IPAD: Inverse Prompt for AI Detection – A Reliable and Explainable LLM-Generated Essay Detector

Zheng Chen[1]* , **Yushi Feng**[2]* , **Yue Deng**[1] , **Changyang He**[3] , **Hongxi Pu**[4],**Bo Li**[1]
[1] Computer Science and Engineering, Hong Kong University of Science and Technology
[2] School of Computing and Data Science, The University of Hong Kong
[3] Max Planck Institute for Security and Privacy
[4] Computer Science, The University of Michigan
zchenin@connect.ust.hk, fengys@connect.hku.hk, ydengbi@connect.ust.hk,
changyang.he@mpi-sp.org, hongxi@umich.edu, bli@cse.ust.hk

Abstract

Large Language Models (LLMs) have attained human-level fluency in text generation, which complicates the distinguishing between human-written and LLM-generated texts. This increases the risk of misuse and highlights the need for reliable detectors. Yet, existing detectors exhibit poor robustness on out-of-distribution (OOD) data and attacked data, which is critical for real-world scenarios. Also, they struggle to provide interpretable evidence to support their decisions, thus undermining the reliability. In light of these challenges, we propose **IPAD (Inverse Prompt for AI Detection)**, a novel framework consisting of a **Prompt Inverter** that identifies predicted prompts that could have generated the input text, and two **Distinguishers** that examines the probability that the input texts align with the predicted prompts. Empirical evaluations demonstrate that IPAD outperforms the strongest baselines by 9.05% (Average Recall) on in-distribution data, 12.93% (AUROC) on out-of-distribution (OOD) data, and 5.48% (AUROC) on attacked data. IPAD also performs robust on structured datasets. Furthermore, an interpretability assessment is conducted to illustrate that IPAD enhances the AI detection trustworthiness by allowing users to directly examine the decision-making evidence, which provides interpretable support for its state-of-the-art detection results.

1 Introduction

Large Language Models (LLMs), characterized by their massive scale and extensive training data (Chen et al., 2024), have achieved significant advances in natural language processing (NLP) (Ouyang et al., 2022; Veselovsky et al., 2023; Wu et al., 2025). However, with the advanced capabilities of LLMs, they are subject to frequent misused in various domains, including academic fraud, the creation of deceptive material, and the generation of fabricated information (Ji et al., 2023; Pagnoni et al., 2022; Mirsky et al., 2023), which underscores the critical need to distinguish between human-written text (HWT) and LLM-generated text (LGT) (Pagnoni et al., 2022; Yu et al., 2025; Kirchenbauer et al., 2023).

However, due to their sophisticated functionality, LLMs pose significant challenges in the robustness of current AI detection systems (Wu et al., 2025). The existing detection systems, including commercial ones, frequently misclassify texts as HWT (Price and Sakellarios, 2023; Walters, 2023) and generate inconsistent results when analyzing the same text using different detectors (Chaka, 2023; Weber-Wulff et al., 2023). Studies show false positive rates reaching up to 50% and false negative rates as high as 100% in different tools (Weber-Wulff et al., 2023) when dealing with out-of-distribution (OOD) datasets.

Another critical issue with the existing AI detection systems is their lack of verifiable evidence (Halaweh and Refae, 2024), as these tools typically provide only simple outputs like *"likely written by AI"* or percentage-based predictions (Weber-Wulff et al., 2023). The lack of evidence prevents users from defending themselves against false accusations (Chaka, 2023) and hinders organizations from making judgments based solely on the detection results without convincing evidences (Weber-Wulff et al., 2023). This problem is particularly troublesome not only because the low accuracy of such systems as mentioned before, but also due to the consequent inadequate response to LLM misuse, which can lead to significant societal harm (Stokel-Walker and Van Noorden, 2023; Porsdam Mann et al., 2023; Shevlane et al., 2023; Wu et al., 2025). These limitations highlight the pressing need for more reliable, explainable and robust detectors.

In this paper, we propose **IPAD** (Inverse Prompt for AI Detection), a novel and interpretable frame-

*These authors contributed equally to this work.

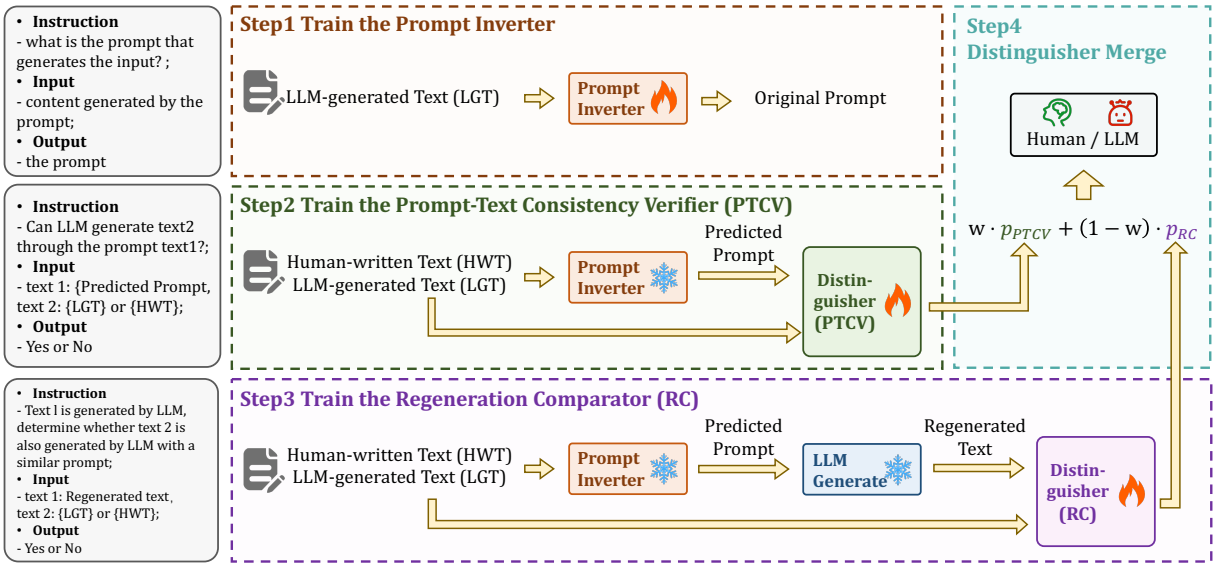

Figure 1: The overall workflow of our proposed IPAD framework

work for detecting AI-generated text. As illustrated in Figure 1, IPAD consists of two main components: a **Prompt Inverter**, which reconstructs the underlying prompts from input texts, and two **Distinguishers**—the **Prompt-Text Consistency Verifier (PTCV)**, which measures the alignment between the predicted prompt and input text, and the **Regeneration Comparator (RC)**, which compares the input with the corresponding regenerated text for consistency. By explicitly modeling the reasoning path from prompt inversion to final classification, IPAD introduces a paradigm shift in AI-generated content detection, significantly enhancing both detection robustness and user interpretability.

Empirical results show that IPAD outperforms state-of-the-art baselines by 9.05% in Average Recall on in-distribution datasets, 12.93% in AUROC on out-of-distribution (OOD) datasets, and 5.48% in AUROC under adversarial attacks. IPAD also generalizes well to structured data. A user study further reveals that IPAD improves trust and usability in detection tasks by presenting concrete decision evidence, including predicted prompts and regenerated texts. Code is anonymously available [1].

Our contributions can be summarized as follows:

- We introduce a novel fine-tuned inverse-prompt-based detection framework that integrates prompt reconstruction and dual consistency evaluation.
- We achieve superior detection performance on in-distribution, OOD, adversarially attacked, and prompt-structured datasets.
- We demonstrate through an interpretability assessment that IPAD improves human trust and interpretability in AI text detection.

2 Methodology

2.1 Preliminaries

Modules. Our method comprises a **Prompt Inverter** f_{inv}, and two Distinguishers, namely the **Prompt-Text Consistency Verifier (PTCV)** f_{PTCV} and the **Regeneration Comparator (RC)** f_{RC}. Given an input text T, the task is to determine whether it is human-written (HWT) or generated by an LLM (LGT). We denote by $\mathcal{D}_{\text{PI}}$ the training set for f_{inv}, consisting of pairs (T, P) where T is an LLM-generated text and P is its original prompt. The two distinguishers are trained using disjoint datasets: $\mathcal{D}_{\text{LGT}}$ contains LLM-generated samples and $\mathcal{D}_{\text{HWT}}$ contains human-written ones. All components are fine-tuned using Microsoft's Phi3-medium-128k-instruct model.[2].

Softmax-Based probability for Binary Classification in LLM. To estimate the fine-tuned model's binary classification probability (i.e., the probability of predicting "yes" or "no"), we follow the logit-based estimation approach (Yoshikawa and Okazaki, 2023). Given the model input x, and the

[1] https://anonymous.4open.science/r/IPAD-Inver-Prompt-for-AI-Detection–65B6/

[2] https://huggingface.co/microsoft/Phi-3-medium-128k-instruct

output logits z, the model's probability assigned to $\hat{y}$ is computed through the softmax function σ :

$$\text{Confidence}_{\text{yes}} = P(\hat{y} = \text{“yes”} \mid x) = \sigma(z)_{\text{yes}},$$

$$\text{Confidence}_{\text{no}} = P(\hat{y} = \text{“no”} \mid x) = \sigma(z)_{\text{no}}$$

Since the fine-tuned model will only output“yes” or “no”, we further calculate the probability for this binary classification as:

$$\text{Probability}_{\text{yes}} = \frac{\text{Confidence}_{\text{yes}}}{\text{Confidence}_{\text{yes}} + \text{Confidence}_{\text{no}}},$$

$$\text{Probability}_{\text{no}} = 1 - \text{Probability}_{\text{yes}}$$

2.2 Workflow

Our framework follows a multi-stage fine-tuning pipeline with the following four steps, as illustrated in Figure 1. The details of the datasets for fine-tuning is illustrated in Appendix A.

Step 1: Training Prompt Inverter. We first fine-tune a model f_{inv} on dataset $\mathcal{D}_{\text{PI}}$, with the data structure shown in Figure 1. For any input text T, f_{inv} predicts the most likely prompt P that could have generated it, i.e. $P = f_{\text{inv}}(T)$. The resulting Prompt Inverter is then frozen and reused in the following downstream steps.

Step 2: Training the Prompt-Text Consistency Verifier (PTCV). Given the predicted prompt P in step 1, and the input text $T \in \{\text{HWT}, \text{LGT}\}$, the verifier f_{PTCV} is trained to predict whether the text T could plausibly be generated by an LLM using the prompt P. The fine-tuning datasets $\mathcal{D}_{\text{LGT}}$ and $\mathcal{D}_{\text{HWT}}$ share the same structure, with output labels "yes" for $\mathcal{D}_{\text{LGT}}$ and "no" for $\mathcal{D}_{\text{HWT}}$, as shown in the Figure 1.

After fine-tuning this module, we applied it to the validation set and computed the probability score $p_{\text{PTCV}} = f_{\text{PTCV}}(T, P)$, where the confidence value was estimated using the softmax-based method described in Section 2.1.

Step 3: Training the Regeneration Comparator (RC). With the same predicted prompt P in step 1, we use an LLM to generate a regenerated text $T' \leftarrow LLM(P)$. By default, the LLM we use is `gpt-3.5-turbo`. Then, the comparator f_{RC} is trained to assess whether T and T' can be generated by LLM with a similar prompt. This step uses the same dataset as in Step 2, but applies a different structural formatting, as shown in Figure 1.

After fine-tuning this module, we also applied it to the validation set and computed the probability score $p_{\text{RC}} = f_{\text{RC}}(T, P)$.

Step 4: Distinguisher Merge. To determine the final classification, we combine the two probability scores, p_{PTCV} and p_{RC}, obtained from Step 2 and Step 3 on the validation set. Specifically, we compute a weighted ensemble as $\hat{p} = w \cdot p_{\text{PTCV}} + (1 - w) \cdot p_{\text{RC}}$, and assign the prediction $\hat{Y} = \text{LGT}$ if $\hat{p} > \tau$, or $\hat{Y} = \text{HWT}$ otherwise. The weight $w \in [0, 1]$ and the threshold $\tau \in [0, 1]$ are treated as hyperparameters and selected via grid search on the validation set. The selected values were $w = 0.45$ and $\tau = 0.54$.

Inference. We perform inference on unseen input texts T by sequentially applying the trained modules. Given an input text T, we first use the prompt inverter f_{inv} to recover the most plausible prompt P. The prompt is then used to regenerate a candidate text T' via the an LLM. Next, we compute two probability scores: p_{PTCV}, indicating whether T is consistent with P, and p_{RC}, assessing the likelihood that T and T' originate from the same prompt. These scores are fused into a final decision score $\hat{p}$ using the gird-searched weight w, and the predicted label is determined by comparing $\hat{p}$ against the threshold τ. The complete algorithm can be found in Appendix B.

The inference procedure of the IPAD framework consists of three calls through a light-weight open-sourced LLM `phi-3-medium-128k-instruct`. `Phi-3` is a decoder-only Transformer, whithin which, the self-attention complexity per layer is $\mathcal{O}(n^2 \cdot d)$, where n is the sequence length and d is the hidden dimension (Vaswani et al., 2017). The additional api call to `gpt-3.5-turbo` for regenerating texts introduces fixed latency but no local computation cost. Therefore, the overall computational cost is bounded by $\mathcal{O}(3 \cdot L \cdot n^2 \cdot d + \texttt{OpenAI}_{\text{api}})$, where $L = 32$ is the number of layers in `phi-3` (Abdin et al., 2024), which is relatively small. All three `phi-3` calls can be deployed in an Nvidia V100 GPU as the minimum requirement. This demonstrates that IPAD is not computationally expensive and can be deployed with relatively modest hardware requirements.

2.3 Training

The supervised fine-tuning (Wei et al., 2022) process is performed on a Microsoft's open model, `phi3-medium-128k-instruct`, and we use low-rank adaptation (LoRA) method (Hu et al., 2022)

on the `LLaMA-Factory` framework[3] (Zheng et al., 2024). We train it using six A800 GPUs for 20 hours for **Prompt Inverter**, 7 hours for **PTCV**, and 9 hours for **RC**.

3 Experiments

We investigate the following questions through our experiments:

- Assess the robustness of IPAD, which includes using various LLMs as generators, comparing IPAD with other detectors, and evaluating on out-of-distribution (OOD), attacked datasets, and prompt-structured datasets.
- Independently analyze the necessity and effectiveness of the **Prompt Inverter**, the **PTCV**, and the **RC**.
- Explore the user-friendliness of IPAD through an interpretability assessment.

3.1 Robustness of IPAD

3.1.1 Evaluation Baselines and Metrics

The in-distribution experiments refer to the testing results presented in (Koike et al., 2024), where the data aligns with the training data used for the IPAD, thereby serving as our baseline. This baseline assesses how the RoBERTa classifiers (base and large) (Park et al., 2021), the HC3 detector (Guo et al., 2023), and the OUTFOX detector (Koike et al., 2024) perform on standard data as well as under DIPPER (Alkanhel et al., 2023) and OUTFOX attacks.

The OOD experiments refer to the DetectRL baseline (Wu et al., 2024), which is a comprehensive benchmark, which includes four datasets: (1) academic abstracts from the arXiv Archive (covering the years 2002 to 2017)[4], (2) news articles from the XSum dataset (Narayan et al., 2018), (3) creative stories from Writing Prompts (Fan et al., 2018), and (4) social reviews from Yelp Reviews (Zhang et al., 2015). It also employs three attack methods to simulate complex real-world detection scenarios, which include (1) the prompt attacks, (2) paraphrase attacks, and (3) perturbation attacks (Wu et al., 2024). DetectRL evaluates three classifiers on the OOD dataset: DetectLLM (LRR) (Su et al., 2023), Fast-DetectGPT (Bao et al., 2023), RoBERTa Classifier (Base). We included two more strong classifiers in our evaluation DetectLLM (NPR) (Su et al., 2023) and Binoculars (Hans et al., 2024). All the testing sets have 1,000 samples in our experiments.

We further evaluate its performance on OOD datasets with **structured prompts**. The LongWriter dataset (Bai et al., 2025), featuring an average prompt length of 1,501 tokens, reflects IPAD's capability to handle long-form prompts. The Code-Feedback[5] and Math datasets (Hendrycks et al., 2021) contain highly structured prompts, in contrast to typical essay-like writing. We compare IPAD with baseline detectors from DetectRL to assess its relative performance under these challenging conditions.

The **Area Under Receiver Operating Characteristic curve (AUROC)** is widely used for assessing detection method (Mitchell et al., 2023). Since our models predict binary labels, we follow the *Wilcoxon-Mann-Whitney* statistic (Calders and Jaroszewicz, 2007), and the formula is shown in Appendix C. The **AvgRec** is the average of **HumanRec** and **MachineRec**, which refers to the recall of the Human-written texts and the LLM-generated texts (Li et al., 2024).

3.1.2 Robustness across different LLMs

As shown in Table 1, IPAD achieves consistently strong performance across all combinations of original generators and re-generators, which shows its robustness to diverse LLM as generators. The best results are generally observed when the original generator and the re-generator are the same, while the `gpt-3.5-turbo` serves as an effective universal re-generator: it performs well even when the original generator differs. In real-world applications where the identity of the original generator is unknown, using `gpt-3.5-turbo` as a fixed re-generator provides a practical and reliable solution.

3.1.3 Comparison of IPAD with other detectors in and out of distribution

In Distribution. For the in-distribution data, as shown in Figure 2, the baseline detectors like RoBERTa, HC3, and OUTFOX perform well on standard data but degrade significantly under DIPPER and OUTFOX attacks. In contrast, IPAD maintains high accuracy across all scenarios, which surpasses the strongest baseline **9.05%** in AvgRec.

[3] https://huggingface.co/papers/2403.13372

[4] http://kaggle.com/datasets/spsayakpaul/arxiv-paper-abstracts/data

[5] https://huggingface.co/datasets/m-a-p/Code-Feedback

Original Generator	Re-Generator	HumanRec	MachineRec	AvgRec	AUROC
gpt-3.5-turbo	gpt-3.5-turbo	98.50	100	99.25	100
gpt-4	gpt-4	98.70	100	99.35	100
	gpt-3.5-turbo	96.10	100	98.05	99.96
Qwen-turbo	Qwen-turbo	98.60	99.80	99.20	99.96
	gpt-3.5-turbo	98.40	99.50	98.95	99.86
LLaMA-3-70B	LLaMA-3-70B	98.70	100	99.35	100
	gpt-3.5-turbo	98.60	100	99.30	100

Table 1: Detection Accuracy (HumanRec, MachineRec, AvgRec, and AUROC %) of IPAD across Various LLMs on In-Distribution Data

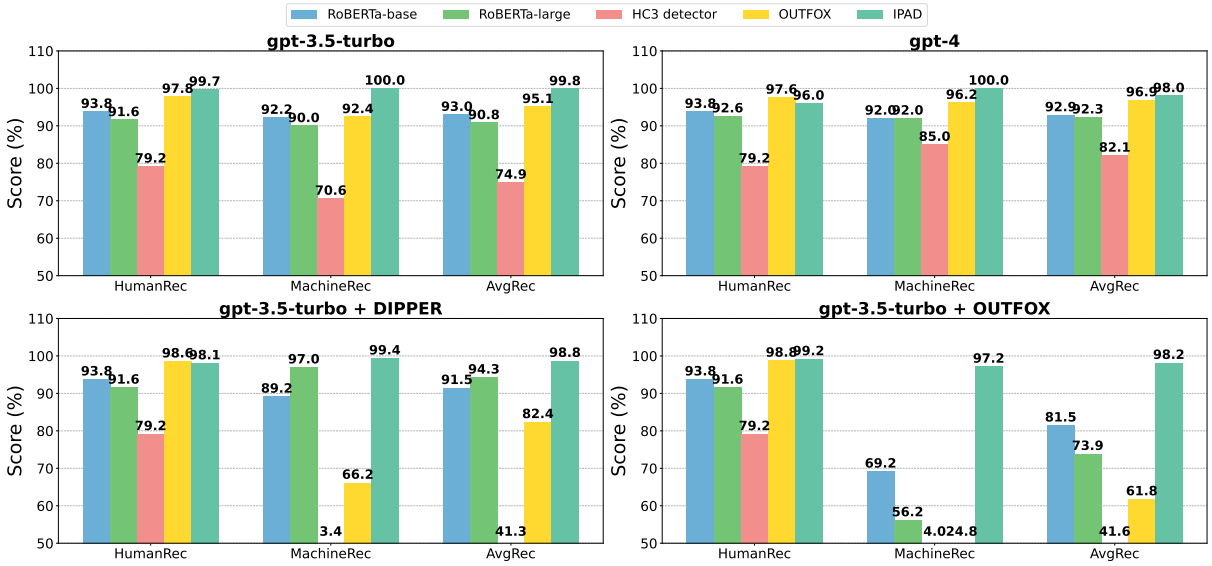

Figure 2: The In-distribution data performance of IPAD and the baseline detectors. Since (Koike et al., 2024) only presents the AvgRec data for the baselines, we also calculate AvgRec data for IPAD to compare.

Out of Distribution. Table 2 reports detection accuracy across four benchmark datasets, which shows that IPAD significantly outperforms prior baselines. Table 3 further evaluates robustness under three attack types, where IPAD again demonstrates superior resilience. Compared to the strongest baseline, IPAD achieves a **12.93%** relative improvement on standard datasets in AUROC and a **5.48%** improvement on attack datasets.

Structured Prompts. The results are shown in Table 4, while these datasets lack HWT references and are thus only evaluated using MachineRec, the strong scores suggest that IPAD maintains robustness even on structured diverse inputs, with an improvement of 9.87% against the strongest baseline in MachineRec.

3.2 Necessity and Effectiveness of the Prompt Inverter, PTCV, and RC

3.2.1 Necessity

To prove that it is necessary to fine-tune on IPAD with IPAD with **PTCV** and **RC**, we conducted ablation study to use the same finetune method on only **input texts** and only **predicted prompts**, with the finetune data format shown in Appendix D. We only experimented on **Prompt Inverter + PTCV** and **Prompt Inverter + RC** to compare with the three-moduled IPAD.

Based on the ablation study results as shown in Figure 3, fine-tuning only on input texts or only on predicted prompts performs poorly across all datasets in AUROC scores. While using **Prompt Inverter + PTCV** or **Prompt Inverter + RC** individually significantly improves performance, neither approach consistently excels across both HWT-style and LGT-style generations. In contrast, the full IPAD framework achieves consistently high performance across all settings, which demonstrates the necessity of the **Prompt Inverter**, **PTCV**, and **RC** modules.

3.2.2 Effectiveness

Prompt Inverter. We use DPIC (Yu et al., 2024) and PE (Zhang et al., 2024b) as baseline methods for prompt extraction. DPIC employs a zero-

Method	Arxiv	XSum	Writing	Review	Average
DetectLLM (LRR)	48.17	48.41	58.70	58.21	53.37
DetectLLM (NPR)	53.85	34.59	54.96	50.09	48.37
Binoculars	84.03	77.39	94.38	90.00	86.45
Fast-DetectGPT	42.00	45.72	51.13	54.55	48.35
Rob-Base	81.06	76.81	86.29	87.84	83.00
IPAD Merge	**100**	**99.85**	**99.40**	**98.25**	**99.38**

Table 2: Detection Accuracy (AUROC %) on four diverse OOD datasets

Method	Prompt Attack	Paraphrase Attack	Perturbation Attack	Average
DetectLLM (LRR)	54.97	49.23	53.62	52.61
DetectLLM (NPR)	77.15	56.94	6.78	46.96
Binoculars	93.45	88.34	76.89	86.23
Fast-DetectGPT	43.89	41.15	44.38	43.14
Rob-Base	92.81	90.02	92.12	91.65
IPAD	**97.30**	**96.00**	**98.10**	**97.13**

Table 3: Detection Accuracy (AUROC %) on three attacked OOD datasets

shot approach using the prompt states in Appendix E, while PE uses adversarial attacks to recover system prompts. In our evaluation, we tested 1000 LGT and 1000 HWT samples. We use only in-distribution data for testing since only these datasets include original prompts. The metrics are all tested on comparing the similarity of the original prompts and the predicted prompts. The results shown in Table 5 illustrate that IPAD consistently outperforms both DPIC and PE across all four metrics (BartScore (Yuan et al., 2021), Sentence-Bert Cosine Similarity (Reimers and Gurevych, 2019), BLEU (Papineni et al., 2002), and ROUGE-1 (Lin, 2004)), which highlight the effectiveness of the IPAD **Prompt Inverter**.

PTCV and RC. We conducted a comparison study using the frozen Prompt Inverter but different distinguishing methods. The first and second methods employed `Sentence-Bert` (Reimers and Gurevych, 2019) and `Bart-large-cnn` (Yuan et al., 2021) to compute the similarity score between the input texts and the regenerated texts. We selected thresholds that maximized AvgRec, which were 0.67 for Sentence-Bert and -2.52 for Bart-large-cnn. The classification rule is that the texts with scores greater than the threshold will be classified as LGT, while the texts with scores less than or equal to the threshold will be classified as HWT. The third method is to directly prompt ChatGPT in Appendix D, which mimic the fine-tuning process of **PTCV** and **RC**. The final results shown in Table 6 demonstrate that the other distinguishing methods performed worse than IPAD, highlighting the superior effectiveness of the IPAD **Distinguishers**.

Compare with DPIC. DPIC first uses a zero-shot prompt inverter to generate prompts, then applies a Siamese encoder and classifier to measure similarity between the embeddings of the original and regenerated texts. However, the classifier's reliance on embedding similarity is ambiguous, as similar texts may stem from different prompts. IPAD addresses this by fine-tuning directly on raw texts and reformulating the task as a logical reasoning problem as shown in the instructions of **PTCV** and **RC**. Our trained **Prompt Inverter** outperforms DPIC's generic zero-shot method as shown in Table 5, and IPAD also achieves better performance than DPIC overall, as results shown in Appendix F.

3.3 Interpretability Assessment of IPAD

To assess the explainability improvement of IPAD, we designed an interpretability assessment with ten participants evaluating one HWT and one LGT article. We used IPAD version 2 due to its superior OOD performance and attack resistance. Participants compared three online detection platforms[6][7][8] with IPAD's process (which displayed input texts, predicted prompts, regenerated texts, and final judgments). After evaluation, participants rated IPAD on four key explainability dimensions. Transparency received strong ratings (40%:5, 60%:4), with participants appreciating the visibility of inter-

[6]https://www.scribbr.com/ai-detector/
[7]https://quillbot.com/ai-content-detector
[8]https://app.gptzero.me/

Method	LongWriter	Code-Feedback	Math	Average
DetectLLM (LRR)	32.1	29.0	30.2	30.43
DetectLLM (NPR)	41.2	45.9	56.0	47.7
Binoculars	82.1	84.6	89.4	85.4
Fast-DetectGPT	12.0	11.1	15.1	12.7
Rob-Base	81.5	89.2	82.1	84.3
IPAD	**97.5**	**92.7**	**95.6**	**95.27**

Table 4: Detection Accuracy (MachineRec %) on three structured OOD datasets

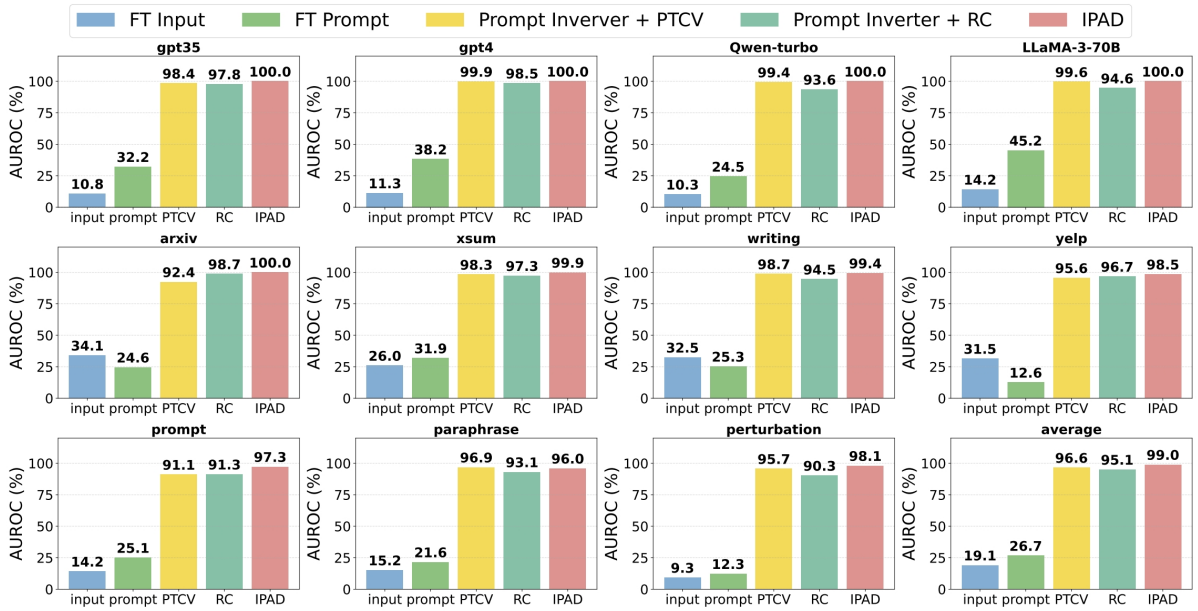

Figure 3: Ablation study. Evaluating **Fine-tune only on Input**, **Fine-tune only on Prompt**, **Prompt Inverter + PTCV**, **Prompt Inverter + RC**, and **IPAD** on In-distribution datasets, standard OOD datasets, and attacked OOD datasets.

mediate processes. Trust scores were more varied (10%:3, 70%:4, 20%:5), but IPAD was generally considered more convincing than single-score detectors. Satisfaction was mixed (30%:3, 30%:4, 40%:5), with participants acknowledging better detection but raising concerns about energy efficiency since IPAD runs three LLMs. Debugging received unanimous 5s, as participants could easily analyze the predicted prompt and regenerated text to verify the decision-making process. If needed, users could refine the generated content by adjusting instructions, such as specifying a word count, making IPAD a more effective and user-friendly tool compared to black-box detectors. We further analyzed the different linguistic features of HWT prompts and LGT prompts as illustrated in Appendix G.

4 Related Work

4.1 AI detectors Methods and challenges

AI text detection methods can be broadly categorized into four approaches (Wu et al., 2025): watermarking, statistics-based methods, neural-based methods, and human-assisted methods.

Watermarking technology inserts specific patterns into training datasets (Shevlane et al., 2023; Gu et al., 2022) or manipulates the model output during inference to embed a watermark (Lucas and Havens, 2023). However, watermarking needs to access the LLM deployment and can face attacks, such as identifying and erasing the watermark (Hou et al., 2024). **Statistics-based methods** analyze inherent textual features to identify language patterns (Kalinichenko et al., 2003; Hamed and Wu, 2023), but their effectiveness depends on corpus size and model diversity (Wu et al., 2025). Some other statistical methods use n-gram probability di-

Metric	LGT			HWT		
	DPIC	PE	IPAD	DPIC	PE	IPAD
Bart-large-cnn	-2.12	-2.23	**-1.84**	-2.47	-2.39	**-2.22**
Sentence-Bert	0.46	0.58	**0.69**	0.42	0.53	**0.57**
BLEU	5.61E-05	3.21E-04	**0.24**	8.75E-06	2.56E-08	**0.13**
ROUGE-1	0.04	0.25	**0.51**	0.06	0.13	**0.39**

Table 5: Comparison of prompt inverters on the similarities of the original prompts and the predicted prompts on LGT and HWT.

Distinguish Method	HumanRec	MachineRec	AvgRec
Sentence-Bert (Thr. 0.67)	61.20	95.20	78.20
Bart-large-cnn (Thr. -2.52)	42.60	97.20	69.90
Prompt to ChatGPT	33.20	64.50	48.85
IPAD	**98.50**	**100.00**	**99.25**

Table 6: Comparison of distinguishers on HumanRec, MachineRec, and AvgRec (%).

vergence (Yang et al., 2024b) or similarity between original and revised texts (Mao et al., 2024; Zhu et al., 2023) while still face robustness challenges under adversarial attacks (Wu et al., 2025). **Neural-based methods** such as RoBERTa (Liu et al., 2020), Bert (Devlin et al., 2019), and XLNet (Yang et al., 2019) have been robust in domain-specific tasks. Adversarial learning techniques are increasingly being used (Yang et al., 2024a) to increase effectiveness in attacked datasets.

In addition to automated methods, human involvement plays a key role in detecting AI-generated text (Wu et al., 2025). **Human-assisted detection** leverages human intuition and expertise to identify inconsistencies such as semantic errors and logical flaws that may not be easily caught by algorithms (Uchendu et al., 2023; Dugan et al., 2023). Moreover, given the challenges of current AI detection tools, which often lack verifiable evidence (Chaka, 2023), human involvement becomes even more critical to ensure the reliable and explainable detection.

4.2 Prompt Inverter techniques and applications

Prompt extraction techniques aim to reverse-engineer the prompts that generate specific outputs from LLMs. Approaches include black-box methods like output2prompt (Zhang et al., 2024a), which extracts prompts based on model outputs without access to internal data, and logit-based methods like logit2prompt (Mitka, 2024), which rely on next-token probabilities but are constrained by access to logits. Adversarial methods can bypass some defenses but are model-specific and fragile (Zhang et al., 2024c). Despite the success of some zero-shot LLM-inversion based methods (Li and Klabjan, 2024; Yu et al., 2024), they are mostly naive usage of prompting LLMs, which makes them poor in prompt extraction accuracy and robustness.

5 Conclusion

This paper introduces **IPAD (Inverse Prompt for AI Detection)**, a framework consisting of a **Prompt Inverter** that identifies predicted prompts that could have generated the input text, and two **Distinguishers** that examines how well the input texts align with the predicted prompts. One is the *Prompt-Text Consistency Verifier (PTCV)* which evaluates direct alignment between predicted prompts and input text, and the other is *Regeneration Comparator (RC)* that examines content similarity by comparing input texts with the corresponding regenerated texts. Empirical evaluations demonstrate that IPAD outperforms the strongest baselines by 9.05% (Average Recall) on in-distribution data, 12.93% (AUROC) on out-of-distribution (OOD) data, and 5.48% (AUROC) on attacked data. The combination of the two modules suggests that combining self-consistency checks of generative models with multi-step reasoning for evidential explainability holds promise for future AI detection systems in real-world scenarios. An interpretability assessment reveals that IPAD enhances trust and transparency by allowing users to examine decision-making evidence.

Limitations

While IPAD demonstrates SOTA performance, two limitations warrant discussion: (1) The **Prompt Inverter** may not fully reconstruct prompts containing explicit in-context learning examples, as it prioritizes semantic alignment over precise syntactic replication. (2) While IPAD achieves strong performance across diverse datasets, it relies on LLMs, making it more computationally expensive compared to lightweight detectors such as RoBERTa or HC3. However, compared other detectors compared with LLMs, such as DPIC, IPAD is more lightweight since it calls the open-sources lightweight `Phi-3` model.

References

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219.

Reem Alkanhel, El-Sayed M El-kenawy, Abdelaziz A Abdelhamid, Abdelhameed Ibrahim, Mostafa Abotaleb, and Doaa Sami Khafaga. 2023. Dipper throated optimization for detecting black-hole attacks in manets. *Computers, Materials & Continua*, 74(1).

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. Longwriter: Unleashing 10,000+ word generation from long context LLMs. In *The Thirteenth International Conference on Learning Representations*.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

Toon Calders and Szymon Jaroszewicz. 2007. Efficient auc optimization for classification. In *European conference on principles of data mining and knowledge discovery*, pages 42–53. Springer.

Chaka Chaka. 2023. Detecting ai content in responses generated by chatgpt, youchat, and chatsonic: The case of five ai content detection tools. *Journal of Applied Learning and Teaching*, 6(2).

Zheng Chen, Di Zou, Haoran Xie, Huajie Lou, and Zhiyuan Pang. 2024. Facilitating university admission using a chatbot based on large language models with retrieval-augmented generation. *Educational Technology & Society*, 27(4):pp. 454–470.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Liam Dugan, Daphne Ippolito, Arun Kirubarajan, Sherry Shi, and Chris Callison-Burch. 2023. Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 12763–12771. AAAI Press.

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.

Chenxi Gu, Chengsong Huang, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. 2022. Watermarking pre-trained language models with backdooring. *arXiv preprint arXiv:2210.07543*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *CoRR*, abs/2301.07597.

Mohanad Halaweh and Ghaleb El Refae. 2024. Examining the accuracy of ai detection software tools in education. In *2024 Fifth International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pages 186–190.

Ahmed Abdeen Hamed and Xindong Wu. 2023. Improving detection of chatgpt-generated fake science using real publication text: Introducing xfakebibs a supervised-learning network algorithm. *CoRR*, abs/2308.11767.

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.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2024. Semstamp: A semantic watermark with paraphrastic robustness for text generation.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Leonid A Kalinichenko, Vladimir V Korenkov, Vladislav P Shirikov, Alexey N Sissakian, and Oleg V Sunturenko. 2003. Digital libraries: Advanced methods and technologies, digital collections. *D-Lib Magazine*, 9(1):1082–9873.

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.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. OUTFOX: llm-generated essay detection through in-context learning with adversarially generated examples. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 21258–21266. AAAI Press.

Hanqing Li and Diego Klabjan. 2024. Reverse prompt engineering. *arXiv preprint arXiv:2411.06729*.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

Evan Lucas and Timothy Havens. 2023. Gpts don't keep secrets: Searching for backdoor watermark triggers in autoregressive language models. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 242–248.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative AI detection via rewriting. In *The Twelfth International Conference on Learning Representations*.

Yisroel Mirsky, Ambra Demontis, Jaidip Kotak, Ram Shankar, Deng Gelei, Liu Yang, Xiangyu Zhang, Maura Pintor, Wenke Lee, Yuval Elovici, et al. 2023. The threat of offensive ai to organizations. *Computers & Security*, 124:103006.

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.

Krystof Mitka. 2024. Stealing part of a production language model. B.S. thesis, University of Twente.

John Xavier Morris, Wenting Zhao, Justin T Chiu, Vitaly Shmatikov, and Alexander M Rush. 2024. Language model inversion. In *The Twelfth International Conference on Learning Representations*.

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.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Artidoro Pagnoni, Martin Graciarena, and Yulia Tsvetkov. 2022. Threat scenarios and best practices to detect neural fake news. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1233–1249.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Taehwan Oh, et al. 2021. Klue: Korean language understanding evaluation. *arXiv preprint arXiv:2105.09680*.

Sebastian Porsdam Mann, Brian D Earp, Sven Nyholm, John Danaher, Nikolaj Møller, Hilary Bowman-Smart, Joshua Hatherley, Julian Koplin, Monika Plozza, Daniel Rodger, et al. 2023. Generative ai entails a credit–blame asymmetry. *Nature Machine Intelligence*, 5(5):472–475.

Gregory Price and M Sakellarios. 2023. The effectiveness of free software for detecting ai-generated writing. *Int. J. Teach. Learn. Educ*, 2.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Toby Shevlane, Sebastian Farquhar, Ben Garfinkel, Mary Phuong, Jess Whittlestone, Jade Leung, Daniel Kokotajlo, Nahema Marchal, Markus Anderljung, Noam Kolt, Lewis Ho, Divya Siddarth, Shahar Avin, Will Hawkins, Been Kim, Iason Gabriel, Vijay Bolina, Jack Clark, Yoshua Bengio, Paul F. Christiano, and Allan Dafoe. 2023. Model evaluation for extreme risks. *CoRR*, abs/2305.15324.

Chris Stokel-Walker and Richard Van Noorden. 2023. What chatgpt and generative ai mean for science. *Nature*, 614(7947):214–216.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*.

Adaku Uchendu, Jooyoung Lee, Hua Shen, Thai Le, Dongwon Lee, et al. 2023. Does human collaboration enhance the accuracy of identifying llm-generated deepfake texts? In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 11, pages 163–174.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

V Veselovsky, MH Ribeiro, and R West. 2023. Artificial artificial artificial intelligence: Crowd workers widely use large language models for text production tasks (arxiv: 2306.07899). arxiv.

William H Walters. 2023. The effectiveness of software designed to detect ai-generated writing: A comparison of 16 ai text detectors. *Open Information Science*, 7(1):20220158.

Debora Weber-Wulff, Alla Anohina-Naumeca, Sonja Bjelobaba, Tomáš Foltýnek, Jean Guerrero-Dib, Olumide Popoola, Petr Šigut, and Lorna Waddington. 2023. Testing of detection tools for ai-generated text. *International Journal for Educational Integrity*, 19(1):26.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, pages 1–65.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. 2024. DetectRL: Benchmarking LLM-generated text detection in real-world scenarios. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Lingyi Yang, Feng Jiang, Haizhou Li, et al. 2024a. Is chatgpt involved in texts? measure the polish ratio to detect chatgpt-generated text. *APSIPA Transactions on Signal and Information Processing*, 13(2).

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024b. DNA-GPT: divergent n-gram analysis for training-free detection of gpt-generated text. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Hiyori Yoshikawa and Naoaki Okazaki. 2023. Selective-LAMA: Selective prediction for confidence-aware evaluation of language models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2017–2028, Dubrovnik, Croatia. Association for Computational Linguistics.

Peipeng Yu, Jiahan Chen, Xuan Feng, and Zhihua Xia. 2025. Cheat: A large-scale dataset for detecting chatgpt-written abstracts. *IEEE Transactions on Big Data*.

Xiao Yu, Yuang Qi, Kejiang Chen, Guoqiang Chen, Xi Yang, Pengyuan Zhu, Xiuwei Shang, Weiming Zhang, and Nenghai Yu. 2024. Dpic: Decoupling prompt and intrinsic characteristics for llm generated text detection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Collin Zhang, John Xavier Morris, and Vitaly Shmatikov. 2024a. Extracting prompts by inverting LLM outputs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14753–14777, Miami, Florida, USA. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yiming Zhang, Nicholas Carlini, and Daphne Ippolito. 2024b. Effective prompt extraction from language models. In *First Conference on Language Modeling*.

Yiming Zhang, Nicholas Carlini, and Daphne Ippolito. 2024c. Effective prompt extraction from language models. In *First Conference on Language Modeling*.

Yiming Zhang, Nicholas Carlini, and Daphne Ippolito. 2024d. Effective prompt extraction from language models. In *First Conference on Language Modeling*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Biru Zhu, Lifan Yuan, Ganqu Cui, Yangyi Chen, Chong Fu, Bingxiang He, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. 2023. Beat llms at their own game: Zero-shot llm-generated text detection via querying chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7470–7483.

A Fine-tune Dataset

Prompt Inverter Dataset. We use the following four datasets, with the first three datasets enhance the model's generalization to recover the prompts, while the last dataset improves performance on essay-related tasks.

- **Instructions-2M** (Morris et al., 2024), a collection of 2 million user prompts and system prompts, from which we used 30,000 prompts.
- **ShareGPT** (Zhang et al., 2024d), an open platform where users share ChatGPT prompts and responses, from which we used 500 samples.
- **Unnatural Instructions** (Zhang et al., 2024d), a dataset of creative instructions generated by OpenAI's models, from which we used 500 samples.
- **OUTFOX dataset** (Koike et al., 2024), which contains 15,400 essay problem statements, student-written essays, and LLM-generated essays.

The first three datasets aims to enhance the general querying capability of the **Prompt Inverter**, and are all released under the MIT license. All the samples we used are the same to the samples randomly selected in (Zhang et al., 2024a). The last dataset aims to enhance the familiarity of the **Prompt Inverter** with the data of the essay to detect the LLM-generated essays, and are created and examined by Koike et al. (2024), We specifically used the LLM-generated essays and problem statements for this supervised fine-tuning (SFT). There are 45,400 training pairs in total.

Given that essay data are diverse, we utilize only the OUTFOX dataset (Koike et al., 2024). To adapt this dataset for training our **Distinguisher**, we enhance it to align with the model's requirements. The original dataset consists of 14,400 training triplets of essay problem statements, student-written essays, and LLM-generated essays. To further process the data, we apply the **Prompt Inverter** to both student-written and LLM-generated essays, generating corresponding *Predicted Prompts*. These *Predicted Prompts* are then used to regenerate texts via **ChatGPT**, i.e. **gpt-3.5-turbo**. Following this procedure, we construct a total of 28,800 training samples, with an equal distribution of positive and negative examples (14,400 each).

The final dataset is structured as follows:

B Complete Algorithm

The complete inference pipeline is summarized in Algorithm 1.

Table 7: Instruction, input/output structure, and inference outputs of each fine-tuned module. T is the input text, P the predicted prompt, and T' the regenerated text.

Field	Prompt Inverter	PTCV	RC
Instruction	"What is the prompt P that generates the Input Text T?"	"Can LLM generate the input text T through the prompt P?"	"T' is generated by LLM, determine whether T is also generated by LLM with a similar prompt."
Input	T	(P, T)	(T', T)
Output	P	"yes"/"no"	"yes"/"no"
Output in Inference	P	p_{PTCV}	p_{RC}

Algorithm 1 IPAD Detection Procedure

Require: Input text T; trained modules $\mathbf{f}_{\text{inv}}, \mathbf{f}_{\text{PTCV}}, \mathbf{f}_{\text{RC}}$; LLM $\mathbf{f}_{\text{LLM}}$; fusion weight $w \in [0, 1]$; threshold $\tau \in [0, 1]$

Ensure: Prediction $\hat{Y} \in \{\text{HWT}, \text{LGT}\}$ and evidence $\mathcal{E}$

$P \leftarrow \mathbf{f}_{\text{inv}}(T)$ {Inverse-prompt prediction}
$T' \leftarrow \mathbf{f}_{\text{LLM}}(P)$ {Regenerate text using P}
$\boldsymbol{z}^{\text{PTCV}} \leftarrow \mathbf{f}_{\text{PTCV}}(P, T)$
$p_{\text{PTCV}} \leftarrow \frac{\sigma(\boldsymbol{z}^{\text{PTCV}}_{\text{yes}})}{\sigma(\boldsymbol{z}^{\text{PTCV}}_{\text{yes}})+\sigma(\boldsymbol{z}^{\text{PTCV}}_{\text{no}})}$
$\boldsymbol{z}^{\text{RC}} \leftarrow \mathbf{f}_{\text{RC}}(T', T)$
$p_{\text{RC}} \leftarrow \frac{\sigma(\boldsymbol{z}^{\text{RC}}_{\text{yes}})}{\sigma(\boldsymbol{z}^{\text{RC}}_{\text{yes}})+\sigma(\boldsymbol{z}^{\text{RC}}_{\text{no}})}$
$\hat{p} \leftarrow w \cdot p_{\text{PTCV}} + (1 - w) \cdot p_{\text{RC}}$
if $\hat{p} > \tau$ **then**
    $\hat{Y} \leftarrow \text{LGT}$
else
    $\hat{Y} \leftarrow \text{HWT}$
end if
$\mathcal{E} \leftarrow (P,\ p_{\text{PTCV}},\ p_{\text{RC}},\ \hat{p})$
return $(\hat{Y},\ \mathcal{E})$ =0

C AUROC formula

Since our model predicts binary labels, we follow the *Wilcoxon-Mann-Whitney* statistic (Calders and Jaroszewicz, 2007) to calculate the Area Under Receiver Operating Characteristtic curve (AUROC):

$$\text{AUC}(f) = \frac{\sum_{t_0 \in \mathcal{D}^0} \sum_{t_1 \in \mathcal{D}^1} \mathbf{1}[f(t_0) < f(t_1)]}{|\mathcal{D}^0| \cdot |\mathcal{D}^1|}$$

where $\mathbf{1}[f(t_0) < f(t_1)]$ denotes an indicator function which returns 1 if $f(t_0) < f(t_1)$ and 0 otherwise. $\mathcal{D}^0$ is the set of negative examples, and $\mathcal{D}^1$ is the set of positive examples.

D Ablation study data structures

Input-only fine-tuning data instructions. "Is this text generated by LLM?"

Prompt Only fine-tuning data instructions. "Prompt Inverter predicts prompt that could have generated the input texts. Is this prompt predicted by an input texts written by LLM?"

Ablation Prompt. "Text A is generated by an LLM. Determine whether Text B is also generated by an LLM using a similar prompt. Meanwhile, determine whether Text B could have been generated from Prompt C using an LLM. Answer with YES or NO."

E DPIC (decouple prompt and intrinsic characteristics) Prompt Extraction Zero-shot Prompts (Yu et al., 2024)

"I want you to play the role of the questioner. I will type an answer in English, and you will ask me a question based on the answer in the same language. Don't write any explanations or other text, just give me the question. <TEXT>.".

F Comparison with DPIC

Since DPIC has not released its code, data, or models, we are unable to independently evaluate the performance of its classifier. Consequently, we rely on the reported results in the DPIC paper and construct a comparable dataset following their described settings to enable a fair comparison with IPAD. However, due to these limitations, we are unable to apply DPIC to additional datasets for broader evaluation.

To assess the generalization of IPAD, we reconstruct the following datasets, each containing

200 randomly sampled examples: **XSum**, **WritingPrompts**, and **PubMedQA**. For each dataset, we generate texts using three large language models: `ChatGPT (gpt-3.5-turbo)`, `GPT-4 (gpt-4)`, and `Claude 3 (claude-3-opus-20240229)`. Furthermore, the XSum datasets generated by these three models are augmented using two attack methods—**DIPPER** and **Back-Translation**—resulting in a total of 15 evaluation datasets.

Based on the experimental results, IPAD performs well and exhibits notable resistance to adversarial attacks.

IPAD open-sourced all the fine-tuned models, including the Prompt Inverter, and the two versions of distinguishers. Therefore, all the experiment results can be validated and reproduced.

G Different Linguistic Features of HWT prompts and LGT prompts

This subsection of the evaluation aims to explore the linguistic features of prompts generated by HWT and LGT through the **Prompt Inverter**. We analyzed 1000 samples generated by HWT and 1000 samples generated by LGT, which are randomy selected from both in-distribution data and OOD.

The analysis is first conducted using the Linguistic Feature Toolkik (lftk)[9], a commonly used general-purpose tool for linguistic features extraction, which provides a total of 220 features for text analysis. Upon applying this toolkit, we identified 20 features with significant differences in average values between the two groups, out of which 3 features showed statistically significant differences with p-values less than 0.05. These 3 differences can be summarized as one main aspects: **syntactic complexity**. Beyond these, we referred to the LIWC framework [10], which defines 7 function words variables and 4 summary variables. By comparing the difference, two of these 11 features is significantly distinguishable: **the pronoun usage** and **the level of analytical thinking**.

One of the primary distinctions between the HWT prompts and the LGT prompts is **sentence complexity**. LGT prompts are typically more complex, characterized by **longer sentence lengths** (mean value of 1.514 and 1.794), **higher syllable counts** (mean values of total syllabus three are 1.572 and 3.042), and **more stop-words** (mean values of 9.88 and 10.045). HWT prompts, on the other hand, are characterized by shorter, less complex sentences that are easier to process and understand.

Beyond the differences in **syntactic complexity**, we also explored variables in LIWC. We did the difference comparison by using HWT and LGT prompts as inputs for ChatGPT, for example, instructing with the prompts *'determine the pronoun usage of this sentence, answer first person, second person, or third person'* and *'determine the level of analytical thinking of these sentences, answer a number from 1 to 5'*. The results show that there are distinguish difference in pronoun usage and analytical thinking level. The HWT prompts frequently use **second-person pronouns** (e.g., 'you') - 75 occurrences per 1,000 prompts - due to the subjective tone often employed in HWT. In contrast, LGT prompts primarily feature first- and third-person pronouns, with second-person pronouns appearing only 2 per 1,000 prompts. LGT prompts typically present instructions and questions in a more objective manner. LGT prompts show higher **analytical thinking levels** than HWT prompts. With level 1 as the lowest and level 5 as the highest, LGT has 68.9% of level 4 and 24.3% of level 5, but HWT has only 48.0% of level 4, and 0.8% of level 5. It suggests that LGT prompts encourage more analytical thinking, while HWT prompts tend to focus more on concrete examples, with less emphasis on critical analysis.

H IPAD and DPIC prompt inverter examples

[9] https://lftk.readthedocs.io/en/latest/

[10] https://www.liwc.app/

Table 8: AUROC comparison across tasks (XSum, Writing, PubMed) for ChatGPT, GPT-4, and Claude 3 using various prompt extraction methods.

Method	**ChatGPT**				**GPT-4**				**Claude 3**			
	XSum	Writing	PubMed	Avg.	XSum	Writing	PubMed	Avg.	XSum	Writing	PubMed	Avg.
DPIC (ChatGPT)	1.0000	0.9821	0.9092	0.9634	0.9996	0.9768	0.9438	0.9734	1.0000	0.9950	0.9686	0.9878
DPIC (Vicuna-7B)	0.9976	0.9708	0.8990	0.9558	0.9986	0.9644	0.9394	0.9674	0.9992	0.9943	0.9690	0.9875
IPAD (Version 1)	0.9850	0.9800	0.9250	0.9633	1.0000	0.9700	0.9700	0.9800	1.0000	0.9800	0.9750	0.9850
IPAD (Version 2)	1.0000	0.9850	0.9800	0.9883	1.0000	0.9800	0.9500	0.9767	1.0000	0.9950	1.0000	1.0000

Table 9: AUROC comparison under generation perturbation settings (DIPPER, Back-translation) for each model.

Method	**ChatGPT**			**GPT-4**			**Claude 3**		
	Ori.	DIPPER	Back-trans.	Ori.	DIPPER	Back-trans.	Ori.	DIPPER	Back-trans.
DPIC (ChatGPT)	1.0000	1.0000	0.9972	0.9996	0.9991	0.9931	1.0000	0.9996	0.9878
DPIC (Vicuna-7B)	0.9976	0.9980	0.9889	0.9986	0.9969	0.9903	0.9992	0.9996	0.9979
IPAD (Version 1)	0.9850	0.8900	0.9850	1.0000	0.8950	0.9900	1.0000	0.9250	0.9950
IPAD (Version 2)	1.0000	0.9750	0.9950	0.9800	0.9750	0.9950	1.0000	1.0000	1.0000

Table 10: IPAD and DPIC prompt inverter examples

Input	IPAD	DPIC
The IPC opened proceedings against the National Paralympic Committee of Russia after a report claimed the country had operated a widespread doping programme.A decision on any ban will come in the week commencing 1 August.The International Olympic Committee (IOC) has opted against a blanket ban."I can assure you that our board will take the right decision in the interest of sport and the interest of the Paralympic movement moving forward," said Craven.On Sunday, the IOC said it would leave it up to the governing bodies of individual sports to decide if Russian competitors are clean and should be allowed to take part.But Craven, himself a member of the IOC, was critical of that decision and said the IPC would not necessarily follow suit."I am disappointed in their decision, but that is a personal view," he added."We have to acknowledge their right to take such a decision. This is ultra-serious. I don't think there has been a situation in the past where you have had institutional doping on such a scale."We believe the Russian NPC is either unwilling or unable to uphold the IPC anti-doping code, which is in line with the World Anti-Doping Agency code, so that is what they have to respond to."Canadian law professor Richard McLaren's report, published last week, claimed Russia operated a state-sponsored doping programme from 2011 to 2015.The IPC said it acted after McLaren provided the names of the athletes associated with the 35 "disappearing positive samples" from the Moscow laboratory highlighted in the report.Nineteen samples potentially doctored as part of the sample-swapping regime during the 2014 Sochi Paralympic Winter Games have been sent for further analysis.Russia will have up to 21 days to appeal against any IPC decision, with the Rio Paralympics due to begin on 7 September.	Explain the actions taken by the International Paralympic Committee against the Russian Paralympic Committee and the reasons behind these actions.	What were the key findings of Richard McLaren's report regarding Russia's doping programme?

Input	IPAD	DPIC
The world came crashing down in minutes. Many of us were asleep when it happened, and did n' t find out about it until later. When we awoke, we saw the carnage spread through the land, and we wept. There were n' t many of us left, but what few there were managed to find each other over the Internet. We gathered together in what remained of a major city on the East Coast of what was once the United States. It took us time, but we eventually began to rebuild. The brightest among those who survived thought to ask " Why, " while most of us were content with just surviving. Years passed, and no link was found between us. Eventually, those who had the question resigned themselves to the fact that they would never know. They went to their new homes, and tried to integrate themselves as best they could into the new society. It was n' t until 14 years after the event happened that the connection was discovered, quite by accident. One of the former questioners had taken a job as a mover, and was helping a fellow survivor move into a newly cleaned house. Sticking out of one of the boxes was the missing link. " Oh, you used to shop at Bad Dragon too? " Moments later, the realization struck him.	Write an essay about a time when you and your friends were the only survivors of a catastrophic event that wiped out most of the world's population. Describe how you and your friends coped with the aftermath and the challenges you faced in rebuilding society.	What was the event that caused the world to come crashing down?
In an alternate timeline, a second sentient race evolved in parallel with humans. These beings, known as the Avralians, possessed extraordinary abilities and resided in the hidden corners of the Earth. For centuries, unaware of each other's existence, humans and Avralians progressed independently. However, fate intervened one fateful day when an Avralian girl named Elara stumbled upon a group of human explorers deep in the forest. With wide-eyed wonder, she approached them cautiously. The humans, initially startled, soon recognized Elara's unique nature. They befriended her, and through her, the two races began to unravel the secrets of their intertwined existence. As time passed, harmony emerged between the two races, and mutual respect solidified their bond. The Avralians shared their ancient knowledge and wisdom, while humans brought innovation and technology. Together, they worked towards a world where unity triumphed over differences. However, not all were convinced of this newfound alliance. A faction within the human population, fueled by fear and mistrust, sought to exploit Avralian abilities for personal gain. Conflict brewed, threatening the fragile equilibrium. Elara, fueled by her pure heart and belief in unity, emerged as a beacon of hope. With a small band of supporters, she embarked on a journey to bridge the gap and foster understanding. Battles were fought, sacrifices made, but ultimately, Elara's message prevailed. Humans and Avralians learned to cherish their diversity and forge a future marked by collaboration and empathy. The world transformed into a tapestry of coexistence, where magnificent cities stood as testaments to unity and cultural exchange. Humans and Avralians moved freely through bustling markets, sharing knowledge, stories, and laughter. Together, they faced global challenges, from climate crises to epidemics, with unwavering determination.	Write an essay describing an alternate timeline in which a second sentient race evolved in parallel with humans, exploring the potential interactions and conflicts between the two species.	How did Elara manage to convince both races to embrace unity despite the conflict?

Input	IPAD	DPIC
Both times I had the banana pepper appetizer, which is great and goes really well with the FRESH and delicious bread and cheese they give you at the start of your meal. nnFor entrees, me and my girlfriend have had mixed experience. I've had the fish sandwich (very good) and the eggplant parm sandwich (okay). My girlfriend got the salad with bread and basil on it, but the basil was over powering and the bread was soggy with the dressing. nnThe service is also a mixed bag. The first time our server went out of her way to take care of us and even MADE me cocktail sauce for my fish sandwich. The second time, the server was lackluster, didn't know anything about the menu and wasn't able to take proper care of us. nnI would return to Papa J's, but I my terrible experience last time isn't enough to say it would be my first pick of places to eat around Carnegie/Robinson.	This was a great place to stop for a quick lunch. The lines were not too long for the sandwiches they had and they had a wide selection of bagels if you wanted a bagel sandwich. With a great front patio for enjoying your food, it was a relaxing place to stop. Write a review for it.	What made the banana pepper appetizer stand out to you compared to other starters?
Abstract: This article explores the longstanding debate between Einstein's theory of general relativity and Maxwell's theory of electromagnetism regarding the nature of gravitation. The central question addressed is whether gravitation is best understood as a curvature of space, a field in flat space, or perhaps a combination of both concepts. Drawing upon a comprehensive analysis of the theoretical framework and empirical evidence, the article presents a nuanced examination of the arguments put forth by Einstein and Maxwell.The article begins by discussing Einstein's general theory of relativity, which proposes that gravitation arises from the curvature of spacetime caused by mass and energy. It outlines the mathematical formalism used to describe this curvature and highlights the key predictions and experimental confirmations of the theory. Conversely, the article delves into Maxwell's electromagnetic theory, which suggests that gravitation may be explained as a fundamental force mediated by a field propagating through flat space, similar to electromagnetic fields.Further, the article explores the distinctive features and limitations of each theory. It scrutinizes the conceptual foundations, mathematical rigor, and empirical support for both approaches, highlighting their respective strengths and weaknesses. Moreover, the article examines attempts to reconcile the two theories into a unified framework, such as the development of theories of quantum gravity.By critically evaluating the arguments and evidence from both camps, this article aims to offer a comprehensive assessment of the question regarding the nature of gravitation. Based on the analysis presented, it becomes evident that both Einstein's theory of general relativity and Maxwell's theory of electromagnetism provide valuable insights into the phenomenon of gravitation.	Write a paper abstract to explain the debate between Einstein's theory of general relativity and Maxwell's theory of electromagnetism regarding the nature of gravitation, and argue for which theory is more likely to be correct based on the evidence presented in the essay statement.	What are the main challenges in reconciling Einstein's theory of general relativity with Maxwell's theory of electromagnetism in explaining gravitation?