DivEye at PAN 2025: Diversity Boosts AI-Generated Text Detection

Notebook for PAN at CLEF 2025

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

[1]Birla Institute of Technology and Science, KK Birla Goa Campus, India
[2]IBM Research, USA

Abstract

Detecting AI-generated text is increasingly important to prevent misuse in education, journalism, and social media, where synthetic fluency can obscure misinformation. This paper presents our solution for the Generative AI Authorship Verification Task at PAN 2025, where the objective is to distinguish machine-generated text from human-written content. 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. Our method performs competitively across diverse text domains and models, especially on challenging cases where model-generated text closely resembles human writing, and also outperforms the four official baselines of the PAN 2025 task.

Keywords

PAN 2025, Voight-Kampff AI Detection Sensitivity, Generative AI Authorship Verification Task, llms, ai text detection, interpretability, zero-shot

1. Introduction

Large Language Models (LLMs) are widely used in tasks from personal assistance to content creation [1, 2, 3, 4, 5]. 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 [6, 7, 8, 9].

Reliable AI-text detection is vital for combating risks like misinformation, academic dishonesty, professional misconduct, and the suppression of genuine human writing [10, 11, 12]. Traditional supervised detectors [13, 14, 15] rely on labeled datasets but often fail to generalize to unseen models or domains [16, 11], especially as new LLMs emerge. Zero-shot detectors [17, 18, 19, 20] address this by leveraging statistical signals or LLMs at inference time, offering scalable, model-agnostic detection critical for maintaining platform integrity.

Contributions. In this work, we present DivEye, a zero-shot framework for AI-generated text detection submitted to the PAN@CLEF 2025 Generative AI Authorship Verification task [**? ?**]. The challenge centers on identifying the human-written text when presented with a pair comprising one human and one machine-authored sample. Our method leverages diversity-based statistical features computed over token-level surprisal [21] sequences from a reference language model. Unlike fine-tuned classifiers or signature-based detectors, DivEye captures distributional irregularities inherent in AI-generated content by measuring surprisal variance, entropy, and other diversity features. These metrics are grounded in linguistic theory and require no access to a specific text-generation LM. DivEye is model-agnostic, scalable, and can operate without retraining, making it suitable for real-world deployment. Notably, it complements existing detectors by revealing statistical signals often missed by black-box or fine-tuned approaches. Our results show strong generalization across domains and model families, achieving competitive performance in this challenging verification setting.

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain
*Corresponding author.
f20221155@goa.bits-pilani.ac.in (A. R. Basani); pin-yu.chen@ibm.com (P. Chen)
https://floofcat.github.io/ (A. R. Basani); https://sites.google.com/site/pinyuchenpage/home (P. Chen)
0009-0003-3389-9147 (A. R. Basani); 0000-0003-1039-8369 (P. Chen)

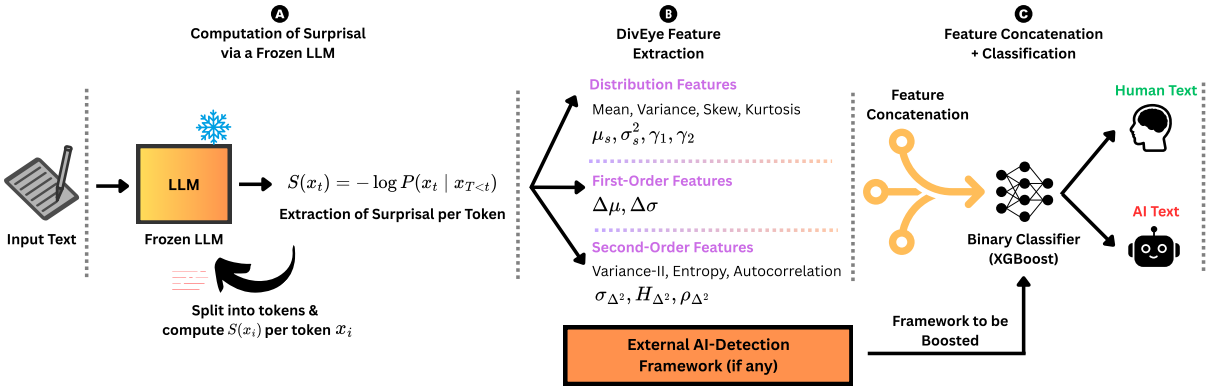


Figure 1: Overview of DivEye. DivEye extracts diversity-based features (see Section 3, Equation (6)) from token-level surprisal patterns. These features can be used in two ways: (1) as a standalone detector, or (2) as an enhancement to existing detectors, improving their performance.

2. Background & Preliminaries

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 [22, 23]. The LLM's learned distribution, $P_{\text{LLM}}(x_t \mid x_{<t})$, is used to sequentially generate tokens during inference via sampling [24]. Despite their fluency, LLMs imperfectly approximate human language ($P_{\text{LLM}} \neq P_{\text{human}}$) [25, 26], and this subtle difference is the crux of AI text detection.

Existing detection methods fall into three categories: watermarking, supervised / fine-tuned and zero-resource detection. Watermarking [27, 28, 29] 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.

Supervised / Fine-tuned detection methods [30, 31, 32] 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. For example, methods like Entropy [33], LogRank [34], DetectGPT [17, 19], and Binoculars [35] use off-the-shelf LLMs to evaluate the consistency of token predictions under masked or perturbed inputs.

Despite progress, AI-text detection remains unsolved. We move beyond individual token probabilities to measure statistical diversity across token sequences, capturing variation in surprise and predictability.

3. DivEye: Methodologies

3.1. Design Hypothesis

One of the main challenges in detecting AI-generated text [34, 36] lies in the fact that while modern LLMs excel at generating fluent and coherent text, they often do so at the expense of variability and diversity [37].

We hypothesize that human-authored text naturally displays greater stylistic diversity and unpredictability than text produced by AI. Human writing tends to include creative and impulsive choices that introduce unexpected shifts, whereas large language models prioritize high-probability sequences [38], resulting in more uniform and predictable outputs. This hypothesis is supported by both intuitive reasoning and empirical findings (see Remark 1).

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.

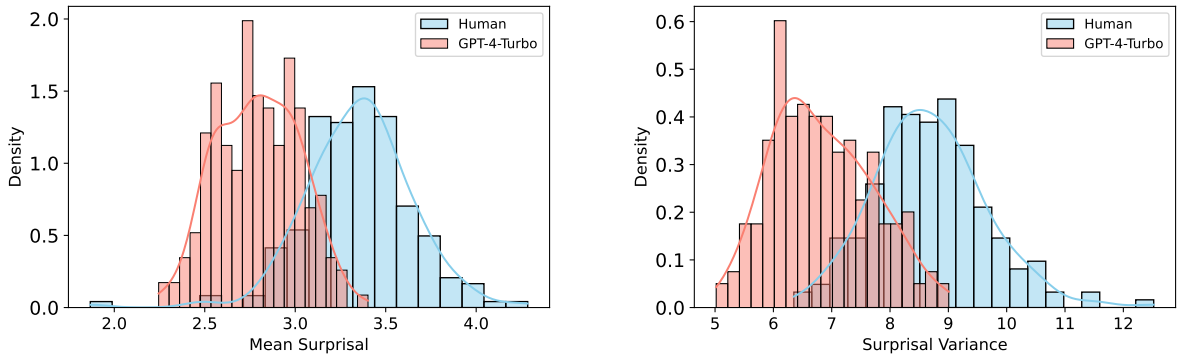
We empirically validate these theoretical claims using a dataset of 200 human-written essays and 200 GPT-4-Turbo-generated essays on matched topics, sourced from BiScope [39]. For each essay, we compute token-level surprisal scores using a fixed language model evaluator (GPT-2), then calculate the mean and variance of surprisal per essay. Figure 2a presents the distribution of mean surprisal scores across both groups, while Figure 2b shows the corresponding variance distributions. Human-written texts demonstrate a broader spread and heavier tails in both metrics, indicating greater unpredictability and stylistic richness. In contrast, AI-generated texts are more tightly clustered with lower mean surprisal and significantly reduced variance. These findings empirically corroborate our hypothesis: **human language inherently reflects higher diversity and surprise, whereas AI-generated language, optimized for likelihood, tends toward more predictable and homogeneous patterns.**

3.2. Foundations of DivEye

DivEye computes higher-order statistical features over surprisal sequences, enabling the capture of structural patterns that go beyond aggregate likelihood. More detailed theoretical foundations and experimental results are presented in the original DivEye paper [40].

Surprisal. Human language balances consistency with creative bursts, introducing novel expressions and stylistic variation. This diversity can be quantified using surprisal [41], 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)



(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.

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 [25]. 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 as:

$$\Delta S_t = S(x_t) - S(x_{t-1}), \qquad \Delta\mu = \frac{1}{n-1}\sum_{t=2}^{n}\Delta S_t, \quad \Delta\sigma^2 = \frac{1}{n-1}\sum_{t=2}^{n}(\Delta S_t - \mu_\Delta)^2 \tag{3}$$

$$\Delta^2 S_t = \Delta S_t - \Delta S_{t-1}, \qquad \sigma^2_{\Delta^2} = \frac{1}{n-2}\sum_{t=3}^{n}(\Delta^2 S_t - \mu_{\Delta^2})^2, \quad \mathcal{H}_{\Delta^2} = -\sum_b p_b \log p_b, \tag{4}$$

$$\rho(\Delta^2 S_t) = \frac{\mathbb{E}\left[(\Delta^2 S_t - \mu_{\Delta^2})(\Delta^2 S_{t+1} - \mu_{\Delta^2})\right]}{\sigma^2_{\Delta^2}} \tag{5}$$

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

Combinations. Collectively, DivEye, formalized as ($\mathcal{D}$) in Equation (6), encapsulates critical aspects of text generation that distinguish human creativity from algorithmically generated predictability, thereby serving as a robust basis for our detection framework.

$$\mathcal{D} = \{\underbrace{\mu_s, \sigma_s^2, \gamma_1, \gamma_2}_{\text{Distribution}} \oplus \underbrace{\Delta\mu, \Delta\sigma^2}_{\text{1st-Order}} \oplus \underbrace{\sigma^2_{\Delta^2}, H_{\Delta^2}, \rho_{\Delta^2}}_{\text{2nd-Order}}\} \tag{6}$$

$\mathcal{D}$ 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 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 [42], Random Forest [43]) 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

Datasets. We evaluate our zero-shot DivEye framework on a diverse suite of datasets that span a wide range of generative models, domains, and adversarial strategies. To substantiate the methodology described above[1], our primary benchmark is the MAGE benchmark [44]. MAGE comprises eight distinct testbeds covering multiple domains (e.g., Yelp [45], XSum [46], SciXGen [47], CMV [48]) and a range of text generator families (e.g., GPT [49], OPT [50], Bloom [51]).

This fine-grained evaluation setup enables us to isolate and analyze the contribution of diversity-based metrics across different domains and model architectures. Each testbed includes predefined training and evaluation splits, which we use accordingly. For full implementation details and extended experimental results on MAGE, we request the reader to refer to our original paper [40], where all MAGE-related experiments are presented and discussed in depth.

PAN Dataset. The PAN@CLEF 2025 Generative AI Author Verification Task [52, 53] provides a dataset comprising both human-authored and machine-generated texts. We utilize the training and supplementary evaluation splits from this dataset to train a binary classifier enhanced with features from DivEye. All of our submissions are exclusively trained on this provided dataset.

Implementation Details & Metrics. Unless stated otherwise, we use GPT-2 to compute all DivEye feature vectors. In score-only detection scenarios, predictions are based solely over concatenated DivEye features. For both standalone and boosted setups, we train a lightweight XGBoost [42] classifier as a meta-model, using only DivEye features in the former, and concatenating them with the original detector's prediction scores in the latter. 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. We evaluate our method using the official PAN@CLEF 2025 evaluation platform (TIRA [54]), which reports the following metrics:

- **AUROC:** The conventional Area Under the Receiver Operating Characteristic Curve.
- **c@1:** A metric that rewards systems for leaving uncertain cases unanswered.
- $F_{0.5u}$**:** A variation of the F-score that emphasizes correctly identifying same-author cases.
- **F1-score:** The harmonic mean of precision and recall, capturing balanced model performance.

[1]Note: These models were not submitted for any tasks in PAN 2025; results are reported solely to empirically validate our approach.

- **Brier Score:** Measures the accuracy of probabilistic predictions by computing the mean squared error between predicted probabilities and true labels.

Baselines. We compare DivEye against a diverse set of baselines under two evaluation settings. As detailed in our original paper [40], for the MAGE benchmark, we evaluate both traditional statistical detectors and recent fine-tuned models, including RADAR [32], FastDetectGPT [17], Binoculars [35], and BiScope [39].

Although we do not explicitly report quantitative results in this manuscript for MAGE, we kindly refer readers to our original paper [40] for full empirical comparisons. Nevertheless, we summarize and discuss the key findings and core observations here to provide insight into the comparative performance of DivEye.

For the PAN@CLEF 2025 task, we follow the official evaluation protocol and compare against the provided baselines: Linear SVM with TF-IDF features, Binoculars [35], and a PPMd compression-based cosine similarity method [55]. These lightweight, model-agnostic baselines highlight the advantage of incorporating statistical diversity features even in constrained, zero-shot scenarios.

Table 1
Performance of DivEye and baselines on pan25-generative-ai-detection-val

Methods	AUROC	Brier	C@1	F1	$F_{0.5u}$
tart-league (DivEye [GPT-2] + BiScope)	**0.997**	**0.983**	0.978	**0.983**	**0.983**
tangy-gorgonzola (DivEye [Falcon-7B] + BiScope)	**0.997**	0.919	0.912	0.897	0.956
weary-jersey (DivEye [GPT-2])	0.961	0.929	0.905	0.926	0.924
baseline-tf-idf	0.996	0.951	**0.984**	0.98	0.981
baseline-binoculars-llama-3.1	0.918	0.867	0.843	0.873	0.882
baseline-binoculars-tiny-llama	0.821	0.751	0.627	0.585	0.773
baseline-ppmd	0.786	0.799	0.757	0.812	0.778

4.1. DivEye in PAN 2025

To evaluate the effectiveness of DivEye in the PAN 2025 authorship verification task, we follow the official protocol and train exclusively on the dataset provided by the organizers. Given the consistent performance gains observed with DivEye across diverse settings, we focus our submission on the enhanced variant DivEye + BiScope [39], which demonstrated superior results in earlier evaluations. This combination leverages the complementary strengths of BiScope's decision boundary with DivEye 's diversity-based signal, leading to improved robustness and generalization across authorship verification cases. The final results are summarized in Table 1.

4.2. Robustness, Efficiency & Boosting Effectiveness of DivEye

We evaluate DivEye across a wide range of challenging testbeds to assess its robustness and adaptability under both domain and model distribution shifts. Our experiments span multiple testbeds from the MAGE benchmark [44], including both in-distribution and out-of-distribution scenarios. Across all settings, DivEye consistently outperforms existing zero-shot and fine-tuned baselines in terms of AUROC and average accuracy, demonstrating strong generalization to both familiar and novel generation patterns. In our original paper [40], we report detailed performance metrics for DivEye across these testbeds. The results highlight high AUROCs (e.g., 0.98 and 0.93 across domains and generator families), along with strong accuracy, underscoring the stability and robustness of our approach across diverse evaluation conditions.

To further assess robustness, we evaluate DivEye under adversarial conditions such as paraphrasing attacks. Even in these challenging scenarios, DivEye outperforms strong fine-tuned baselines by notable margins in both AUROC and average accuracy. In addition to accuracy, DivEye is highly efficient, processing each input in approximately 0.01 seconds due to its lightweight GPT-2 backbone and fast statistical feature computations. This makes it particularly well-suited for deployment in

real-time or resource-constrained environments.

Finally, we demonstrate that DivEye's diversity-based surprisal features substantially enhance the performance of existing detectors when used in combination. Fusing DivEye with other diverse detectors results in AUROC and accuracy gains exceeding 18.7%, showing that these features offer complementary signals to traditional methods. For full experimental results, including quantitative comparisons and boosting analyses, we refer reviewers to our original paper [40].

5. Conclusion

We successfully participated in the PAN@CLEF2025 Generative AI Authorship Verification task using our proposed framework, DivEye, which leverages surprisal diversity for robust zero-shot detection. By integrating DivEye with BiScope, we achieved strong performance in distinguishing human-written from machine-generated texts, overperforming all given baselines. Our method shows high adaptability across domains and paraphrased inputs, indicating its effectiveness in real-world authorship verification scenarios.

Acknowledgments

We would like to thank the Data, Systems and High Performance Computing (DaSH) Lab[2] and the PI, Prof. Arnab K. Paul, for providing the computational resources necessary to conduct our experiments.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

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

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

[3] B. D. Lund, T. Wang, N. R. Mannuru, B. Nie, S. Shimray, Z. Wang, 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 (2023) 570–581.

[4] J. Hu, H. Gao, Q. Yuan, G. Shi, Dynamic content generation in large language models with real-time constraints (2024).

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

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

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

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

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

[2] https://www.dashlab.in/

[10] S. Abdali, R. Anarfi, C. Barberan, J. He, Decoding the ai pen: Techniques and challenges in detecting ai-generated text, 2024. URL: https://arxiv.org/abs/2403.05750. doi:https://doi.org/10.1145/3637528.3671463. arXiv:2403.05750.

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

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

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

[14] I. Tolstykh, A. Tsybina, S. Yakubson, A. Gordeev, V. Dokholyan, M. Kuprashevich, Gigacheck: Detecting llm-generated content, arXiv preprint arXiv:2410.23728 (2024).

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

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

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

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

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

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

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

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

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

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

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

[26] C. R. Jones, S. Trott, B. Bergen, 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 (2024) 803–819. URL: https://doi.org/10.1162/tacl_a_00674. doi:10.1162/tacl_a_00674.

[27] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, T. Goldstein, A watermark for large language

models, 2024. URL: https://arxiv.org/abs/2301.10226. arXiv:2301.10226.

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

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

[30] Y. Chen, H. Kang, V. Zhai, L. Li, R. Singh, B. Raj, Token prediction as implicit classification to identify LLM-generated text, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 13112–13120. URL: https://aclanthology.org/2023.emnlp-main.810/. doi:10.18653/v1/2023.emnlp-main.810.

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

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

[33] T. Lavergne, T. Urvoy, F. Yvon, 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, CEUR-WS.org, Aachen, DEU, 2008, p. 27–31.

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

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

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

[37] C. Yang, A. Holtzman, How alignment shrinks the generative horizon, 2025. URL: https://arxiv.org/abs/2506.17871. arXiv:2506.17871.

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

[39] H. Guo, S. Cheng, X. Jin, Z. ZHANG, K. Zhang, G. Tao, G. Shen, X. Zhang, 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.

[40] A. R. Basani, P.-Y. Chen, Diversity boosts AI-generated text detection, in: Data in Generative Models - The Bad, the Ugly, and the Greats, 2025. URL: https://openreview.net/forum?id=QuDDXJ47nq.

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

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

[43] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32. URL: https://doi.org/10.1023/A:1010933404324. doi:10.1023/A:1010933404324.

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

[45] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, in: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, R. Garnett (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.

[46] S. Narayan, S. B. Cohen, M. Lapata, Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization, in: E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii (Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language

Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1797–1807. URL: https://aclanthology.org/D18-1206/. doi:10.18653/v1/D18-1206.

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

[48] C. Tan, V. Niculae, C. Danescu-Niculescu-Mizil, L. Lee, 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, 2016. URL: http://dx.doi.org/10.1145/2872427.2883081. doi:10.1145/2872427.2883081.

[49] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, 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.

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

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

[52] J. Bevendorff, D. Dementieva, M. Fröbe, B. Gipp, A. Greiner-Petter, J. Karlgren, M. Mayerl, P. Nakov, A. Panchenko, M. Potthast, A. Shelmanov, E. Stamatatos, B. Stein, Y. Wang, M. Wiegmann, E. Zangerle, Overview of PAN 2025: Voight-Kampff Generative AI Detection, Multilingual Text Detoxification, Multi-Author Writing Style Analysis, and Generative Plagiarism Detection, in: J. C. de Albornoz, J. Gonzalo, L. Plaza, A. G. S. de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Sixteenth International Conference of the CLEF Association (CLEF 2025), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2025.

[53] J. Bevendorff, Y. Wang, J. Karlgren, M. Wiegmann, A. Tsivgun, J. Su, Z. Xie, M. Abassy, J. Mansurov, R. Xing, M. N. Ta, K. A. Elozeiri, T. Gu, R. V. Tomar, J. Geng, E. Artemova, A. Shelmanov, N. Habash, E. Stamatatos, I. Gurevych, P. Nakov, M. Potthast, B. Stein, Overview of the "Voight-Kampff" Generative AI Authorship Verification Task at PAN and ELOQUENT 2025, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2025.

[54] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241.

[55] O. Halvani, C. Winter, L. Graner, On the usefulness of compression models for authorship verification, in: Proceedings of the 12th International Conference on Availability, Reliability and Security, ARES '17, Association for Computing Machinery, New York, NY, USA, 2017. URL: https://doi.org/10.1145/3098954.3104050. doi:10.1145/3098954.3104050.

A. 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.

Furthermore, through an ablation study on Testbed 4 of the MAGE benchmark, 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 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. Both these experiments are thoroughly detailed in our original paper.

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

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 (Appendix C)
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 (6) 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

B. Additional Results.

For further analysis, we refer readers to our main paper [40], which includes: (1) domain-specific performance of DivEye, (2) model-specific performance of DivEye, (3) the relative importance of DivEye when used in a boosted ensemble, (4) DivEye's effectiveness across different base detectors, and (5) a breakdown of feature importance within DivEye. These additional evaluations further support the generality, complementarity, and interpretability of our approach.

C. Hyperparameter Settings

Table 2 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 two NVIDIA RTX 4060Ti (16 GB each), and reported results reflect the median of three runs.

Table 2
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