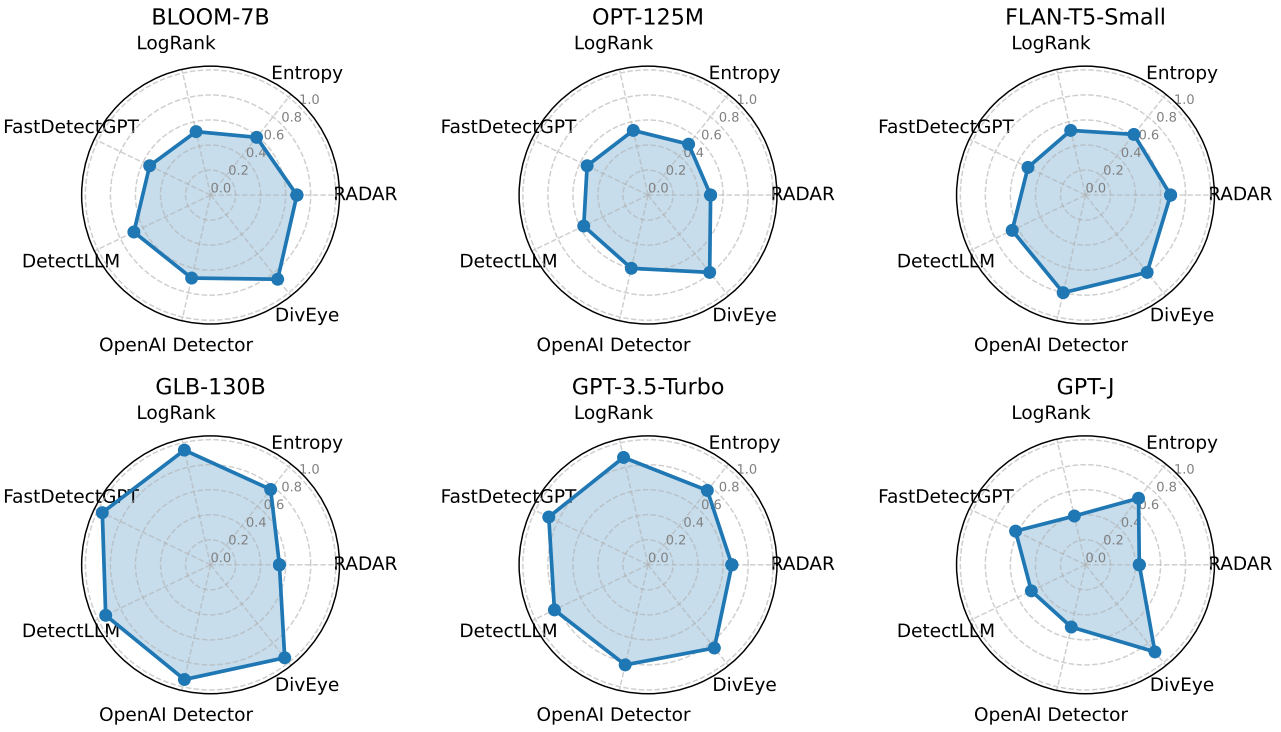


Figure 8: AUROC performance profiles of seven AI-detection tools evaluated on text generated by six different LLMs. Each spider plot corresponds to a specific language model, with radial axes representing the AUROC score (ranging from 0 to 1) and angular axes representing the detection tools: RADAR, Entropy, LogRank, FastDetectGPT, DetectLLM, OpenAI Detector, and `DivEye`.

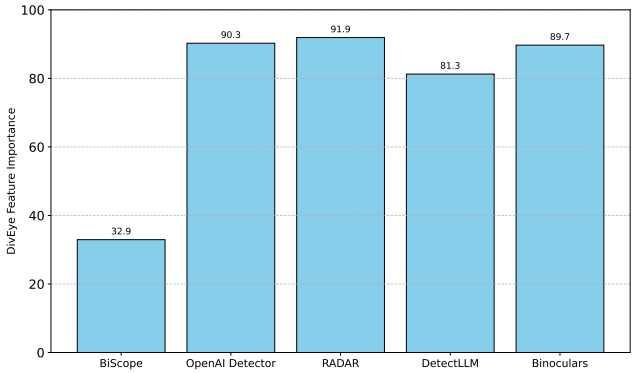


Figure 9: Feature importance of `DivEye` when integrated with various existing detectors. The plot shows how much `DivEye` contributes to the overall detection model when combined with BiScope, OpenAI Detector, RADAR, DetectLLM, and Binoculars. Higher values indicate stronger complementary impact from `DivEye` 's diversity-based features.

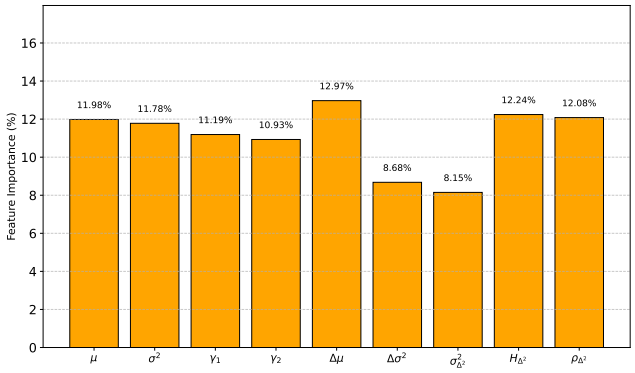


Figure 10: Relative feature importances for the nine diversity-based features used in DivEye. The features, as listed in Equation (3), represent distinct surprisal-based statistics. Higher percentages indicate greater influence in model decisions when combined with existing detectors.

Essay) and remains robust under paraphrased inputs, with AUROC scores generally above 0.95. These results highlight DivEye's ability to generalize across diverse generation models and domains, even under text transformations.

D.6. Feature Importance of DivEye

Figure 10 presents the relative importance of each of the nine diversity-based features incorporated in DivEye, which are derived from surprisal statistics as detailed in Equation (3). The feature importances, ranging from approximately 8.1% to 13.0%, indicate that all features contribute meaningfully to model decisions, with temporal features such as, $\Delta\mu$, entropy of second derivatives H_{Δ^2}, and autocorrelation ρ_{Δ^2} exhibiting the highest influence. This balanced contribution underscores the complementary nature of these statistical descriptors in enhancing DivEye 's detection capability when combined with existing baseline detectors.

E. Testbed Details

We evaluate DivEye on a comprehensive testbed spanning three major AI-text detection benchmarks, MAGE (Li et al., 2024), HC3 (Guo et al., 2023) & RAID (Dugan et al., 2024), covering a diverse range of domains, language models, and adversarial attacks. These benchmarks allow us to assess the generalizability and robustness of our method across realistic deployment scenarios. This section provides a comprehensive overview of the testbeds used in our evaluation, including all domains, language models, and adversarial attacks featured in the MAGE and RAID benchmarks, along with relevant configuration details.

Details about MAGE Benchmark. The MAGE benchmark (Li et al., 2024) comprises eight diverse testbeds designed for evaluating machine-generated text detection. Testbeds 1 through 4 include standard train, validation, and test splits, while Testbeds 5 through 8 serve as out-of-distribution (OOD) datasets, evaluated using models trained on Testbed 4. Notably, Testbed 4, Arbitrary Domains & Arbitrary Models, is the most comprehensive, enabling evaluation across the full range of domains and language models listed in the MAGE paper. Detailed information regarding dataset splits and sample counts is available in the original benchmark documentation.

MAGE covers a wide array of domains, including CMV (Tan et al., 2016), Yelp (Zhang et al., 2015), XSum (Narayan et al., 2018), TLDR, ELI5 (Fan et al., 2019), WP (Fan et al., 2018), ROC (Mostafazadeh et al., 2016), HellaSwag (Zellers et al., 2019), SQuAD (Rajpurkar et al., 2016), and SciXGen (Chen et al., 2021a). The OOD domains include CNN/DailyMail (See et al., 2017), DialogSum (Chen et al., 2021b), PubMedQA (Jin et al., 2019), and IMDb (Maas et al., 2011).

MAGE also incorporates text generated from over 27 different LLMs (Brown et al., 2020; Chung et al., 2022; et al., 2023; Sanh et al., 2022; Touvron et al., 2023a; Zeng et al., 2023; Zhang et al., 2022), enabling rigorous and varied evaluations. For further implementation specifics, readers are encouraged to consult the MAGE paper.

Details about RAID Benchmark. The RAID benchmark comprises over 6.2 million samples, offering extensive coverage across domains, language models, sample sizes, and adversarial attacks. It provides a clear separation into training, validation, and testing splits to support rigorous evaluation. The benchmark spans a wide range of domains, including scientific abstracts (Paul & Rakshit, 2021), book summaries (Bamman & Smith, 2013), BBC News articles (Greene & Cunningham, 2006), poems (Arman, 2020), recipes (Bień et al., 2020), Reddit posts (Völske et al., 2017), movie reviews (Maas et al., 2011), Wikipedia entries (Aaditya Bhat, 2023), Python code, Czech news (Boháček et al., 2022), and German news articles (Schabus et al., 2017).

RAID employs text generated from 11 diverse LLMs (Radford et al., 2019; MosaicML, 2023; Jiang et al., 2023; Cohere, 2024; Ouyang et al., 2022; Touvron et al., 2023b; et al., 2024), ensuring broad model representation. Additionally, it includes over 11 adversarial attack strategies (Liang et al., 2023b;a; Wolff & Wolff, 2022; Bhat & Parthasarathy, 2020; Krishna et al., 2023; Pu et al., 2023; Gagiano et al., 2021; Guerrero et al., 2022), designed to test the robustness of detectors under challenging settings. Detailed results and descriptions of these attacks are provided in Appendix D.3. For further implementation specifics, readers are encouraged to consult the RAID paper.

F. Hyperparameter Settings

Table 7 outlines the hyperparameter configurations used for our experiments. We utilize the XGBoost classifier with standard but tuned settings to handle class imbalance and optimize detection performance. For our proposed method `DivEye`, we set the number of bins for entropy computation to 20 and truncate input sequences at a maximum length of 1024 tokens. All experiments were run on a single NVIDIA DGX A100 (40 GB), and reported results reflect the median of three runs.

G. Future Work, Limitations, Reproducibility & Ethical Considerations

Future Work. While `DivEye` demonstrates strong generalization across domains and models in zero-shot settings, several extensions remain open. A key extension is evaluating `DivEye`'s robustness against paraphrasing attacks using online rewriters and adversarial perturbation tools, which better simulate real-world evasion tactics. Another important direction is assessing cross-lingual generalizability by applying `DivEye` to texts written in or translated to other languages, as our current evaluation focuses only on English.

Limitations. Our approach relies on features derived from LLM token-level behavior, which may vary across model sizes, architectures, and tokenization schemes. While our current performance is robust, it is unclear whether we are approaching an optimal limit for AI-text detection, leaving room for further improvements across diverse models. Finally, we focus primarily on English-language text, and the feature distributions may behave differently in other linguistic or cultural settings. Furthermore, `DivEye` is optimized for English text; performance on multilingual content remains unexplored.

Reproducibility. We release all code and evaluation scripts to ensure full reproducibility. Detailed training, testing and hyperparameter configurations are included in Appendices E and B.

Ethical Considerations. As with all AI-text detectors, `DivEye` is not infallible and may produce incorrect classifications. We emphasize that detection outputs should be treated as probabilistic signals rather than definitive evidence. When used in high-stakes settings, such as academic integrity or content moderation, additional human review and validation are essential. We encourage responsible deployment of `DivEye` to support large-scale analysis, but caution against its use in critical decision-making.

Table 4: Performance of DivEye and open-source baselines on all listed adversarial attacks on the RAID benchmark.

Settings	Methods	AvgAcc
[RAID] Adversarial Attacks		
Whitespace Attack	Desklib AI	94.9%
	e5-small-lora	93.9%
	DivEye (Ours)	79.8%
	Binoculars	68.7%
	RADAR	61.1%
	GLTR	43.1%
Upper-Lower Attack	Desklib AI	87.2%
	e5-small-lora	93.9%
	DivEye (Ours)	85.3%
	Binoculars	72.8%
	RADAR	65.1%
	GLTR	45.3%
Synonym Attack	Desklib AI	80.6%
	e5-small-lora	85.6%
	DivEye (Ours)	67.1%
	Binoculars	42.1%
	RADAR	62.7%
	GLTR	28.7%
Paraphrase Attack	Desklib AI	83.7%
	e5-small-lora	85.5%
	DivEye (Ours)	74.4%
	Binoculars	N/A
	RADAR	62.4%
	GLTR	43.0%
Perplexity Misspelling	Desklib AI	92.9%
	e5-small-lora	92.5%
	DivEye (Ours)	90.6%
	Binoculars	77.2%
	RADAR	64.3%
	GLTR	57.0%
Number Attack	Desklib AI	93.0%
	e5-small-lora	93.5%
	DivEye (Ours)	92.1%
	Binoculars	76.4%
	RADAR	65.7%
	GLTR	57.3%
Insert Paragraph	Desklib AI	94.9%
	e5-small-lora	93.9%
	DivEye (Ours)	92.2%
	Binoculars	70.7%
	RADAR	68.2%
	GLTR	58.3%
Homoglyph Attack	Desklib AI	99.7%
	e5-small-lora	11.1%
	DivEye (Ours)	61.6%
	Binoculars	36.1%
	RADAR	44.8%
	GLTR	20.3%
Article Deletion	Desklib AI	90.5%
	e5-small-lora	92.0%
	DivEye (Ours)	88.0%
	Binoculars	73.3%
	RADAR	63.0%
	GLTR	48.9%
Alt. Spelling Attack	Desklib AI	94.3%
	e5-small-lora	93.4%
	DivEye (Ours)	92.01%
	Binoculars	77.6%
	RADAR	65.5%
	GLTR	58.2%
Zero Width Space	Desklib AI	87.5%
	e5-small-lora	93.9%
	DivEye (Ours)	92.0%
	Binoculars	98.4%
	RADAR	78.4%
	GLTR	97.9%

Table 5: Integration with DivEye consistently boosts performance across detectors, particularly on diverse domains (Testbed 4) and paraphrasing attacks (Testbed 7).

Methods	HumanAcc	MachineAcc	AvgAcc	AUROC	δ: Boost
Testbed 4: Arbitrary Domains & Arbitrary Models					
RADAR	47.74%	74.86%	61.30%	0.62	-
DetectLLM	64.74%	69.02%	66.88%	0.75	-
FastDetectGPT	93.65%	41.73%	67.69%	0.7	-
Binoculars	76.1%	54.89%	65.49%	0.71	-
BiScope	91.54%	58.70%	75.12%	0.86	
DivEye	73.72%	82.57%	78.15%	0.88	-
DivEye + RADAR	74.69%	85.31%	80%	0.90	**18.7%**
DivEye + DetectLLM	75.44%	84.23%	79.34%	0.9	12.96%
DivEye + FastDetectGPT	79.42%	83.90%	81.66%	0.91	13.97%
DivEye + Binoculars	69.81%	83.47%	76.64%	0.87	11.15%
DivEye + BiScope	80.69%	88.31%	**84.5%**	**0.93**	9.38%
Testbed 7: Paraphrasing Attacks					
BiScope	48.80%	89.79%	69.30%	0.81	-
DivEye	69.75%	83.22%	76.49%	0.87	-
DivEye + BiScope	65.38%	90.84%	**78.11%**	**0.89**	**8.81%**

Table 6: AUROC scores achieved by DivEye on five commercial LLMs across various domains. Results are shown for both the Normal and Paraphrased datasets.

Domain	Normal Dataset					Paraphrased Dataset				
	Claude-3 Opus	Claude-3 Sonnet	Gemini 1.0-pro	GPT-3.5 Turbo	GPT-4 Turbo	Claude-3 Opus	Claude-3 Sonnet	Gemini 1.0-pro	GPT-3.5 Turbo	GPT-4 Turbo
Arxiv	0.9942	0.9770	0.9795	0.9658	0.9793	0.9778	0.9552	0.9616	0.9689	0.9558
Code	0.7528	0.8557	0.7824	0.9577	0.9044	0.8456	0.9053	0.7521	0.9279	0.9302
Creative	0.9888	0.9773	0.9835	0.9951	0.9608	0.9930	0.9900	0.9957	0.9917	0.9949
Essay	0.9950	0.9988	0.9972	1.0000	0.9823	0.9975	0.9877	0.9814	0.9895	0.9559
Yelp	0.8855	0.8813	0.9220	0.8384	0.8942	0.9543	0.9780	0.9683	0.8524	0.9571

Table 7: Hyperparameters used for the XGBoost Classifier and DivEye.

XGBoost Hyperparameter	Value
random_state	42
scale_pos_weight	$(\text{len}(Y_{\text{train}}) - \sum Y_{\text{train}}) / \sum Y_{\text{train}}$
max_depth	12
n_estimators	200
colsample_bytree	0.8
subsample	0.7
min_child_weight	5
gamma	1.0
DivEye Parameter	**Value**
Entropy bins	20
Tokenizer Max Length	1024 + Truncation