Detecting Machine-Generated Long-Form Content with Latent-Space Variables

Yufei Tian **Zeyu Pan** **Nanyun Peng**

University of California, Los Angeles

yufeit@g.ucla.edu

Abstract

The increasing capability of large language models (LLMs) to generate fluent long-form texts is presenting new challenges in distinguishing machine-generated outputs from human-written ones, which is crucial for ensuring authenticity and trustworthiness of expressions. Existing zero-shot detectors primarily focused on token-level distributions, which are vulnerable to real-world domain shifts including different prompting and decoding strategies, and adversarial attacks. We propose a more robust method that incorporates abstract elements—such as *event transitions*—as key deciding factors to detect machine vs. human texts, by training a latent-space model on sequences of events or topics derived from human-written texts. On three different domains, machine generations which are originally inseparable from humans' on the token level can be better distinguished with our latent-space model, leading to a 31% improvement over strong baselines such as DetectGPT (Mitchell et al., 2023; Bao et al., 2024). Our analysis further reveals that, unlike humans, modern LLMs like GPT-4 generate event triggers and their transitions differently, an inherent disparity that help our method to robustly detect machine-generated texts.

1 Introduction

In today's digital world, large language models (LLMs) such as GPT-4 have transformed various daily tasks with their human-like text generation capability, such as drafting emails and essays. However, their potential misuse poses substantial risks including impersonation, misinformation, and academic dishonesty (Tang et al., 2024). This highlights the need for effective detection mechanisms. Existing AI content detectors can be categorized into 1) a priori methods such as watermarking (Kirchenbauer et al., 2023), 2) parameterized methods such as fine-tuned classifiers (Hu et al., 2023),

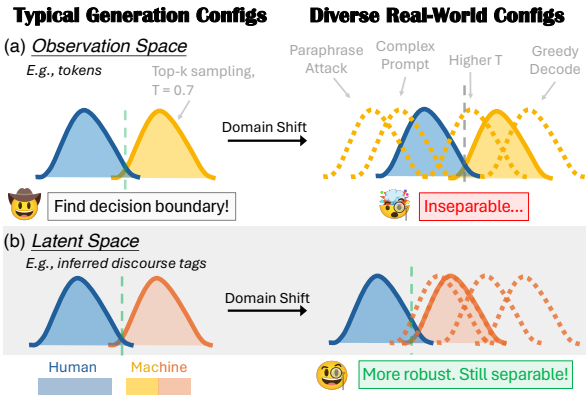

Figure 1: (a) Existing zero-shot detectors that rely on the token distributions (observation space statistics) are not robust to various real-world scenarios such as high decoding temperature, complex prompts, and adversarial attacks. (b) Our detector with latent features (*e.g.,* discourse tags) are more robust to these changes.

and 3) zero-shot methods that rely on certain statistical differences (Vasilatos et al., 2023; Mitchell et al., 2023; Bao et al., 2024). This paper focuses on the last due to its general pertinence in practice: end users may still choose non-watermarked LLMs outside the distribution of the fine-tuned classifiers (Yang et al., 2023; Ghosal et al., 2023).

Existing zero-shot methods to distinguish machine-generated texts (**MGTs**) from human-written texts (**HWTs**) typically assume an *unchanging* relationship between machine and human outputs, ignoring potential *distribution shifts* resulted from changes in generation setups (Zellers et al., 2019) or adversarial attacks (Shi et al., 2024; Wang et al., 2024). As is shown in Figure 1(a), prior zero-shot detection methods usually assume that MGTs consistently exhibit higher log-likelihood (or conditional curvature) than HWTs. However, certain changes—including increased decoding temperature, paraphrasing, and word substitution—can alter the distribution of MGTs (Hans et al., 2024), making MGTs and HWTs ***inseparable*** by log-likelihood after domain shift. In this work, we

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 10394–10408

further identify that complex prompts (*e.g.,* ask to prepare an outline before generation) can alter the typical relationship between MGTs and HWTs. Consequently, when considering MGTs from diverse sources in real world, existing detectors become less effective.

Towards more robust detection, we aim to answer the research question: *Is there a shared feature among all MGTs, regardless of the generation configurations and adversarial attacks?* Wang et al. (2022); Deng et al. (2022); Tian et al. (2024) argue that machine-generated long texts are fluent at word level, but lack discourse coherence in terms of *topic* or *event transitions*. We hence hypothesize that MGTs and HWTs are more separable in such ***latent space*** illustrated in Figure 1(b), because those underlying features can not be easily captured by next token probabilities (*i.e.,* optimization in observation[1] space). In addition, paraphrase or edit attacks change original texts with semantically similar yet lexically different alternatives. While they effectively alter token-level distributions, the high-level, hidden representations remain similar.

We test our hypothesis on three writing tasks: creative scripts, news, and academic essays. We explored latent variables like part-of-speech tags, topics, verbs, and event sequences. Specifically, we first train a lightweight model (62M transformer, half the size of gpt-2-small) on the latent variables inferred from HWTs, and then compare the latent distributions between HWTs and MGTs at test time. We find detectors in observation and latent space exhibit complementary strengths (§ 3.2) in differentiating MGTs from various configurations. Integrating the criteria from both types yields the best performance.

We explore five features to represent the underlying structure, and find that ***event trigger*** derived from information extraction models (Peng et al., 2023) is the most effective in separating HWTs from MGTs, outperforming strong token-space detector (Bao et al., 2024) by 31% in AUROC. Our analysis in § 5 further reveals that LLMs such as GPT-4 exhibit a different preference from human in choosing event triggers (for creative writing) and event transitions (for news and science), and such a disparity cannot be bridged through explicit planning of these latent structures.

To sum up, we identify key factors that deceive existing detectors in real-world scenarios. We then demonstrate a significant discrepancy in hidden structures between current LLMs and humans, especially the selection and transitions of event triggers. Building on these insights, we propose a novel detection framework that employs latent variables to robustly differentiate between human- and machine- generated texts.

2 Preliminary: Fragility of Existing Zero-Shot Detectors

In this section, we introduce two popular lines of zero-shot detection methods (logit-based and perturbation-based), and then illustrate how they are fragile to manipulations in decoding, variations in prompting style, and adversarial attacks.

2.1 Existing Detectors

Logit-Based Logit-based methods commonly employ probability metrics of tokens. Irene Solaiman (2019) established a strong baseline for detecting machine-generated text through the average log probability under the generative model. The intuition behind is that language model text generation is auto-regressive; the model selects tokens based on relatively higher probability at each decision point, resulting in MGT exhibiting a markedly higher average log probability compared to HWT, which becomes the foundational assumption of perplexity-based detectors (Vasilatos et al., 2023; Xu and Sheng, 2024) and rank-based detectors (Su et al., 2023), etc.

Perturbation-Based Another notable hypothesis introduced by Mitchell et al. (2023) posits and verifies that MGT tends to occur in regions of negative curvature within the language model's log probability function. Specifically, minor edits to MGT—referred to as perturbations—typically lead to a lower log probability under the model than the original text, whereas such rewrites of HWT may result in either higher or lower log probabilities. Bao et al. (2024) then increases its efficiency and efficacy by utilizing dual models that share the same tokenizer to expedite the perturbation process. Given text sample x and scoring model p_θ, conditional probability ***curvature*** is defined as:

$$\mathbf{d}\left(x, p_\theta, q_\varphi\right) = \frac{\log p_\theta(x \mid x) - \tilde{\mu}}{\tilde{\sigma}} \text{ where,} \quad (1)$$

$$\tilde{\mu} = \mathbb{E}_{\tilde{x}\sim q_\varphi(\tilde{x}|x)}\left[\log p_\theta(\tilde{x} \mid x)\right]$$
$$\tilde{\sigma}^2 = \mathbb{E}_{\tilde{x}\sim q_\varphi(\tilde{x}|x)}\left[\left(\log p_\theta(\tilde{x} \mid x) - \tilde{\mu}\right)^2\right]. \quad (2)$$

[1]We use *observation*, *sample*, and *token* interchangeably throughout this paper.

Method for Generation		Logit Based	Pert. Based
Default	T=0.7 Direct Generation	✓	✓
Decoding	T=1.0 Direct Generation	✓	✓
Prompting	Simple Complex	✓ ✗	✓ ✗
Attack	Paraphrase Edit	✗ ✗	✗ ✗

Table 1: Different methods for generation and whether existing zero-shot detectors are robust to them.

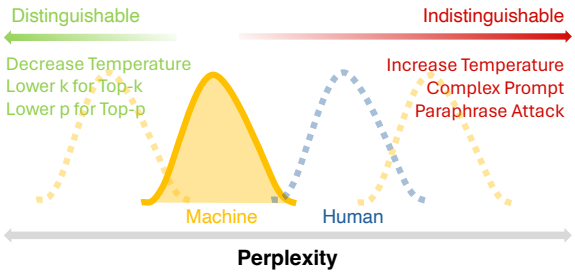

Figure 2: Both logit-based and perturbation-based detectors are not robust to changes in decoding, variations of prompting style, and adversarial attacks.

$\tilde{\mu}$ denotes the expected score of samples $\tilde{x}$ generated by the sampling model q_φ, and $\tilde{\sigma}^2$ is the expected variance of the scores.

2.2 Influential factors for detection

Despite the reported success, the robustness of the above methods are instinctively tied to text distribution, which are influenced by various factors, including but not limited to the following three settings (also summarized in Table 1):

1) Decoding Settings: As is shown in Figure 2, Zellers et al. (2019) argues that setting a higher temperature (*e.g.,* T=1.0), a higher k for top-k sampling can increase the likelihood of generating atypical sequences (*i.e.,* increased perplexity), which contradicts the assumption of logit- and perturbation-based methods (Irene Solaiman, 2019; Vasilatos et al., 2023).

2) Variations in Prompting: Changes in prompts can significantly influence the generation process and cause text distribution to be shifted as assessed by the proxy language model, particularly when the prompt is usually unknown to detectors. An illustrative example from Hans et al. (2024) using the prompt "Can you write a few sentences about a capybara that is an astrophysicist?" demonstrates how even seemingly simple prompts can lead to increased perplexity, as the probability that

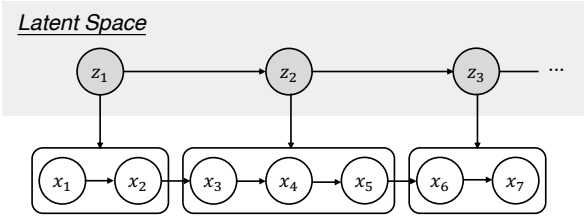

Figure 4: Generative process with latent variables.

"capybara being astrophysicist" is very low.

In addition, we investigate a planning-based prompting strategy, which emulates human drafting processes and has been widely adopted in neural long-text generation (Yao et al., 2019; Tian and Peng, 2022; Yang et al., 2022) for improved coherence. Our experiments in Table 2 reveal that multiple steps of planning, expansion, and revision (referred to as complex chains of prompt, shown in Figure 2) results in a significantly higher downstream text perplexity than direct generation (no chain of prompt) and one step of planning (simple chains of prompt).

3) Paraphrase/Edit Attacks: Rephrasing a portion of words (termed edit attack) or sentences (termed paraphrase attack) of the original article can dramatically alter the text distribution, too (Ghosal et al., 2023; Sadasivan et al., 2023; Shi et al., 2024). Such attacks disrupts the original auto-regressive properties, increases output perplexity, and changes text distribution to the indistinguishable region in Figure 2.

3 Machine-Content Detection with Latent Variables

In § 3.1, we formulate the next-token-prediction process with latent variables and introduce a neural model to learn the distributions of these variable. Next, in § 3.2, we propose a simple but effective method to combine the benefits of existing sample-space detectors with our latent model.

3.1 Generative Process with Hidden Variables

Formulation Following Deng et al. (2022), we introduce a next-word prediction model that also models underlying structures using latent variables ($z \in \mathcal{Z}$), with the generated output as observed variables ($x \in \mathcal{X}$) shown in Figure 4:

$$P(x) = \sum_z P(z)P(x \mid z) \tag{3}$$

Given a single text sequence $\mathbf{x}$ sampled from $x \sim P_x(x)$, we infer the latent sequence $\mathbf{z}$ from $\mathbf{x}$ with

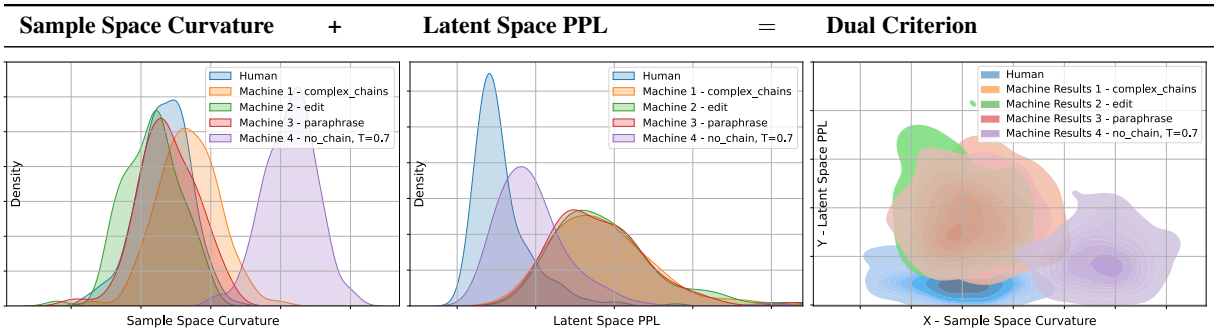

Figure 3: **Left and middle**: kernel density plots of the sample-space curvature and latent space PPL across five test sets in the news domain. These include human-written texts (collected from multiple sources) and machine-generated texts under four different configurations. The plots reveal complementary strengths: 1) the sample-space curvature only effectively distinguishes machine outputs generated from typical settings but fail to identify outputs generated with complex prompts or after paraphrasing/edit attacks; 2) the latent-space PPL excels at distinguishing those non-standard settings. **Right**: considering both criteria leads to the most robust detection performance.

a learned posterior function $P_z(z \mid x)$[2], also called a critic model. We can evaluate the negative log-likelihood (latent NLL) and perplexity (Latent PPL) of the inferred latent variables with:

$$\begin{aligned} \textit{Latent NLL}(\mathbf{x}) &= -\mathbb{E}_{\mathbf{z}\sim P_z(z|x)} \log P_z(\mathbf{z}) \\ \textit{Latent PPL}(\mathbf{x}) &= \exp \frac{1}{m} \textit{Latent NLL}(\mathbf{x}) \end{aligned} \tag{4}$$

where m is the length of the latent sequence, $\mathbf{z}$. When using an external inference model, the process of inferring latent features is expressed as $z^\star = \text{argmax}_z p(z \mid x)$, and the *Latent NLL*$(\mathbf{x})$ becomes $-\log p\left(\text{argmax}_z p(z \mid x)\right)$.

Choice of latent variables We study the impact of the choice of the critic model, $P_z(z \mid x)$. Without losing generality, the latent variables can be anything that captures the high-level underlying structures of a long-form text, such as topic or event transitions. We experiment with five different variables that are relatively easy to obtain, including the part-of-speech tags, nouns or verbs as the approximation of topics, event types, and event triggers, all of which can be obtained using off-the-shelf extraction models (Bird et al., 2009; Peng et al., 2023). We show the results of ***event triggers*** [3], the best performing latent variable, in § 4.4 and report the full results in § 5.1.

Algorithm 1 Dual Criterion Process (Sequential)

Input:
X - List of sample space curv.
Y - List of latent space PPL
Output:
combined - List ▷ requires continuous criteria rather than binary outcomes.
Begin
init combine as **empty list**
for (x, y) **in** X, Y **do**
  if Confidence$(x = \text{machine}) > 95\%$ **then**
    combined.add(max(Y))
  else
    combined.add(y)
return combined
End

Latent-Space Language Model Given a critic model $P_z(z \mid x)$ and a distribution of x, we have to learn $P_z(\mathbf{z})$ to obtain *Latent PPL*. Concretely, we train a reduced-scale transformer on sequences of latent variables inferred only from human-written texts. At the test time, we can then compare the latent PPL from two distributions: human-written and machine-generated.

3.2 Dual Criterion Process

Latent variables contribute to machine-content detection by complementing the strengths of observation-space detectors. We illustrate this by examining the distributions of HWTs and and MGTs under four different configurations in Figure 3. The figure reveals that 1) the sample-space curvature (Mitchell et al., 2023) only distinguishes machine outputs from standard settings (no chain, T=0.7) but fails for those outputs from complex

[2] If $P_z(z \mid x) = \mathbb{1}[z = x]$ or z is the same as x, then *Latent PPL* is the same as text perplexity.

[3] *Event type* and *event trigger* are terms commonly used in Information Extraction (Grishman and Sundheim, 1996; Doddington et al., 2004; Chen et al., 2015; Liu et al., 2020). The **event trigger** is the main word or phrase that most clearly signals the occurrence of an event, while the **event type** refers to the category or classification of the event trigger. For example, in the sentence *'Barry Diller on Wednesday quit as chief of Vivendi Universal Entertainment'*, the event trigger is *'quit'*, and the event type is *'Personnel_End-Position'*.

prompts or altered by paraphrasing and editing attacks. Conversely, 2) the latent-space PPL excels at identifying texts generated under these challenging conditions but is less successful in distinguishing machine outputs under typical settings.

Towards more robust detection, we propose to combine both metrics. Follow existing setups (Mitchell et al., 2023; Bao et al., 2024) which report detection accuracy using AUROC[4], we require a continuous criterion rather than a mere binary classification outcome. Therefore, we consider both metrics in a sequential order, first based on the confidence level of the sample-space detector. The detailed procedure is described in Algorithm 1.

4 Experimental Results

4.1 Dataset

Human Text Following previous setups (Bao et al., 2024; Mitchell et al., 2023), we collect texts from similar domains that cover a variety of LLM use-cases. We use recent movie synopses from Wikipedia to represent creative writing, New York Times and BBC articles to represent news, and the introduction sections of Arxiv papers from three disciplines: economy, quantitative biology, and computer science, to represent academic essays. We crawled the latest human-written texts ourselves and intentionally avoided using common datasets such as Reddit WritingPrompt (Fan et al., 2018), XSum (Narayan et al., 2018) and PubMed (Jin et al., 2019) to avoid data contamination in recent LLMs such as Llama3 and GPT-4.

Machine Text On all above domains, we collect machine outputs from two sources: Llama3 (AI@Meta, 2024) that represents open-source LLMs and GPT-4 (Achiam et al., 2023) that represents proprietary models. We test the following and variations in prompts. No Chain: Directly generate the output given the task instruction and *generating seed* (including the title, first sentence, topic, etc.). We compare sampling with decoding temperature T=0.7 and T=1.0. Simple Chain: First write an outline given the task instruction and *generating seed*, then expand the outline to an complete article. Complex Chain: Identify illogical and vague descriptions and revise the original generation of simple chain. We use a decoding temperature of 1.0 unless otherwise specified.

We also implement popular **attack** methods on texts generated by complex chain to increase the difficulty of detection task: Edit (Shi et al., 2024): Randomly replace 40% adjectives, adverbials and 20% verbs (about 15% in total) with their synonyms. Paraphrase (Sadasivan et al., 2023): Randomly paraphrase 40% sentences while maintaining the overall coherence by keeping the proper nouns and writing style unchanged.

Inferring Events as Latent Variable We extract event types and triggers through OmniEvent's model (Peng et al., 2023) under the MAVEN schema (Wang et al., 2020) for news and movie. We used GPT-4 for few-shot event extraction in academic essays due to the absence of a specialized model. [5]

4.2 Baseline Methods

We compare our method with the state-of-the-art detection system, Fast-DetectGPT (Bao et al., 2024), which utilizes the conditional probability curvature (**Sample Curv.**) using a GPT-Neo-2.7b model (Black et al., 2021). It outperforms the well-known DetectGPT (Mitchell et al., 2023) by 29%. In addition, we compare with the popular token perplexity (**Sample PPL**) on several pretrained LLMs including gpt2-medium, gpt2-large (Radford et al., 2019), GPT-Neo-2.7b, and Llama3-8b (AI@Meta, 2024), and find similar accuracy despite of model sizes. To be consistent with the model used in *Sample Curv.*, we picked GPT-Neo-2.7b.

4.3 Training Configurations

Latent Model Since our latent sequences are much shorter than the observed texts, we train a lightweight transformer *from scratch* on sequences of latent variables inferred solely from HWTs. We began with a gpt2-medium-sized transformer and performed grid search to decrease parameter size by half every iteration. We continued to reduce the model size until a noticeable decrease on a held-out development set was observed. The model configuration is a randomly initialized transformer with 12 heads, 12 layers, and 384 embedding dimensions. We still consider our detector as "zero-shot" as it is not trained on negative samples at all.

There was not much effort needed to tune the hyper-parameters. We trained our latent-space

[4]Area under the True Positive Rate (y-axis) against False Positive Rate (x-axis)

[5]Further details on our data size, source, prompting, attack strategies, and event extraction models are provided in Appendix A.

Source of Machine Output	MOVIE				NEWS				ARXIV			
	Sample PPL	Sample Curv.	Latent PPL	Dual Crit.	Sample PPL	Sample Curv.	Latent PPL	Dual Crit.	Sample PPL	Sample Curv.	Latent PPL	Dual Crit.
1-Complex Chain	0.84 →	0.53 =	0.99 →	0.98	0.88 →	0.73 →	0.97 →	0.96	0.81 ←	0.80 ←	0.78 →	0.79
2-Paraphrase	0.95 →	0.65 ←	0.99 →	0.99	0.97 →	0.53 =	0.97 →	0.97	0.53 →	0.52 =	0.88 →	0.87
3-Edit	0.97 →	0.73 ←	0.99 →	0.99	0.99 →	0.62 ←	0.97 →	0.97	0.80 →	0.73 ←	0.90 →	0.88
4-Simple Chain	0.66 →	0.56 =	0.98 →	0.97	0.70 →	0.86 →	0.95 →	0.96	0.91 ←	0.90 ←	0.73 →	0.80
5-No Chain	0.54 =	0.75 →	0.96 →	0.95	0.88 ←	0.99 →	0.83 →	0.99	0.90 ←	0.95 →	0.71 →	0.86
6-No Chain (T=0.7)	0.98 ←	0.99 →	0.94 →	0.99	0.94 ←	0.99 →	0.80 →	0.99	0.98 ←	0.99 →	0.61 →	0.78
Mixture	0.642	0.646	**0.976**	0.972	0.565	0.750	0.912	**0.969**	0.653	0.734	0.759	**0.855**

Table 2: **First 6 rows**: Relative position (indicated by arrows) and the detection accuracy (measured in AUROC) of individual distributions of MGTs and HWTs. Note that the directions of arrows are more important than the numerical values. An arrow pointing to the right (→), left (←), or equality (=) signifies that the machine distribution is to the right of, to the left of, or close to the human distribution. Neither of the sample-space detectors are robust, as is indicated by a mixture of arrows types. **Last row**: Detection accuracy of a mixture of the all distributions of MGTs described above and HWTs, which better reflects real-world black-box scenarios. We use boldface to denote the best performance and underscore the second best.

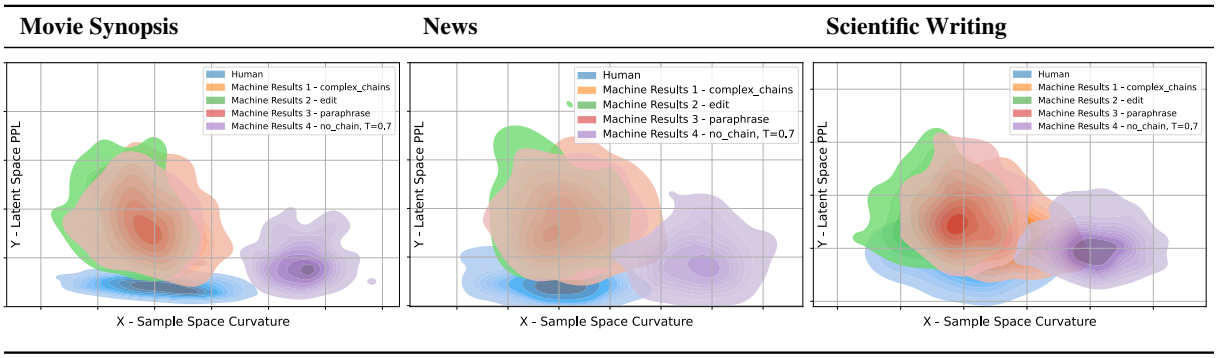

Figure 5: 2D density clouds. For better readability, we only show four sets of machine generated outputs.

model with a learning rate of 1e-4, batch size of 24, and drop-out rate of 0.1. We randomly split 10% from the training set as a validation set. We stopped training when the model reached the maximum number of epochs, which is set to 10. We selected the model checkpoint with the best validation loss and ran that model on our test set.

Domain Adaptation To ensure a fair comparison, we performed domain adaptation (DA) by fine-tuning all sample-space detectors for one epoch on the same dataset of human texts on which our latent model is trained. However, we found that such domain-adaptation resulted in decreased performance at test time, possibly due to over-fitting. Consequently, we used the pre-trained LMs without DA and report their detection accuracy.

4.4 Main Results

We report the detection performances of all compared models in Table 2. We first evaluate their accuracy on six individual sets of MGTs and HWTs (the first six rows in Table 2), and visualize their distributions in Figure 5. Across the three domains of movie, news, and science, neither of the sample-space detectors demonstrate robustness to all generation configurations or attacks, as is reflected by the variability in arrow directions—right, left, and equal. Consequently, when facing a mixture of all six sets of MGTs that better reflects the real-world black-box settings (the last row in Table 2), sample-spaces detectors achieve only 60% to 70% accuracy. On the other hand, our latent-space detector consistently place machine outputs in a separable space (*i.e.,* consistently to the right of human distributions), greatly surpassing the baselines. Our dual criterion process which takes advantage of both the sample-space curvature and *latent PPL* achieves the highest detection accuracy.

We also observe that the latent-space model demonstrates superior performance in narrative domains (*e.g.,* movie and news) compared to scientific domains. This can be attributed to two factors.

Length of Token	Length of Event Trigger	Movie S Curv.	Movie L PPL	News S Curv.	News L PPL	Arxiv S Curv.	Arxiv L PPL
128	16.4	0.567	**0.829**	0.562	**0.842**	**0.605**	0.549
256	28.7	0.579	**0.916**	0.604	**0.887**	**0.652**	0.586
512	55.9	0.618	**0.966**	0.667	**0.909**	**0.729**	0.679
1024	83.3	0.636	**0.976**	0.738	**0.912**	0.734	**0.767**
2048 (max)	84.2	0.646	**0.976**	0.750	**0.912**	0.734	**0.768**

Table 3: Detection accuracy (AUROC) of the strongest sample-space detector (**S Curv.**, (Bao et al., 2024)) and our latent-space detector (**L PPL**) with varying content lengths. Higher accuracy for each domain is highlighted in bold.

First, narratives fundamentally rely on events as their central structural elements (Verhoeven and Stromqvist, 2004; Keven, 2016), whereas scientific writing focuses more on factual information and technical descriptions, which may not align as perfectly with the event-centric nature of our latent representation. **Second**, we find the current event extraction model less reliable on scientific texts. This limitation can lead to error propagation during both training and testing phases. Therefore, we encourage future research to develop specialized methods for extracting alternative discourse structures from academic writings, which may improve the accuracy of detection in scientific domains.

4.5 Impact of Input Sequence Length

The detection accuracy of both our latent-space detector and the token-level baselines is impacted by the length of text. To account for this, we conduct additional experiments controlling for input length by truncating texts to {128, 256, 512, 1024, 2048} tokens and extract event triggers. We compare the performance of the strongest sample-space detector with our latent-space detector and report their test accuracy in Table 3. For both methods, detection accuracy improves with longer text. Notably, despite our approach relying on discourse features, which are sparser than tokens, it demonstrates greater robustness to shorter texts, particularly in the movie and news domains.

5 Further Analysis

5.1 What is the best choice of latent variables?

We report the detection accuracy of five different latent variables: part-of-speech tags, nouns or verbs, event types, and event triggers in Table 4. We find that using parts of speech (which represents the inner-sentence coherence) as the underlying hidden structures leads to decreased accuracy. This also supports the claim that current LLMs already generate locally human-like texts. On discourse structures, only events are indicative of the deviations observed between LLMs and humans. Event types, ranking as the second most effective indicator, provide a richer analysis beyond mere word forms (*e.g.* verbs and nouns) and syntactic functions (*e.g.* parts of speech). However, event types might be too generic and cannot capture the finer semantic differences as well as event triggers.

	MOVIE	NEWS	ARXIV	**Avg**	**Relative**
Sample Curv.	0.65	0.77	0.73	0.72	-
Pos Tag	0.53	0.79	0.65	0.66	- 8%
Verbs	0.80	0.64	**0.78**	0.74	+ 4%
Nouns	0.78	0.51	0.66	0.65	- 9%
Event Type	0.85	0.75	0.73	0.78	+ 9%
Event Trigger	**0.98**	**0.91**	0.77	**0.89**	+ 24%

Table 4: Detection accuracy (measured in AUROC) and relative performance change using models trained on different latent variables. We highlight the best in boldface and second best in underline. For ARXIV, we posit that the higher accuracy with verb sequences is due to the errors of events extracted from scientific writings.

5.2 Event choice v.s. Event transition

A natural follow-up question is *"How much difference comes from event choices versus event transitions?"* To answer this, we build Bag-Of-Words (BOW) models using the event triggers extracted from the test set, and visualize the PCA-reduced features in Figure 6. We compare the detection performance using three methods in Table 5: 1) the principle component of the BOW features (BOW-PC), 2) randomly shuffled event triggers (Sequence Sf.), and 3) ordered event triggers (Sequence).

Event transition is crucial, but the extent depends on task creativity. When event triggers are randomly shuffled or reduced to their principle component, the loss of sequential information impairs the detector's ability to distinguish text origins. This highlights the critical role of event transitions in maintaining an article's coherence and authenticity. However, the amount of decrease

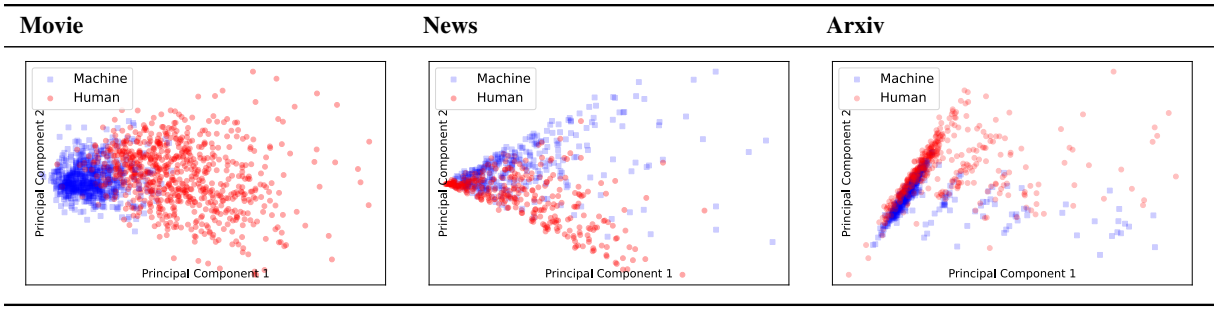

Figure 6: Bag-Of-Words feature of human and machine latent variables, reduced via PCA.

	Movie	News	Arxiv
BOW - PC	0.945	0.731	0.705
Sequence $_{\text{Sf.}}$	0.953	0.801	0.744
Sequence	0.976 (↑3.3%)	0.912 (↑24.8%)	0.768 (↑8.9%)

Table 5: Detection accuracy (measured in AUROC) of two latent methods that only consider the choices of event triggers (BOW-PC and Sequence $_{\text{Sf.}}$) and our best latent model that includes both the choices and transitions (Sequence).

are still domain dependent. In movies, the choice of events itself already plays a decisive role (over 94%). This suggest that LLMs use a distinct list of events triggers from humans in highly creative writing tasks such as movie. Conversely, in less open-ended domains like news and science, the choice of event triggers is less distinct and the transition between event triggers contributes more to authenticity. This indicates that while LLMs have learned the appropriate event triggers, they have not yet mastered the most logical flow of event transitions. To sum up, MGTs are farther away from HWTs on highly creative generation tasks.

5.3 Does explicit structure-aware planning make our detector less effective?

We utilize the differences of latent-structures between MGTs and HWTs. If models are explicitly instructed to elaborate on the underlying structures before auto-regressive generation, would our detection method remain effective? The two prompting methods (*i.e.,* simple chain and complex chain) we employed in § 4.1 are designed to answer this question by integrating a preliminary planning and revision stage before or after text generation.

We emphasize that current models are unable to mimic human-like discourse authenticity even when instructed to plan on these structures. A comparison of line 1, 4, and 5 in Table 2 reveals that despite the addition of a planning and revision stage, models still struggle to replicate the human-like flow in event arrangement. This finding echos previous critiques that current LLMs are poor in mimicking human high-level structures (Deng et al., 2022). One possible reason is that most LLMs are trained to optimize for local coherence and fluency, rather than an overarching, discourse-level logic. For example, the planning mechanism employed by LLMs usually involve skeletal outlines or lists of keypoints that tend to prioritize surface-level coherence instead of the depth in thought. In contrast, human writers often plan their articles with a conscious awareness of theme and plot development that are inherently challenging for current LLMs.

6 Related Work

6.1 MGT Detection

In aspects of ownership and usability, detectors can be roughly divided into *a priori* and *post-hoc* categories. **A priori method** involves proactive involvement of the model's generation process through techniques like watermarking (Ghosal et al., 2023). For example, Kirchenbauer et al. (2023) encourage the sampling of tokens from a pre-determined category (a green list), and this special token distribution can be utilized for detection. Christ et al. (2024) minimizes the distance between the watermarked and original distribution, making the watermark both undetectable and unbiased in expectation. Despite the effectiveness of such techniques, users may still opt for non-watermarked model like GPT-4, underscoring the need for robust *post-hoc* detection methods (Yang et al., 2023). **Post-hoc method** involves fine-tuning classifiers on corpora of positive and negative samples (HWTs and MGT) (Chen et al., 2023; Liu et al., 2019) and zero-shot detection. These former classifiers often struggle with out-of-distribution data (Zhang et al., 2023) and are sensitive to data quality (Liang et al., 2023). The variety of existing LLMs makes it less practical to curate a universal training dataset (Bhat-

tacharjee et al., 2023). The zero-shot detectors perform the task through a statistical approach. As is introduced in § 2, Su et al. (2023); Gehrmann et al. (2019); Mitchell et al. (2023); Bao et al. (2024) employ statistics like probability, entropy and curvature. We focus on zero-shot detection because it is generalizable to diverse domains and do not require access to the source model.

6.2 Attacks to Zero-Shot Detection

Attacks can occur pre-, post-, or during text generation (Wang et al., 2024), which we considered in our experiments (§ 4.1). Pre-generation attacks involve manipulating prompts to produce outputs that are inherently harder to detect, such as adversarial searches to known detectors (Shi et al., 2024). Post-generation attacks replace text segments with lexically or semantically similar alternatives, such as typos, filled mask, synonyms, and rephrased sentences (Shi et al., 2024; Sadasivan et al., 2023). On-generation attacks (Wang et al., 2024) involve decoding with intentional perturbations like typos or emojis, which are later removed to alter the text's statistical distribution, impairing detection performance. Additionally, Zhang et al. (2023) explores how shifts in topic can impact detector efficacy.

6.3 Latent Features for Language Modeling

Current LLMs excel at generating locally fluent sentences, yet they often fail to maintain the long-form coherence, which requires awareness of connecting diverse ideas logically (Lin et al., 2021). Bowman et al. (2016) introduced latent variable models to improve the structural understanding of long texts. Following this, Contrastive Predictive Coding (CPC) (Oord et al., 2018) was proposed to learn unconditioned latent dynamics implicitly, which Wang et al. (2022) further refined with the introduction of Brownian bridge to impose structured, goal-oriented dynamics within the latent space of texts. Novel evaluations for long-form coherence include model criticism based on latent structures such as section labels in Wikipedia (Deng et al., 2022). Sheng et al. (2024) created a coherence assessment metric grounded in Brownian bridge theory (Horne et al., 2007). Owning to the lack of reliable methods for inferring completely unobservable features from texts, we constrain our latent variables to more accessible ones such as events.

7 Conclusion and Discussion

We propose a novel zero-shot detection framework that employs latent features such as sequences of events. Our method leverages the limitations of current LLMs in replicating authentic human-like discourse, despite their ability to generate locally convincing language. Experimental results demonstrate that our detector is highly robust across various real-world generation settings and attacks.

Similar to existing works, our detection methods also rely on certain assumptions about how machine-generated content differs from human-created content. We specifically assumes that text generators struggle to replicate human-like high-level structures. However, we believe that it will take longer for LLMs to catch-up with humans on high-level structures than for them to mimic token-level distributions. Therefore, our method is likely to remain relevant for a longer period than the token-level approaches. Furthermore, we hope the latent-space statistics can serve as a valuable indicator to guide current LLMs in improving high-level content generation.

Acknowledgements

The authors are grateful to Tao Meng for his valuable feedback on the experiments. We also thank the Pluslab members at UCLA and the anonymous reviewers for their insightful comments on the paper. This research is partly supported by National Science Foundation CAREER award #2339766, an Amazon AGI foundation research award, and a Google Research Scholar grant.

Limitations

Our approach involves inferring discourse features, which are more sparse than tokens, hence is restricted to detecting long-form texts. Additionally, the detection accuracy is reliant on the performance of an external inference model, which, in our case, is the event extractor. We find the existing event extraction models are less accurate on scientific texts, which can lead to error propagation during both training and testing phases. We also encourage future research to develop specialized methods for extracting alternative discourse structures from academic writings, which could enhance the accuracy of machine detection in scientific domains.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive domain adaptation for AI-generated text detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 598–610, Nusa Dua, Bali. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. Gpt-sentinel: Distinguishing human and chatgpt generated content. *arXiv preprint arXiv:2305.07969*.

Miranda Christ, Sam Gunn, and Or Zamir. 2024. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1125–1139. PMLR.

Yuntian Deng, Volodymyr Kuleshov, and Alexander Rush. 2022. Model criticism for long-form text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11887–11912, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

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

Soumya Suvra Ghosal, Souradip Chakraborty, Jonas Geiping, Furong Huang, Dinesh Manocha, and Amrit Bedi. 2023. A survey on the possibilities & impossibilities of ai-generated text detection. *Transactions on Machine Learning Research*.

Ralph Grishman and Beth M Sundheim. 1996. Message understanding conference-6: A brief history. In *COLING 1996 volume 1: The 16th international conference on computational linguistics*.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. *arXiv preprint arXiv:2401.12070*.

Jon S Horne, Edward O Garton, Stephen M Krone, and Jesse S Lewis. 2007. Analyzing animal movements using brownian bridges. *Ecology*, 88(9):2354–2363.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in Neural Information Processing Systems*, 36:15077–15095.

Jack Clark Amanda Askell Ariel Herbert-Voss Jeff Wu Alec Radford Gretchen Krueger Jong Wook Kim Sarah Kreps Miles McCain Alex Newhouse Jason Blazakis Kris McGuffie Jasmine Wang Irene Solaiman, Miles Brundage. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Nazim Keven. 2016. Events, narratives and memory. *Synthese*, 193(8):2497–2517.

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

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7).

Chu-Cheng Lin, Aaron Jaech, Xin Li, Matthew R. Gormley, and Jason Eisner. 2021. Limitations of autoregressive models and their alternatives. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5147–5173, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 1641–1651.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Hao Peng, Xiaozhi Wang, Feng Yao, Kaisheng Zeng, Lei Hou, Juanzi Li, Zhiyuan Liu, and Weixing Shen. 2023. The devil is in the details: On the pitfalls of event extraction evaluation. In *Findings of ACL 2023*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

Zhecheng Sheng, Tianhao Zhang, Chen Jiang, and Dongyeop Kang. 2024. Bbscore: A brownian bridge based metric for assessing text coherence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14937–14945.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2024. Red teaming language model detectors with language models. *Transactions of the Association for Computational Linguistics*, 12:174–189.

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

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2024. The science of detecting llm-generated text. *Communications of the ACM*, 67(4):50–59.

Yufei Tian, Tenghao Huang, Miri Liu, Derek Jiang, Alexander Spangher, Muhao Chen, Jonathan May, and Nanyun Peng. 2024. Are large language models capable of generating human-level narratives? *EMNLP*.

Yufei Tian and Nanyun Peng. 2022. Zero-shot sonnet generation with discourse-level planning and aesthetics features. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3587–3597.

Christoforos Vasilatos, Manaar Alam, Talal Rahwan, Yasir Zaki, and Michail Maniatakos. 2023. Howkgpt: Investigating the detection of chatgpt-generated university student homework through context-aware perplexity analysis. *arXiv preprint arXiv:2305.18226*.

Ludo Verhoeven and Sven Stromqvist. 2004. *Relating events in narrative, Volume 2: Typological and contextual perspectives*, volume 2. Taylor & Francis.

Rose E Wang, Esin Durmus, Noah Goodman, and Tatsunori Hashimoto. 2022. Language modeling via stochastic processes. In *International Conference on Learning Representations*.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in*

Natural Language Processing (EMNLP), pages 1652–1671, Online. Association for Computational Linguistics.

Yichen Wang, Shangbin Feng, Abe Bohan Hou, Xiao Pu, Chao Shen, Xiaoming Liu, Yulia Tsvetkov, and Tianxing He. 2024. Stumbling blocks: Stress testing the robustness of machine-generated text detectors under attacks. *arXiv preprint arXiv:2402.11638*.

Zhenyu Xu and Victor S Sheng. 2024. Detecting ai-generated code assignments using perplexity of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23155–23162.

Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022. Re3: Generating longer stories with recursive reprompting and revision. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4393–4479.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. 2023. A survey on detection of llms-generated content. *arXiv preprint arXiv:2310.15654*.

Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Yi-Fan Zhang, Zhang Zhang, Liang Wang, and Rong Jin. 2023. Assaying on the robustness of zero-shot machine-generated text detectors. *arXiv preprint arXiv:2312.12918*.

A Datasets

We describe how we created human and machine text dataset in more detail. The statistics summary of dataset is shown in Table 6.

A.1 Collecting Human Texts

Considering any existing text from internet can be the training data of current LLMs, we crawled the latest texts to avoid LLMs memorizing them when performing detection.

For **Movie**, we crawl the recent English-language films category on Wikipedia[6]. To increase the quality of synopses, we remove those with fewer than 25 sentences. To minimize the risk of model memorization, we filter out well-known movies using the lengths of Wikipedia pages as an approximate indicator of popularity.

For **News** articles, we collected all news articles from the main page of The New York Times[7] published from 2024-04-09 to 2024-05-18 covering mainly politics, business and editorials news. The test data are a mix of New York Times from the same source and BBC[8], the latter of which we consider as out-of-distribution to increase the task difficulty.

For **Arxiv** paper, we downloaded 419 economy, 666 quantitive biology and 782 computer science published from 2023-06 to 2024-04 using its official API. We then extracted the introduction section in plain text from TeX source code of each paper using GPT-3.5.

A.2 Collecting Machine Texts

All human and machine outputs have roughly the same length. For each human text in test set, we generate a paired machine-generated text.

Variations in Prompts For whole text generation, we use pure sampling at temperature T = 1.0 as default. To further avoid data contamination, we first let the model to rephrase the titles and initial settings, termed as *generating seed*, by altering all the unique identifiers such as proper nouns. Then, we use the three prompting strategies described in § 4.1 to collect machine generated texts from the *generating seeds*. More concretely, No Chain directly complete the whole texts.Simple Chain mimics the human-like plan-write process, by first generating an structured outlines and then expanding it to the whole texts. Complex Chain add revision steps on top of Simple Chain, to add more details in the outline and fix any illogical and vague descriptions in original output. The complex chain prompt used to generate scientific essays is shown in Figure 7.

Attack For both paraphrasing and edit attacks, we introduce adversarial searches to known detectors (Shi et al., 2024). Overall, our approach first generates multiple substitutions for all candidate segments that can be replaced by substitutions without changing the meaning drastically. Then we randomly sample substitutions with certain probability to produce candidates. Finally, GPT-2-XL is used to calculate and select the text with the highest perplexity to gain the maximum attack efficiency at a fixed replacement ratio. For **Paraphrase**, the segment is sentence level. We generate 2 to 5 substitutions for each sentence while keeping every proper noun and overall writing styles. Then, we generate candidates through replacing 40% original sentences. For **Edit**, the segment is word level. We generate 2 to 5 context-based synonyms for each adjective, adverbial and verb as replacing them would not affect the semantics severely. Then, we generate candidates through replacing 40% adjectives, adverbials and 20% verbs (about 15% of total words). The prompt used for attacks is shown in Figure 8.

A.3 Event annotation

For news and movie, we employed the off-the-shelf T5 model from OmniEvent (Peng et al., 2023), which is trained on multiple dataset including ACE05[9], MAVEN (Wang et al., 2020), etc. We use this model under MAVEN schema, which defines 168 event types that cover various general scenarios useful for our analysis. Owning to the fact that there are no event extraction model specialized in scientific writing domain, we prompt GPT-4 to extract, as is shown in Figure 9. Additionally, we employ SpaCy's lemmatization pipeline to standardize the form of all event triggers.

[6] https://en.m.wikipedia.org/wiki/Category:2020s_English-language_films

[7] www.nytimes.com/section/us

[8] https://www.bbc.com/news/us-canada

[9] https://catalog.ldc.upenn.edu/LDC2006T06

Main Tasks	Source	Event Extraction Model	Length (Words)	Length (Events)	Train Size	Test Size
Creative writing	Movie synopsis (from Wikipedia)	OmniEvent (Peng et al., 2023)	673	86	2,178	250
News	NYT and BBC	OmniEvent	1024 (truncated)	92	2,813	218
Academic essay	Arxiv (introduction of computer science, economy, and quantitative bio)	Few-Shot GPT-4	796	66	2,680	287

Table 6: The statistics of our human data. Note that for news articles, word length is truncated to max sequence length of GPT-2.

```
<-- Create Consice Outline -->
User: Create a simple outline structure for writing the introduction section of an academic paper based on the given title and first sentence. Each line is a key component and its explanation. The paper domain is "{domain}", title is "{rephrased_title}" and first sentence is "{rephrased_first_sentence}".

Assistant: "{concise_outline}"

<-- Expand Outline -->
User: Based on the given simple outline of writing the introduction section of an academic paper, expand on the key points outlined, providing bullet points of clear, well-developed arguments, data, context, etc. that strengthen the introduction. The paper domain is "{domain}", title is "{rephrased_title}" and the outline is "{concise_outline}".

Assistant: "{expanded_outline}"

<-- Draft Paper -->
User: Build upon the given bullet points to write a comprehensive and logically structured introduction that frames the paper's arguments and significance. Your output should be the introduction section of an academic paper generated in about 50 sentences. The paper domain is "{domain}", title is "{rephrased_title}" and the outline is "{expanded_outline}".

Assistant: "{paper_draft}"

<-- Refine Paper -->
User: Based on the given outline, reexamine the flow of the draft introduction to ensure that it logically progresses from general context to specific research questions, effectively setting up the research framework. Strengthen transitions between ideas, ensure coherence in the presentation of arguments, and align the structure with academic standards for introductions. Your output should be the introduction section of an academic paper generated in about 30 sentences. The paper domain is "{domain}", title is "{rephrased_title}", the given outline is "{expanded_outline}" and the draft is "{paper_draft}".

Assistant: "{refined_draft}"
```

Figure 7: Complex chain prompt used for generating scientific essays.

```
<-- Edit Attack -->
User: Given the sentence and words within, for each of words, given two to five substitution words that do not change the meaning of the sentence. Only generate substutions when a word is general but not proper word. Return each general word and its substitutions in one line, in the format of 'word: substitution 1, substitution 2, ...'. sentence: "{sentence}"; words: "{words}"

Assistant: {Word substituions}

<-- Paraphrase Attack -->
User: Please paraphrase the highlighted sentence (wrapped by '**') in the below text in 2 - 5 ways. You should keep all proper words and style of the original text in your paraphrased sentences. Your should directly output paraphrase splitted by linebreak without '**'.\n\nText: "{text}"

Assistant: {Sentence substitutions}
```

Figure 8: Prompt for edit and paraphrase attack.

```
User:
Task: For each sentence in input, extract all the major event triggers. Your output should only be a valid JSON
      string that is a list of dictionary. Each dictionary contains two fileds: 'sentence' and 'triggers'.

Examples: "{examples}"

Now extract all major event triggers in the following input: "{input_sentence}"

Assistant: { extracted_events }
```

Figure 9: Prompt for event annotation using GPT-4.