Detection of Partially-Synthesized LLM Text

Eric Lei[1,2]*, **Hsiang Hsu**[2], **Chun-Fu (Richard) Chen**[2]
[1]University of Pennsylvania, [2]JPMorganChase Global Technology Applied Research
elei@seas.upenn.edu, {hsiang.hsu, richard.cf.chen}@jpmchase.com

Abstract

Advances in large language models (LLM) have produced artificial text that appear increasingly human-like and difficult to detect with the human eye. In order to improve LLMs' safety and mitigate potential nefarious uses, it has been desirable to develop automated detectors that can differentiate human and LLM-written text. While recent work has focused on classifying entire text samples (e.g., paragraphs) as human or LLM-written, this paper investigates the setting where the text's individual segments (e.g., sentences) could each be written by either a human or LLM. We study two relevant problems: (i) estimating the percentage of a text that was LLM-written, and (ii) determining which segments were LLM-written. To this end, we propose *Partial-LLM Detector* (PaLD), a black-box method that leverages the scores of text classifiers. Experimentally, we demonstrate the effectiveness of PaLD compared to baseline methods that build on prior text detectors.

1 Introduction

In recent years, large language model (LLM) capabilities have grown immensely [1, 9, 6], producing artificial text that appear convincingly human-like. As a result, LLM-generated text have been increasingly used across all aspects of society and industries [9, 3, 22]. Throughout its usage, LLM-generated text is difficult to detect with the human eye [19, 10, 15, 12], with recent GPT-4 models [1] shown to have success impersonating humans [14]. This introduces challenges for fair student assessment, fake news and information, copyright infringement, and many more scenarios where humans can be easily fooled by artificially-written text [3, 8]. As a solution to this, many recent tools to automatically detect whether text is LLM- or human-generated have been developed.

Most of these LLM detector works [10, 13, 21, 18, 28] approach the problem in the binary classification setup: given a piece of text (e.g., paragraph), the goal is to classify it as either human- or LLM-generated. The implicit assumption made in these works is that a piece of text to be classified is *entirely* human- or LLM-generated. In practice, however, this may not always be the case. A more realistic use-case is the *mixed-text* setting, shown in Fig. 1, where a text of interest may consist of both human and LLM portions. Potential causes of this may include, but are not limited to, the case when human-written text is edited or refined by a LLM before it is deployed. As a result, we investigate

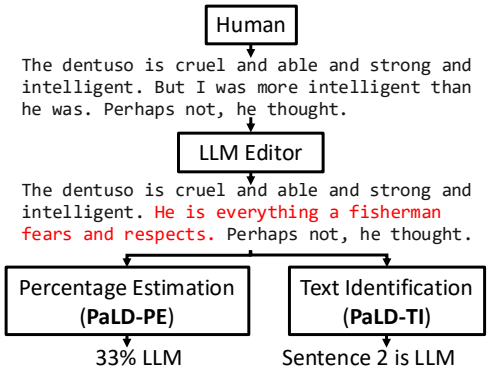

Figure 1: In practice, text encountered may be partially-LLM written. PaLD enables LLM percentage estimation and LLM text identification.

*Work done during the internship at JPMorganChase.

Socially Responsible Language Modelling Research (SoLaR) Workshop at NeurIPS 2024.

LLM-generated text detection in the mixed-text setting, in which two problems naturally arise that go beyond the binary classification setup: (i) *percentage estimation*, which estimates the amount of text that was LLM-generated, and (ii) *text identification*, which predicts the text segments that were LLM-generated. Shown in Fig. 1, percentage estimation aims to identify the amount of LLM content, regardless of their location within the text, whereas text identification provides more fine-grained information by identify which portions were LLM-written.

In what follows, we develop the Partial-LLM Detector (PaLD) framework for solving both problems. For former problem, PaLD-PE (Percentage Estimation) measures the distribution shift as text varies from fully-human to fully-LLM. The shift in distribution is then used to compute predictions of the LLM-written percentage on a given mixed text. For the latter problem, PaLD-TI (Text Identification) solves a subset selection problem which attempts to search for the subset of text segments that is maximally indicative of having been written by a LLM. Our contributions are the following.

1) We formulate the mixed-text setting in Sec. 3, and pose the percentage estimation and LLM text identification problems. These problems present a greater challenge than the binary classification setup, as they require more fine-grained information about the text.
2) For percentage estimation, we propose PaLD-PE, a Bayesian framework for producing point and interval estimates of a text's ground-truth LLM percentage. For LLM text identification, we propose PaLD-TI, a subset selection approach for finding LLM-written segments of the text.
3) We experimentally demonstrate that our method outperforms baseline methods based on prior works in LLM text detection across a variety of datasets, in Sec. 4.

2 Related Work

Many prior works have been developed to solve the LLM detection problem in the binary classification setup [10, 13, 21, 18, 28, 5]. These works are typically separated into white-box methods, which have access to LLM token likelihoods of the model that generated the text, and black-box methods, which do not. In general, these methods either discover a signature particular to LLM text that can be used to discriminate between LLM and human-written text, or design/train such a signature. For the former, Gehrmann et al. [10] designs a suite of statistical tools to aid in LLM text detection based on top-k log probabilities. Solaiman et al. [27] and Ippolito et al. [13] use the average log probability under the LLM as a signature to threshold, whereas Mitchell et al. [21] studies the curvature of the model's log probability; this was further explored and improved by Mireshghallah et al. [20] and Bao et al. [2]. Another line of work is Mao et al. [18], which notes that LLMs tend to repeat LLM-written text more often than human-written text, as use this as a signature to threshold. Sadasivan et al. [25] characterize the performance of binary classification of LLM text from a fundamental perspective. Regarding the latter, Guo et al. [12], Chen et al. [5] fine-tune RoBERTa [17] models to classify text as LLM- or human-written. Verma et al. [28] improves upon their generalization performance by using a logistic regression classifier on top of feature-selected LLM token probabilities.

3 PaLD: Partial-LLM Detector

We formulate the mixed-text setting as follows. Let $x = x_1 \dots x_n$ be a text composed of concatenated segments $\{x_i\}_{i=1}^n$. For example, these segments could be the sentences within a paragraph x. We then assume that each segment x_i has either been written by a human or LLM. Note that previous works focus on the binary classification case where $\{x_i\}_{i=1}^n$ are either all human or all LLM.

3.1 Text Scores T

We first introduce the concept of a text score T, which forms the backbone of many prior works that classify text as LLM or human. These text scores T, which we refer to as T-scores, return a real number $T(x)$ when presented with a text x. They are designed to be indicative of whether or not the text was LLM- or human-written. Specifically, if x_H and x_L are random fully-human and fully-LLM texts, then the desired property is that the distributions of $T(x_H)$ and $T(x_L)$ result in two statistically separated modes. To classify text, T is simply thresholded. T-scores can be constructed from pre-trained models, as in DetectGPT [21], or trained explicitly for the binary classification task, as in Ghostbuster [28] or RoBERTa-based models [12], or by combining both [18]. For instance, DetectGPT computes $T(x) = \log p_\theta(x) - \mathbb{E}_{\tilde{x}}[\log p_\theta(\tilde{x})]$, representing a LLM's log probability

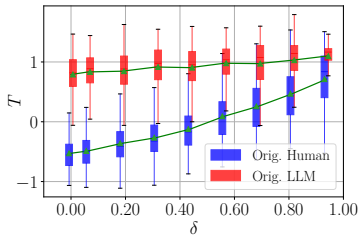

Figure 2: Distribution shift of T with GPT-4o fraction δ. Boxplots reflect $T(x)$ samples.

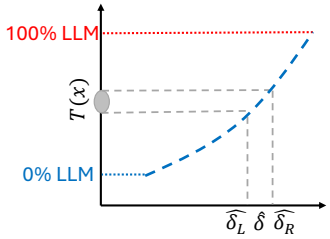

Figure 3: PaLD-PE for percentage estimation returns point $\hat{\delta}$ and interval $(\widehat{\delta_L}, \widehat{\delta_R})$ estimates.

curvature, where $\tilde{x}$ are perturbed versions of x generated by T5 [24], and p_θ is a LLM likelihood. Classifiers that are trained for binary classification yield T-scores defined by the logits of the model.

Our method relies on the the T-score concept for both percentage estimation and LLM text identification problems. For a mixed-text x, let $\delta \in [0, 1]$ denote the fraction of text in x that was LLM-generated. While $T(x_H)$ ($\delta = 0$) and $T(x_L)$ ($\delta = 1$) result in two separated modes, we would expect that for a mixed text x ($0 < \delta < 1$), $T(x)$ would result in a mode in between that of $T(x_H)$ and $T(x_L)$. Thus, the distribution shift of T should provide an indicator of the degree to which a text is mixed. We experimentally demonstrate this in Fig. 2. We track the shift in distribution of T (computed by RoBERTa logits) for text that is originally fully-human (Yelp reviews [30]) or originally fully-LLM (LLM-rewritten version). The distribution shift is observed as the original text is progressively overwritten by GPT-4o [1] using a sentence-level mask-and-fill approach, where δ represents the fraction of GPT-4o text. At $\delta = 0$, there is a clear separation between fully-human and fully-LLM modes. As δ increases from 0 to 1, the originally fully-human text (blue curve) yields a T-score distribution shift that smoothly transitions from the fully-human mode to the fully-LLM mode. In contrast, the originally fully-LLM text (red curve) yields a T-score distribution that roughly stays the same as more of the text is overwritten by GPT-4o. Further details for how GPT-4o is used can be found in Sec. A.1. This observation implies that in addition to discriminating fully-human and fully-LLM text, the T-score provides values correlated with the amount of LLM text in a mixed text.

3.2 Percentage Estimation

Given a mixed-text x with ground-truth LLM percentage δ, we would like to produce either a point estimate $\widehat{\delta}$ of δ, or a predictive interval $(\widehat{\delta_L}, \widehat{\delta_R})$ that contains δ with high probability. A predictive interval provides the user with a measure of confidence on the estimated percentage value.

At a high-level, our approach PaLD-PE first estimates the joint statistics between the LLM text percentage and the T-scores, then uses this model to return point estimates and/or predictive intervals of δ; see Fig. 3. In the first step, mixed texts are generated from a fully-human dataset to measure the shift in distribution of the text score T as the LLM percentage ranges from 0 to 1; we then fit a mixture kernel density estimate (KDE) to estimate the likelihood $P(T|\delta)$. In the second step, when we estimate the LLM percentage of an unseen text sample, we use the posterior $P(\delta|T)$ to return maximum a posteriori (MAP) estimates for $\widehat{\delta}$ and highest density intervals (HDI) for $(\widehat{\delta_L}, \widehat{\delta_R})$.

Measuring the $P(T|\delta)$ likelihood. We first generate synthetic mixed texts by using a LLM to mask-and-fill randomly masked out sentences of fully-human texts (Sec. A.1). The T-scores are then binned to a set of target levels $0 \leq \delta_1 < \cdots < \delta_K \leq 1$. All the T-score values binned to δ_k are assumed drawn from the $P(T|\delta = \delta_k)$ distribution. To parameterize a model for the likelihood $P(T|\delta)$, for any $\delta \in [0, 1]$, we use a mixture of KDEs. Specifically, let $\phi_k(T) = \frac{1}{n_k} \sum_{i=1}^{n_k} \frac{1}{h} K(^{T-T_i^{(k)}}/_h)$ be a Gaussian KDE fit to the samples $\{T_i^{(k)}\}_{i=1}^{n_k}$ representing the $P(T|\delta = \delta_k)$ conditional. Here, $K(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2}$, n_k is the number of T-score samples collected at δ_k, and h is a bandwidth parameter. Then, our mixture KDE computes the likelihood as $P(T|\delta) = \theta\phi_{k^*}(T) + (1-\theta)\phi_{k^*+1}(T)$, where k^* is the index such that $\delta_{k^*} \leq \delta < \delta_{k^*+1}$, and $\theta = \frac{\delta_{k^*+1} - \delta}{\delta_{k^*+1} - \delta_{k^*}}$.

Percentage prediction. To predict δ, we use the posterior density $P(\delta|T(x)) \propto P(T(x)|\delta)P(\delta)$, where we assume a prior distribution $P(\delta)$ over $0 \leq \delta \leq 1$. For the point estimate, we return

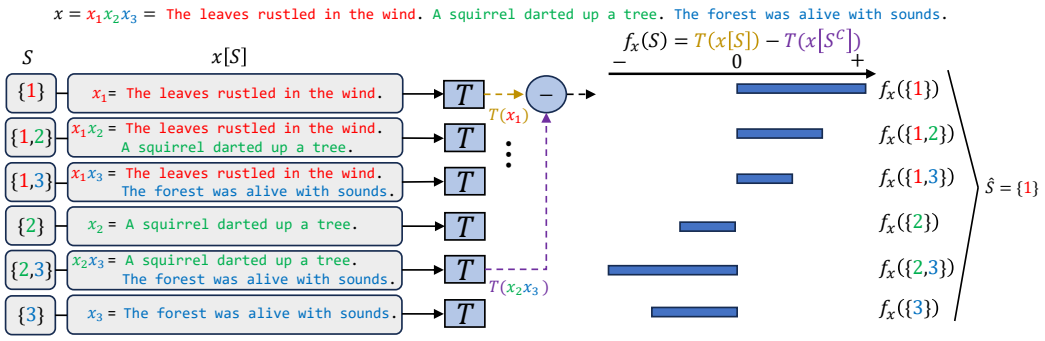

Figure 4: PaLD-TI for LLM text identification. Left: stitched texts $x[S]$ are enumerated, and T-scores computed. Middle: $f_x(S)$, the T-score difference, is computed for all S. We illustrate this for $S = \{1\}$. Right: the maximum $f_x(S)$ is computed, and its maximizing set $\hat{S}$ is returned as the segment indices predicted as LLM.

$\hat{\delta} = \arg\max_\delta P(\delta|T(x))$. For the predictive interval, we return the $(1-\alpha)$-HDI [4],

$$\arg\min_{0\le\delta_L<\delta_R\le 1} |P(\delta_R|T(x)) - P(\delta_L|T(x))| + |F(\delta_R|T(x)) - F(\delta_L|T(x)) - (1-\alpha)|, \quad (1)$$

where $F(\delta|T)$ is the cumulative distribution function of $P(\delta|T)$, and α is a parameter we can set to control the posterior probability that δ is contained in the interval. In practice, since we do not have the exact posterior density, we use a Markov Chain Monte Carlo approach by sampling $\delta'_1, \ldots, \delta'_M$ from the posterior $P(\delta|T)$ via Metropolis-Hastings [11]. For the MAP estimate, we return the sample mode $\hat{\delta} = \arg\max_{1\le i\le M} P(\delta'_i|T(x))$. For the $(1-\alpha)$-HDI estimate, we return $(\widehat{\delta_L}, \widehat{\delta_R}) = (\delta'_{(i^*)}, \delta'_{(i^*+[(1-\alpha)M])})$, where $\delta'_{(i)}$ is the i-th smallest sample, and $i^* = \arg\min_{1\le i\le M} \delta'_{(i^*+[(1-\alpha)M])} - \delta'_{(i^*)}$ [4].

3.3 Text Identification

We assume there is a minimal level of granularity for segmenting the texts (i.e., given a segmentation $x = x_1 \ldots x_n$, each x_i can not be infinitely segmented); in this paper, we consider the minimal segment to be one sentence. The goal is to return an index set $\hat{S} \subseteq \{1, \ldots, n\}$ corresponding to a segmentation $x = x_1 \ldots x_n$ such that $\{x_i : i \in \hat{S}\}$ contains all the LLM-written segments.

One baseline approach could be to classify each x_i individually using one of the binary classification approaches. However, these methods are designed for longer texts, and are known to perform poorly on short texts [28]. Our results in Sec. 4 demonstrate the poor performnace of this approach. Instead, we propose to "stitch" together different segments of x to construct stitched texts, see Fig. 4. Concretely, let $S \subseteq \{1, \ldots, n\}$ be an index set which we use to select a subset of the segments in x. Define $x[S]$ to be the text formed by concatenating the segments of x indexed by S in the order of the indices. For example, if $S = \{1, 3, 4\}$, then $x[S] = x_1x_3x_4$. These stitched texts $x[S]$ mostly consist of multiple segments, and should be of sufficient length for some of the binary classifier methods to be effective. Drawing from observations in Fig. 2, for the S that selects all the LLM text, $S^{\complement} := \{1, \ldots, n\} \setminus S$ will select all human text, and this should result in a large discrepancy between $T(x[S])$ and $T(x[S^{\complement}])$. On the other hand, if S contains a mixture of LLM and human text, then so will $S^{\complement}$, and the T-scores should be more similar.

Thus, our goal is to find the S that the T-score maximally discriminates $x[S]$ and $x[S^{\complement}]$:

$$\hat{S} = \arg\max_{S\subseteq\{1,\ldots,n\}} f_x(S) := T(x[S]) - T(x[S^{\complement}]). \quad (2)$$

The overall method, which we call PaLD-TI, is shown in Fig. 4. In practice, we disregard $S = \emptyset$ and $S^{\complement} = \emptyset$, yielding a total of $2^n - 2$ sets to consider. Thus, Eq. 2 is a subset selection problem which is combinatorial and has complexity exponential in n. In our experiments, we consider texts with at most $n = 10$ segments which is feasible to solve Eq. 2 exactly. For texts with more sentences, one can chunk the text into paragraphs before solving Eq. 2, or use approximate algorithms (e.g., greedy)

Table 1: Point estimate results; mean absolute error. Top method is bolded.

Dataset	**PaLD-PE (Ours)**	**PaLD-TI (Ours)**	DetectGPT-Seg	RoBERTa-Seg	RoBERTa-LN-Seg	Ghostbuster-Seg	RoBERTa-Reg	RoBERTa-QuantileReg
WP	**0.116**	0.210	0.381	0.342	0.434	0.215	0.207	0.186
Yelp	**0.137**	0.177	0.407	0.382	0.437	0.282	0.181	0.175

that trade-off optimality for efficiency. Note that PaLD-TI can be used for percentage estimation via the total predicted LLM length. Details regarding hyperparameters can be found in Sec. A.2.

4 Empirical Study

In our experiments, we use the WritingPrompts (WP) [7] and Yelp Reviews (Yelp) [30] dataset which are fully-human texts typically used to benchmark LLM detection. For LLM-written texts, we prompt the LLM to rewrite a human-written text sample. For mixed texts, we adopt a sentence-level mask-and-fill approach with GPT-4o [1]. Fully-LLM-written and mixed text versions of both datasets are generated for both train and test splits. For mixed texts, the training split is generated at character-level model percentages of approximately $0.1, 0.2, \ldots, 0.9$. For the testing split, the ground-truth percentages are approximately $0.25, 0.5, 0.75$. Further details can be found in Sec. A.1. We report performance averaged across $0.25, 0.5, 0.75$ in the main text, and individually in Sec. A.3.

Percentage estimation results. We report the point estimate results in Tab. 1, 5. For baseline methods, we use binary classifiers such as DetectGPT [21], Ghostbuster [28], and a RoBERTa baseline [28], applied segment-wise to the mixed-text samples. DetectGPT's threshold is set such that the false-positive and true-positive rates are equal in binary classification setting. The predicted percentage is then the total character length of the predicted model segments divided by the total number of characters of the text. We use PaLD-TI here in a similar fashion. Also, we fine-tune a RoBERTa model for regression to predict δ with square loss. PaLD-PE uses RoBERTa logits as the T-score, whereas PaLD-TI uses DetectGPT. Each method's point estimate $\widehat{\delta}$ is measured by absolute error $|\widehat{\delta} - \delta|$, where δ is the ground-truth fraction of LLM-written text measured at the character-level. We see that our method's performance is superior to all the segment baselines, indicating that the binary classifiers applied segment-wise do not perform well. The RoBERTa models trained on mixed-text data are competitive, but our method results in better overall accuracy. Additionally, PaLD-TI is superior to all segment models, and competitive with regression models.

For predictive intervals, we compare with a RoBERTa model trained on mixed-text data with quantile regression [23, 16], which can return prediction intervals. We report two metrics: *coverage* (C), the frequency the interval covers δ (i.e., $\widehat{\delta_L} \leq \delta \leq \widehat{\delta_R}$), and *precision* (P), the width of the interval (i.e., $\widehat{\delta_R} - \widehat{\delta_L}$). To provide a comparison, for each dataset, we tune α for our method and the quantiles for the baseline so that the coverage is on average 85%. Tab. 2, 6 show on average, our method yield a superior precision-coverage tradeoff compared to the quantile regression baseline.

Table 2: Interval estimate results.

Dataset	**PaLD-PE (Ours)** C ↑	**PaLD-PE (Ours)** P ↓	RoBERTa-QuantileReg C ↑	RoBERTa-QuantileReg P ↓
WP	84%	0.385	74%	0.578
Yelp	86%	0.470	83%	0.633

Text identification results. Tab. 3, 7 shows the LLM text identification performance. We compare PaLD-TI with DetectGPT and Ghostbuster applied segment-wise. We evaluate the segment accuracy, i.e., the fraction of correctly classified segments. We see that PaLD-TI outperforms the segment-wise baselines by 10-21% in terms of segment accuracy when averaged over the test datasets. For PaLD-TI, we also report the top-1 and top-p accuracy, which measures when the ground-truth model segments lie in the top p fraction of all $f_x(S), S \subseteq \{1, \ldots, n\}$, in Tab. 8. We see that even if the ground-truth is not recovered by $\hat{S}$, it is in the top 0.2 fraction of our objective function more than 60% of the time. This further supports the validity of $f_x(S)$ in determining LLM-written segments.

Table 3: LLM Text Identification results; segment-wise accuracy.

Dataset	**PaLD-TI (Ours)**	DetectGPT-Seg	Ghostbuster-Seg
WP	**0.690**	0.486	0.599
Yelp	**0.676**	0.452	0.540

5 Final Remarks

PaLD-TI is NP-hard and does not scale well with the number of segments. While it is suitable for paragraphs, longer texts may require approximate methods or a chunking approach; we leave for future work. Additionally, PaLD-TI requires a fixed segmentation. A chosen segmentation may not align perfectly with ground-truth segmentation. Future work can investigate the effect of misalignment and mitigate any resulting suboptimality.

Social impacts statement. LLM text detection has significant social, ethical, and practical implications. Our work contributes to the broader effort to maintain the integrity of human communication and authenticity, as well as mitigate risks associated with misuse of AI-generated text.

Disclaimer. This paper was prepared for informational purposes by the Global Technology Applied Research center of JPMorgan Chase & Co. This paper is not a product of the Research Department of JPMorgan Chase & Co. or its affiliates. Neither JPMorgan Chase & Co. nor any of its affiliates makes any explicit or implied representation or warranty and none of them accept any liability in connection with this paper, including, without limitation, with respect to the completeness, accuracy, or reliability of the information contained herein and the potential legal, compliance, tax, or accounting effects thereof. This document is not intended as investment research or investment advice, or as a recommendation, offer, or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] G. Bao, Y. Zhao, Z. Teng, L. Yang, and Y. Zhang. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*, 2023.

[3] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

[4] M.-H. Chen and Q.-M. Shao. Monte carlo estimation of bayesian credible and hpd intervals. *Journal of Computational and Graphical Statistics*, 8(1):69–92, 1999. ISSN 10618600. URL http://www.jstor.org/stable/1390921.

[5] Y. Chen, H. Kang, V. Zhai, L. Li, R. Singh, and B. Raj. Gpt-sentinel: Distinguishing human and chatgpt generated content. *arXiv preprint arXiv:2305.07969*, 2023.

[6] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

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

[8] M. Farina and A. Lavazza. Chatgpt in society: emerging issues. *Frontiers in Artificial Intelligence*, 6, 2023. ISSN 2624-8212. doi: 10.3389/frai.2023.1130913. URL https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2023.1130913.

[9] L. Floridi and M. Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds Mach.*, 30 (4):681–694, dec 2020. ISSN 0924-6495. doi: 10.1007/s11023-020-09548-1. URL https://doi.org/10.1007/s11023-020-09548-1.

[10] S. Gehrmann, H. Strobelt, and A. Rush. GLTR: Statistical detection and visualization of generated text. In M. R. Costa-jussà and E. Alfonseca, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-3019. URL https://aclanthology.org/P19-3019.

[11] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd ed. edition, 2004.

[12] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, and Y. Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*, 2023.

[13] D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck. Automatic detection of generated text is easiest when humans are fooled. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 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.

[14] C. Jones and B. Bergen. Does GPT-4 pass the Turing test? *arXiv preprint arXiv:2310.20216*, 2023.

[15] C. R. Jones and B. K. Bergen. People cannot distinguish gpt-4 from a human in a turing test. *arXiv preprint arXiv:2405.08007*, 2024.

[16] R. Koenker and G. Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1913643.

[17] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

[18] C. Mao, C. Vondrick, H. Wang, and J. Yang. Raidar: generative AI detection via rewriting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=bQWE2UqXmf.

[19] Q. Mei, Y. Xie, W. Yuan, and M. O. Jackson. A turing test of whether ai chatbots are behaviorally similar to humans. *Proceedings of the National Academy of Sciences*, 121(9):e2313925121, 2024. doi: 10.1073/pnas.2313925121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2313925121.

[20] N. Mireshghallah, J. Mattern, S. Gao, R. Shokri, and T. Berg-Kirkpatrick. Smaller language models are better black-box machine-generated text detectors. *arXiv preprint arXiv:2305.09859*, 2023.

[21] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR, 2023.

[22] M. Mitchell and D. C. Krakauer. The debate over understanding in ai's large language models. *Proceedings of the National Academy of Sciences*, 120(13):e2215907120, 2023. doi: 10.1073/pnas.2215907120. URL https://www.pnas.org/doi/abs/10.1073/pnas.2215907120.

[23] O. H. M. Padilla, W. Tansey, and Y. Chen. Quantile regression with relu networks: Estimators and minimax rates. *Journal of Machine Learning Research*, 23(247):1–42, 2022. URL http://jmlr.org/papers/v23/21-0309.html.

[24] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[25] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023.

[26] D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. A Wiley-interscience publication. Wiley, 1992. ISBN 9780471547709. URL https://books.google.com/books?id=7crCUS_F2ocC.

[27] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

[28] V. Verma, E. Fleisig, N. Tomlin, and D. Klein. Ghostbuster: Detecting text ghostwritten by large language models. *arXiv preprint arXiv:2305.15047*, 2023.

[29] H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on machine learning*, pages 23631–23644. PMLR, 2022.

[30] Yelp. Yelp review dataset, 2014. URL http://www.yelp.com/dataset_challenge.

A Appendix

A.1 Model Text Generation

We use a mask-and-fill approach to generate synthetic mixed texts. This process is used for both generating the histograms in Fig. 2 and fitting the mixture KDE model in Sec. 3. Specifically, if we denote x^H as a text from our human-written dataset (e.g. WritingPrompts or Yelp Reviews), then $x^H = x_1^H \dots x_n^H$ is segmented at the sentence level. Then, segments are randomly masked, to form a masked text x^M. For example, a 5-segment human text $x^H = x_1^H x_2^H x_3^H x_4^H x_5^H$, masked at sentences 2 and 5, would become $x^M = x_1^H \texttt{[MASK]} x_3^H x_4^H \texttt{[MASK]}$. GPT-4o is then prompted with the following prepended to x^M:

"`What sentences should go in the` n_{replace} `[MASK] locations of the following text? Only provide exactly one sentence per [MASK] location. Only provide the sentences as a numbered list with` n_{replace} `sentences total.`"

Here, n_{replace} is the number of `[MASK]` symbols in x^M. We found that when prompted in this way, GPT-4o successfully returns a numbered list containing exactly n_{replace} sentences corresponding to the `[MASK]` symbols, with a failure rate of 1-2%. When successful, the sentences in the numbered list are inserted into the `[MASK]` positions of the masked text, to form the mixed text. We show a few examples of these mixed texts in Tab. 4.

Since this process does not control the length of the segments returned by GPT-4o, it is not easy to exactly control the fraction of LLM text δ at the character level. As a result, we target δ approximately by masking and filling a δ fraction of the human sentences. This is done for both the training split (targeting δ approximately at $0.1, 0.2, \dots, 0.9$) and testing splits (targeting δ approximately at $0.25, 0.5, 0.75$). The true δ (at the character level), which can be computed post-hoc, is used for (i) the binning step in PaLD-PE collect the T-score samples for the $P(T|\delta = \delta_k)$ distributions, and (ii) to benchmark the performance metrics (absolute error, coverage, and prediction) for percentage estimation.

A.2 PaLD Implementation Details

During the distribution-fitting stage of PaLD-PE, the entire training split of a dataset is masked and filled using the above procedure, where the fraction of sentences masked is approximately $\delta_1 = 0.1, \delta_2 = 0.2, \dots, \delta_9 = 0.9$. We then we compute T-scores across all the text samples using the logits of a RoBERTa model trained with the LogitNorm loss [29]. We found this T-score to work the best for percentage estimation, as the LogitNorm improves calibration of the model which is necessary for the T-score to smoothly interpolate between fully-human and fully-LLM. We use LogitNorm with temperature $\tau = 0.005$, and train the RoBERTa model on the training split for the respective datasets. These T-scores and then binned to $\delta_1 = 0.1, \delta_2 = 0.2, \dots, \delta_9 = 0.9$, using the character-level fraction of LLM text in the mixed text. Then, a KDE with Gaussian kernel, with bandwidth chosen using Scott's rule [26], is fit to the T-scores at each level of δ_k to form the likelihood model $P(T|\delta)$ as described in the main text. For the posterior, we choose the prior $P(\delta)$ to be the Beta$(2, 2)$ distribution. During the inference stage, we sample 5000 samples, discarding the first 1000 due to burn-in, using Metropolis-Hastings [11] with a proposal distribution as the truncated normal centered at the previous sample, truncated to $[0, 1]$. The predicted percentage and interval are given by the MAP estimate and $(1 - \alpha)$-HDI interval, respectively.

A.3 Additional Results

Here, we provide percentage estimation and LLM text identification results split by ground truth δ, in addition to top-p performance of PaLD-TI.

Table 4: Examples of synthetic mixed text generation using the mask-and-fill approach with GPT-4o. Text highlighted in red were originally segments masked in the human text and filled by GPT-4o.

Human Text	Mixed Text
Thank you for a lovely morning! I was in NJ early and decided to stop in for a delicious diner breakfast. I got a taylor ham, egg, and cheese and a short stack of blueberry pancakes (I simply couldn't decide between sweet or savory, plus leftovers are never a bad thing). The egg sandwich was fantastic. I was nervous when they said they didn't have kaiser rolls so I went with a hamburger roll. It was an excellent decision. The roll was excellent - not just an average cheap hamburger roll. They layered the cheese on the sandwich, which to me is a must for a true egg sandwich. It was served with home fries, which were sauteed with large slices of peppers and onions.	This morning, I decided to treat myself to breakfast at a local diner. I got a taylor ham, egg, and cheese and a short stack of blueberry pancakes (I simply couldn't decide between sweet or savory, plus leftovers are never a bad thing). The egg sandwich was fantastic. I was nervous when they said they didn't have kaiser rolls so I went with a hamburger roll. It was an excellent decision. The pancakes were fluffy and bursting with blueberries. They layered the cheese on the sandwich, which to me is a must for a true egg sandwich. The meal also came with a side of breakfast potatoes.
This was a very busy place. Was told I had to try this place while I was in St Louis. I was not disappointed. I got a peach shake that was amazing. The rest of my group tried a number of different options that they all enjoyed. The wait was a little long especially on a very hot date. The prices were very reasonable. The ordering at the window was confusing. Multiple windows with not a lot of direction of which line to get in and where to wait for the food. It's a great place to stop if you are in the area.	I recently visited a local ice cream shop. The place had a charming, old-fashioned vibe. I was not disappointed. I got a peach shake that was amazing. The staff was friendly and helpful. The menu had a wide variety of flavors and treats. The prices were very reasonable. The ordering at the window was confusing. Multiple windows with not a lot of direction of which line to get in and where to wait for the food. It's a great place to stop if you are in the area.
A man is banished to the wilderness for 20 years. Write his diary entries for his first and last days of exile. I was born to fire. It flowed over my skin, danced upon my face, and stripped me of what little humanity I had left. Within the ruined cavity of my left eye I held the final images of my family as they were fed to the same fires I was pulled from. My death would not be so quick and so I was allowed to burn with them, but live. As soon as I was able to walk, I was ushered out into the wilderness. The final piece of society I was allowed to keep was in the ink buried in my chest that had once formed my son's hand print, now twisted with my burned skin into a misshapen claw. They promised twenty years, but swore under their breath	A man is banished to the wilderness for 20 years. Write his diary entries for his first and last days of exile. Today marks the beginning of my exile, a punishment I must endure for the next two decades. The pain of separation from my loved ones is unbearable, but I must find the strength to survive. Within the ruined cavity of my left eye I held the final images of my family as they were fed to the same fires I was pulled from. My death would not be so quick and so I was allowed to burn with them, but live. As soon as I was able to walk, I was ushered out into the wilderness. The years have been long and arduous, but I have learned to find solace in the solitude of the wilderness. As I take my final steps back to civilization, I carry with me the scars and wisdom of my exile
Describe an object within five feet of you in as much detail as possible. A pair of simple black converse lie on the floor of the baby blue Honda fit my girlfriend is kind enough to let me drive. They are a far cry from the crisp kicks I'd received in the mail only a year ago. This has been a hard 12 months for them. The once crisp white inner lining has degraded into something a generous person might call "cream" or "off-white" to me they're just brown. The forces of time have transmuted the laces into a soft grey, like clouds in fall which promise a gentle patter of rain to listen to as you while away the hours. The rubber has had it particularly bad, time and constant use has worn down the bottom edges. Scuff marks cover the once pristine expanse. When they were new I'd taken, so much care to keep them scuff free I waited until the wedding to wear them.	Describe an object within five feet of you in as much detail as possible. A pair of simple black converse lie on the floor of the baby blue Honda fit my girlfriend is kind enough to let me drive. They are a far cry from the crisp kicks I'd received in the mail only a year ago. The laces are frayed and stained, no longer the bright white they once were. The once crisp white inner lining has degraded into something a generous person might call "cream" or "off-white" to me they're just brown. The rubber soles are worn down, evidence of countless steps taken. The black canvas is faded, showing signs of wear and tear from daily use. Scuff marks cover the once pristine expanse. When they were new I'd taken, so much care to keep them scuff free I waited until the wedding to wear them.

Table 5: Point estimate results broken down by ground truth δ; mean absolute error. Top method is bolded; † indicates 2nd-best.

Dataset	**PaLD-PE (Ours)**	**PaLD-TI (Ours)**	DetectGPT-Seg	RoBERTa-Seg	RoBERTa-LN-Seg	Ghostbuster-Seg	RoBERTa-Reg	RoBERTa-QuantileReg
WP-25%	**0.125**	0.293	0.166	0.148	0.212	0.141†	0.346	0.281
WP-50%	0.100†	0.202	0.359	0.330	0.418	0.176	0.151	**0.089**
WP-75%	0.121†	0.136	0.617	0.548	0.617	0.327	**0.123**	0.188
Yelp-25%	**0.150**	0.220	0.194	0.195	0.216	0.197	0.129†	0.180
Yelp-50%	0.115†	0.156	0.387	0.360	0.425	0.229	**0.099**	0.082
Yelp-75%	**0.146**	0.155†	0.640	0.591	0.670	0.420	0.314	0.262

Table 6: Interval estimate results broken down by ground truth δ.

Dataset	**PaLD-PE (Ours)**		RoBERTa-QuantileReg	
	C ↑	P ↓	C ↑	P ↓
WP-25%	77%	0.373	30%	0.570
WP-50%	93%	0.398	98%	0.577
WP-75%	83%	0.386	95%	0.588
Yelp-25%	82%	0.466	53%	0.631
Yelp-50%	90%	0.477	99%	0.633
Yelp-75%	85%	0.466	97%	0.635

Table 7: LLM Text Sdentification results broken down by ground truth δ; segment-wise accuracy.

Dataset	**PaLD-TI (Ours)**	DetectGPT-Seg	Ghostbuster-Seg
WP-25%	0.600	0.632	**0.678**
WP-50%	**0.683**	0.500	0.596
WP-75%	**0.786**	0.325	0.524
Yelp-25%	0.613	0.641	**0.674**
Yelp-50%	**0.693**	0.476	0.537
Yelp-75%	**0.722**	0.240	0.410

Table 8: PaLD-TI top-1 and top-p performance.

Dataset	Top-1	Top-0.05	Top-0.20
WP-25%	0.000	0.243	0.460
WP-50%	0.086	0.343	0.714
WP-75%	0.162	0.568	0.811
Yelp-25%	0.020	0.143	0.429
Yelp-50%	0.071	0.265	0.724
Yelp-75%	0.071	0.357	0.745
WP (avg)	0.082	0.385	0.622
Yelp (avg)	0.054	0.255	0.633