IPAD: Inverse Prompt for AI Detection - A Robust and Explainable LLM-Generated Text Detector

Anonymous ACL submission

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 explainable 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 a **Distinguisher** that examines how well the input texts align with the predicted prompts. We develop and examine two versions of **Distinguishers**. Empirical evaluations demonstrate that both **Distinguishers** perform significantly better than the baseline methods, with version2 outperforming baselines by 9.73% on in-distribution data (F1-score) and 12.65% on OOD data (AUROC). Furthermore, a user study is conducted to illustrate that IPAD enhances the AI detection trustworthiness by allowing users to directly examine the decision-making evidence which provide 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 framework comprising two key components as shown in Figure 1: a **Prompt Inverter** that reconstructs prompts from

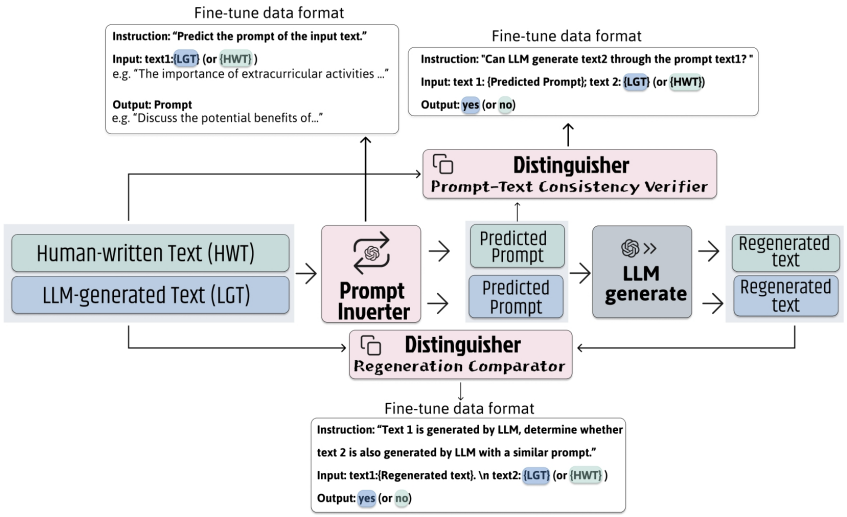

Figure 1: The overall workflow of our proposed IPAD framework

input text, and a **Distinguisher** that classifies text as HWT or LGT. We consider and examine two distinct approaches for the Distinguisher: the *Prompt-Text Consistency Verifier* evaluates direct alignment between predicted prompts and input text, while the *Regeneration Comparator* examines contents similarity by comparing input texts with the corresponding regenerated texts. Our framework introduces a paradigm shift in AI text detection by establishing an interpretable pipeline that reveals the underlying step-by-step reasoning process, therefore it enhances both detection robustness and explainability. Through comprehensive experiments comparing these two **Distinguishers**, we demonstrate their respective strengths and limitations, providing new insights into how different text characteristics affect detection performance.

Empirical evaluations demonstrate that both **Distinguishers** significantly surpass baseline methods, with the *Regeneration Comparator* outperforming baselines by 9.73% (F1-score) on in-distribution data and 12.65% (AUROC) on out-of-distribution (OOD) data. Additionally, the *Regeneration Comparator* exhibits better performance than the *Prompt-Text Consistency Verifier* on attacked data with 3.78% (F1-score), and slightly better on OOD data with 0.13% (F1-score). Furthermore, a user study indicates that IPAD enhances the AI detection experience and trustworthiness by allowing users to directly examine its decision-making evidence, which includes the predicted prompts and regenerated texts, and hence provide transparent and interpretable support for its state-of-the-art detection results. Code is anonymously available [1].

2 Methodology

In this section, we illustrate our method step by step. First, we introduce the overall workflow. After that, we demonstrate the details of supervised fine-tuning (SFT) the **Prompt Inverter** and **Distinguisher** .

2.1 Workflow

IPAD consists of a **Prompt Inverter** and a **Distinguisher**, both fine-tuned on Microsoft's open model Phi3-medium-128k-instruct, which together form a complete detection workflow as illustrated in Figure 1. For the **Distinguisher**, we develop two models and examine them in Section 3.

The *Input Text* (T) is either human-written (HWT) or LLM-generated (LGT), and it is processed by the **Prompt Inverter** to predict the most likely prompt that could have generated it. This *Predicted Prompt* (P) is assumed to be the input that an LLM would have used to produce the text.

$$P = f_{\text{inv}}(T)$$

where f_{inv} stands for **Prompt Inverter**.

For the next step, the *Predicted Prompt* (P) is fed into an LLM (we use ChatGPT, i.e. gpt-3.5-turbo by default, and other LLMs for evaluations), to generate a corresponding *Regenerated Text* (T').

$$T' = f_{\text{LLM}}(P)$$

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

After that, we consider two **Distinguishers**. The first one is *Prompt-Text Consistency Verifier*, in which the *Input Text* (T) and the *Predicted Prompt* (P) are passed to the model.

The *Prompt-Text Consistency Verifier* determines whether the *Predicted Prompt* (P) can reasonably generate the given *Input Text* (T) using an LLM. The model outputs either a **"yes"** or **"no"** response. If the *Predicted Prompt* (P) is likely to produce the *Input Text* (T) when fed into the LLM, the model is expected to output **"yes"**, indicating that the *Input Text* (T) is likely LGT. Conversely, if the *Predicted Prompt* (P) does not align well with the *Input Text* (T), the model outputs **"no"**, suggesting that the *Input Text* (T) is less likely to have been generated by the LLM with the *Predicted Prompt* (P), and is therefore more likely to be HWT.

$$S = f_{\text{PTCV}}(T, P)$$

where f_{PTCV} stands for *Prompt-Text Consistency Verifier* in the **Distinguisher**.

The second **Distinguisher** is *Regeneration Comparator*, which considers both the *Input Text* (T) and the *Regenerated Text* (T').

The *Regeneration Comparator* determines whether the *Input Text* (T) aligns with the *Regenerated Text* (T'), and then outputs either a **"yes"** or **"no"** response. If the *Input Text* (T) is LGT, the model is expected to output "yes," which indicates that both the *Input Text* (T) and the *Regenerated Text* (T') were generated by an LLM from similar prompts. Conversely, if the *Input Text* (T) is HWT, the model is expected to output "no," which signifies that the *Input Text* (T) is meaningfully distinct from the *Regenerated Text* (T') and thus unlikely to have been generated by an LLM.

$$S = f_{\text{RC}}(T, T')$$

where f_{RC} stands for *Regeneration Comparator* in the **Distinguisher**.

Finally, for both **Distinguishers**,

$$\hat{Y} = \begin{cases} \text{LGT}, & \text{if } S = \text{Yes} \\ \text{HWT}, & \text{if } S = \text{No} \end{cases}$$

where $\hat{Y}$ is the final decision of the *Input Text* (T).

2.2 Datasets

2.2.1 Prompt Inverter

The datasets used to fine-tune the **Prompt Inverter** include several widely adopted resources in the field. These are:

- **Instructions-2M** (Morris et al., 2024), a collection of 2 million user and system prompts, from which we used 30,000 prompts.
- **ShareGPT** (Zhang et al., 2024b), an open platform where users share ChatGPT prompts and responses, from which we used 500 samples.
- **Unnatural Instructions** (Zhang et al., 2024b), a dataset of diverse, creative instructions generated by OpenAI's text-davinci-002, 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).

For all 45,400 training pairs, the format is standardized as follows: **Instruction**: *"Predict the prompt of the Input Text."* **Input**: *{LGT} or {HWT}* **Output**: *{Corresponding prompt}*.

2.2.2 Distinguishers

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**.

The final dataset is structured as follows:

Distinguisher version1 - *Prompt-Text Consistency verifier*: **Instruction**:*"Can LLM generate text2 through the prompt text1? "* **Input**: *text 1: {Predicted Prompt}; text 2: {LGT} (or {HWT})* **Output**: *yes* (or *no*)

Distinguisher version2 - *Regeneration Comparator*: **Instruction**: *"Text 1 is generated by an LLM. Determine whether Text 2 is also generated by an LLM with a similar prompt."* **Input**: *text 1: {Regenerated Text}; text 2: {LGT} (or {HWT})* **Output**: *yes* (or *no*)

Following this procedure, we construct a total of 28,800 training samples, with an equal distribution of positive and negative examples (14,400 each).

2.3 Training

The supervised fine-tuning (SFT) (Wei et al., 2022) process is performed on a dataset comprising the above-mentioned 45,400 pairs for **Prompt Inverter** and 28,800 pairs for both **Distinguishers**. We utilize 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[2] (Zheng et al., 2024). We train it using six A800 GPUs for 20 hours for **Prompt Inverter**, 7 hours for Distinguisher version1, and 4 hours for Distinguisher version2.

3 Experiments

We investigate the following questions through our experiments:

- Assess the robustness of IPAD (using various LLMs as generators, comparing with other detectors, and evaluating on out-of-distribution (OOD) datasets).
- Independently analyze the necessity and effectiveness of the **Prompt Inverter** and the **Distinguishers**.
- Explore the explainability of IPAD (through a user study and analysis of linguistic differences between prompts generated by HWT and LGT).

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 **Distinguishers**, thereby serving as our baseline. The OOD experiments refer to the DetectRL baseline (Wu et al., 2024), which is a comprehensive benchmark consisting of academic abstracts from the arXiv Archive (covering the years 2002 to 2017)[3], news articles from the XSum dataset (Narayan et al., 2018), creative stories from Writing Prompts (Fan et al., 2018), and social reviews from Yelp Reviews (**?**). It also employs three attack methods to simulate complex real-world detection scenarios, which includes the prompt attacks, paraphrase attacks, and perturbation attacks (Wu et al., 2024). All the testing sets have 1,000 samples in our experiments.

The **Area Under Receiver Operating Characteristic curve (AUROC)** is widely used for assessing detection method (Mitchell et al., 2023) because it considers the True Positive Rate (TPR) and False Positive Rate (FPR) across different classification thresholds. Since our models predicts binary labels, we follow the *Wilcoxon-Mann-Whitney* statistic (Calders and Jaroszewicz, 2007), and the formula is shown in appendix A. The **AvgRec** is the average of **HumanRec** and **MachineRec**. In our evaluation, **HumanRec** is the recall for detecting Human-written texts, and **MachineRec** is the recall for detecting LLM-generated texts (Li et al., 2024). The **F1 Score** provides a comprehensive evaluation of detector capabilities by balancing the model's Precision and Recall. We use **AvgRec** and **F1** on in-distribution data, and we use **AUROC** for OOD data to align the test benchmarks for the same dataset.

3.1.2 Robustness across different LLMs

The results of IPAD for detecting the dataset OUTFOX (Koike et al., 2024) across LLMs are presented in Table 1 and Table 2, respectively. They show that both versions are highly robust across various LLMs, while *Regeneration Comparator* is a bit more efficient.

As for *Regeneration Comparator*, when the original generator and re-generator are the same model, the performance is optimal. However, even when the re-generator is different from the original generator, the results remain impressive with ChatGPT used as the re-generator. These results imply that, in practical applications, it is possible to use a common set of LLMs as re-generators. If one or more correponding **Distinguishers** from different LLMs classify the results as 'yes', it can be inferred that the text is likely to be LGT, whereas if all **Distinguishers** classify the results as 'no', the text is more likely to be HWT. Furthermore, for applications aiming to save computational resources and

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

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

improve efficiency, using ChatGPT as the sole re-generator still yields robust performance across all tested models.

Original Generator	Metrics (%)			
	HumanRec	MachineRec	AvgRec	F1
ChatGPT	98.00%	99.80%	98.90%	98.89%
GPT-3.5	97.20%	99.90%	98.55%	98.53%
Qwen-turbo	98.00%	98.10%	98.05%	98.05%
Llama-3-70B	98.00%	100.00%	99.00%	98.99%

Table 1: IPAD with *Prompt-Text Consistency Verifier* performance on different LLMs

Original Generator	Re-Generator	Metrics (%)			
		HumanRec	MachineRec	AvgRec	F1
ChatGPT	ChatGPT	99.70%	100.00%	**99.85%**	**99.85%**
GPT-3.5	GPT-3.5	98.00%	100.00%	**99.00%**	**99.00%**
	ChatGPT	97.00%	100.00%	98.50%	98.50%
Qwen-turbo	Qwen-turbo	98.00%	98.40%	**98.20%**	**98.20%**
	ChatGPT	99.70%	94.40%	97.05%	97.13%
Llama-3-70B	Llama-3-70B	96.60%	100.00%	98.30%	98.30%
	ChatGPT	99.70%	99.40%	**99.55%**	**99.55%**

Table 2: IPAD with *Regeneration Comparator* performance on different LLMs

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

Table 3 compares the performance of two versions of IPAD with other detection methods in the OUTFOX dataset with and without attacks (Koike et al., 2024). The results show that both versions of IPAD generally outperform other detectors, while that IPAD with *Prompt-Text Consistency Verifier* for detecting ChatGPT with DIPPER attack performs worse. These results imply that IPAD with *Regeneration Comparator* demonstrates superior robustness compared to alternative detection methods in the OUTFOX dataset with and without attacks.

Table 4 presents the performance of various detection methods on OOD datasets to assess their generalizability, where the baseline data refer to DetectRL (Wu et al., 2024). The results demonstrate that IPAD with *Regeneration Comparator* consistently outperforms all other baselines in all OOD datasets with and without attacks. In contrast, IPAD with *Prompt-Text Consistency Verifier* exhibits strong performance on OOD datasets without attacks but shows a noticeable drop in effectiveness when subjected to attacks. For instance, while it achieves competitive results on datasets like XSum (99.90%) and Writing (99.20%), its performance against attacks, such as Prompt Attack (86.90%) and Paraphrase Attack (82.72%), is significantly lower than IPAD with *Regeneration Comparator*. This suggests that **IPAD with *Regeneration Comparator* demonstrates better generalizability and robustness.**

Original Generator	Detection Methods	Metrics (%)			
		HumanRec	MachineRec	AvgRec	F1
ChatGPT	RoBERTa-base	93.80%	92.20%	93.00%	92.90%
	RoBERTa-large	91.60%	90.00%	90.80%	90.70%
	HC3 detector	79.20%	70.60%	74.90%	73.80%
	OUTFOX	97.80%	92.40%	95.10%	95.00%
	IPAD version1	98.00%	99.80%	98.90%	98.89%
	IPAD version2	99.70%	100.00%	**99.85%**	**99.85%**
GPT-3.5	RoBERTa-base	93.80%	92.00%	92.90%	92.80%
	RoBERTa-large	92.60%	92.00%	92.30%	92.30%
	HC3 detector	79.20%	85.00%	82.10%	82.60%
	OUTFOX	97.60%	96.20%	96.90%	96.90%
	IPAD version1	97.20%	99.90%	**98.55%**	**98.53%**
	IPAD version2	97.00%	100.00%	98.50%	98.50%
ChatGPT with DIPPER Attack	RoBERTa-base	93.80%	89.20%	91.50%	91.30%
	RoBERTa-large	91.60%	97.00%	94.30%	94.40%
	HC3 detector	79.20%	3.40%	41.30%	5.50%
	OUTFOX	98.60%	66.20%	82.40%	79.00%
	IPAD version1	98.00%	75.10%	86.55%	87.93%
	IPAD version2	99.70%	95.40%	**97.55%**	**97.60%**
ChatGPT with OUTFOX Attack	RoBERTa-base	93.80%	69.20%	81.50%	78.90%
	RoBERTa-large	91.60%	56.20%	73.90%	68.30%
	HC3 detector	79.20%	0.40%	39.80%	0.70%
	OUTFOX	98.80%	24.80%	61.80%	39.40%
	IPAD version1	98.00%	95.40%	96.70%	96.74%
	IPAD version2	99.70%	98.00%	**98.85%**	**98.86%**

Table 3: Comparison of IPAD with other detectors on in-distribution data, where **IPAD version1** stands for **IPAD with *Prompt-Text Consistency Verifier*** and **IPAD version2** stands for **IPAD with *Regeneration Comparator***

OOD Datasets or attack type	Detection Methods				
	LRR	Fast-DetectGPT	Rob-Base	IPAD with version1	IPAD version2
Arxiv	48.17%	42.00%	81.06%	84.47%	**98.60%**
XSum	48.41%	45.72%	76.81%	**99.90%**	98.90%
Writing	58.70%	51.13%	86.29%	**99.20%**	95.80%
Review	58.21%	54.55%	87.84%	98.50%	**89.30%**
Avg. for non-attacked datasets	53.37%	48.35%	83.00%	95.52%	**95.65%**
Prompt Attack	54.97%	43.89%	92.81%	86.90%	**93.05%**
Paraphrase Attack	49.23%	41.15%	90.02%	82.72%	**95.89%**
Perturbation Attack	53.62%	44.38%	92.12%	94.96%	**95.32%**
Avg. for attacked datasets	52.61%	43.14%	91.65%	88.26%	**94.75%**
Avg.	53.04%	46.12%	86.70%	92.41%	**95.26%**

Table 4: The performance of IPAD in generalization assessment (AUROC). The selected detectors are evaluated on OOD data, all sourced from and processed using the DetectRL baseline, where **IPAD version1** stands for **IPAD with *Prompt-Text Consistency Verifier*** and **IPAD version2** stands for **IPAD with *Regeneration Comparator*.**

3.1.4 Robustness conclusion

Our experimental results demonstrate that both IPAD versions exhibit strong performance across different LLMs, outperforming existing detection methods and maintaining robustness on OOD datasets. The IPAD with *Regeneration Comparator* outperforming baselines by 9.73% (F1-score) on in-distribution data and 12.65% (AUROC) OOD data. Notably, IPAD with *Regeneration Comparator* achieves significantly better performance than IPAD with *Prompt-Text Consistency Verifier* in attack scenarios of 3.78% (F1-score). While IPAD with *Prompt-Text Consistency Verifier* performs robustly in standard settings, its performance declines when facing attacks. The calculation of these statistics are shown in Appendix B.

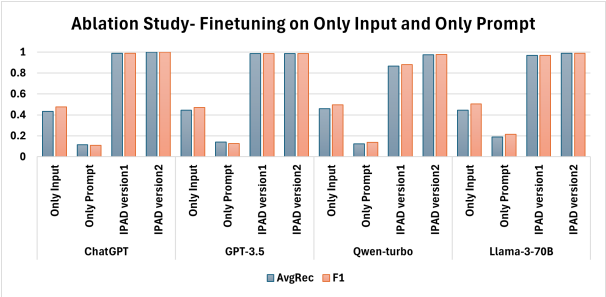

Figure 2: Ablation Study Results. The **IPAD version1** stands for **IPAD with *Prompt-Text Consistency Verifier*** and **IPAD version2** stands for **IPAD with *Regeneration Comparator*.**

3.2 Necessity and Effectiveness of Prompt Inverter and Distinguishers

3.2.1 Necissity of the Prompt Inverter and Distinguishers

To prove that it is necessary to fine-tune on IPAD with IPAD with *Prompt-Text Consistency Verifier* and *Regeneration Comparator*, we conducted ablation study to use the same finetune method on only *input texts* and only *predicted prompts*. The instructions are *"Is this text generated by LLM?"*, and *"Prompt Inverter predicts prompt that could have generated the input texts. Is this prompt predicted by an input texts written by LLM?"*, respectively.

The results shown in Figure 2 from the ablation study show that fine-tuning on either only the *input text* or only the *predicted prompt* leads to poor performance. This underscores the importance of fine-tuning on a combination of both the input text and predicted prompt, as explored in the *Prompt-Text Consistency Verifier*, or on the input text and regenerated text, as examined in the *Regeneration Comparator*, for more effective detection.

3.2.2 The effectivenss of the IPAD Prompt Inverter

We use DPIC (Yu et al., 2024) and PE (Zhang et al., 2024c) as baseline methods for prompt extraction. DPIC employs a zero-shot approach using the prompt states in Appendix C, 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**.

Evaluation	Bart-large-cnn	Sentence-Bert	BLEU	ROUGE-1
		LGT		
DPIC	-2.12	0.46	5.61E-05	0.04
PE	-2.23	0.58	3.21E-04	0.25
IPAD	**-1.84**	**0.69**	**0.24**	**0.51**
		HWT		
DPIC	-2.47	0.42	8.75E-06	0.06
PE	-2.39	0.53	2.56E-08	0.13
IPAD	**-2.22**	**0.57**	**1.30E-01**	**0.39**

Table 5: Comparison of the IPAD **Prompt Inverter** with other prompt extractors

3.2.3 The Effectiveness of the IPAD Distinguishers

To examine the effectiveness of the IPAD **Distinguishers**, we conducted a comparison study using the same dataset 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 and fourth methods involved directly prompting ChatGPT as follows:

Instruction: *"Text 1 is generated by an LLM. Determine whether Text 2 is also generated by an LLM with a similar prompt. Answer with only YES or NO."* **Input:** *"Text 1: {Regenerated Text}; Text 2: {LGT} or {HWT}"*.

and **Instruction:** *"Can LLM generate text2 through the prompt text1? Answer with only YES or NO."* with **Input:** *"Text 1: {Predicted Prompt}; Text 2: {Input text}"*.

The final results demonstrated that the other distinguishing methods performed worse than the two IPAD **Distinguishers**, highlighting the superior effectiveness of the IPAD **Distinguishers**.

3.3 Explanability Assessment of IPAD

3.3.1 Different Linguistic Features of HWT prompts and LGT prompts

This subsection of the evaluation aims to explore the linguistic features of prompts generated by

Distinguish Method	HumanRec	MachineRec	AvgRec	F1
Sentence-Bert (Threshold 0.67)	61.20%	95.20%	78.20%	63.51%
Bart-large-cnn (Threshold -2.52)	42.60%	97.20%	69.90%	43.96%
Prompt to ChatGPT version 1	33.20%	64.50%	48.85%	44.77%
Prompt to ChatGPT version 2	12.50%	100%	56.25%	12.50%
IPAD version 1	98.00%	99.80%	98.90%	98.10%
IPAD version 2	**99.70%**	**100%**	**99.85%**	**99.70%**

Table 6: Comparison of Different Distinguishers, where **IPAD version1** stands for **IPAD with *Prompt-Text Consistency Verifier*** and **IPAD version2** stands for **IPAD with *Regeneration Comparator*.**

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)[4], 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 [5], 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, as examples shown in Appendix D Figure 3.

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. As shown in Appendix D Figure 4, 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, as examples shown in Appendix D Figure 5.

3.4 User Study

To assess the explainability improvement of IPAD, we designed an IRB-approved user study 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 with screenshots shown in Appendix E[6][7][8] with IPAD's process (which displayed input texts, predicted prompts, regenerated texts, and final judgments). After evaluation, users rated IPAD on four key explainability dimensions. Transparency received strong ratings (40%:5, 60%:4), with users appreciating the visibility of intermediate 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 users acknowledging better detection but raising concerns about energy efficiency since IPAD runs three LLMs. Debugging received unanimous 5s, as users 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-

[4]https://lftk.readthedocs.io/en/latest/
[5]https://www.liwc.app/
[6]https://www.scribbr.com/ai-detector/
[7]https://quillbot.com/ai-content-detector
[8]https://app.gptzero.me/

box detectors.

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 of 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, 2023), but their effectiveness depends on corpus size and model diversity (Wu et al., 2025). Some other statistical methods use n-gram probability divergence (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., 2024d). 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 a **Distinguisher** that examines how well the input texts align with the predicted prompts. This design enables explainable evidence chains tracing unavailable in existing black-box detectors. Empirical results show that IPAD surpasses the baselines on all in-distribution, OOD, and attacked data. Furthermore, the **Distinguisher** (version2) - *Regeneration Comparator* outperforms the **Distinguisher** (version1) - *Prompt-Text Consistency Verifier*, especially on OOD and attacked data. While the local alignment in veresion1 approach provides explicit interpretability, it is more sensitive to adversarial attacks. In contrast, the global distribution in veresion2 matching approach implicitly learns generative LLM's distributional properties, which offers more robustness while maintaining explainability. This insight 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. A user study reveals that IPAD enhances trust and transparency by allowing users to examine decision-making evidence. Overall, IPAD establishes a new paradigm for more robust, reliable, and interpretable AI detection systems to combat the misuse of LLMs.

6 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 (e.g., formatted demonstrations), as it prioritizes semantic alignment over precise syntactic replication.

(2) Since IPAD achieves satisfactory OOD performance (12.65% improvement over baselines) by only adopting essay writing datasets for the fine-tuning of **Distinguishers**, we strategically deferred the exploration of more datasets. We will incorporate a wider and more diverse range of data in future works to explore if it can enhance robustness even further, including: creative/news domains, and triplet data formats (i.e., *"Can this {predicted prompt} generate the {Input text} using an LLM? One example generated by the predicted prompt is: {regenerated text}"*)

Acknowledgments

References

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12763–12771.

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*.

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. 2023. Improving detection of chatgpt-generated fake science using real publication text: Introducing xfakebibs a supervised learning network algorithm.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21258–21266.

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.

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, and Allan Dafoe. 2023. Model evaluation for extreme risks.

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

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.

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 Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. *XLNet: generalized autoregressive pretraining for language understanding*. Curran Associates Inc., Red Hook, NY, USA.

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.

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 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):

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

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.

B Calculation of Summary Statistics

- IPAD with *Regeneration Comparator* outperforms the baselines by 9.73% on in-distribution data. As shown in Table 3, RoBERTa-base has the best average F1 score of (92.9% + 92.8% + 91.3% + 78.9%) / 4. In comparison, the average F1 score for IPAD version 2 is (99.85% + 98.5% + 97.6% + 98.86%) / 4, showing an improvement of 9.73%.

- IPAD with *Regeneration Comparator* outperforms the baselines by 12.65% on in-distribution data. As shown in Table 4, RoBERTa-base achieves the highest average AUROC score, but since the F1-score is not available for the baseline, we use the AUROC difference to calculate the improvement, which is (95.65% - 83%) = 12.65%.

- IPAD with *Regeneration Comparator* outperforms IPAD with *Prompt-Text Consistency Verifier* by 0.13% on out-of-distribution (OOD) data. As shown in Table 4, IPAD version 2 has the highest AUROC of 95.65%, while IPAD version 1 has an AUROC of 95.52%, resulting in a 0.13% difference.

HWT Prompt	LGT Prompt
Should students be required to do **community** service? Why or why not?	Should community service be a **requirement** for students? Discuss the potential benefits and drawbacks of **mandatory** community service, including the impact on students' academic **performance**, **physical** and mental health, and personal interests. Consider the potential benefits of community service for the **community** as a whole, as well as the potential drawbacks for students who may not enjoy or benefit from such **activities**. Use specific examples and **evidence** to support your argument.
Explain why the author does not support the **Electoral** College.	Explain the reasons why the **Electoral** College should be canceled, including its outdated nature, **potential** for **corruption**, and lack of **representation** for the people.

Figure 3: Sentence Complexity Examples, where **HWT Prompt** stands for prompt generated by the Prompt Inverter from HWT, and **LGT Prompt** stands for prompt generated by the Prompt Inverter from LGT. The HWT Prompts have longer sentence lengths, more words with more than three syllabus (as shown in bold), and more stop-words (as shown with underline).

- IPAD with *Regeneration Comparator* outperforms IPAD with *Prompt-Text Consistency Verifier* by 3.78% on attacked data. As shown in Table 3 (rows 3-4) and Table 4 (rows 6-8), IPAD version 2 achieves the best F1 score and AUROC scores. To calculate the overall attacked dataset score, we calculate the F1 scores for Table 4: 94.82%, 95.35%, 95.31% for IPAD version 2, and 83.58%, 88.34%, and 94.70% for IPAD version 1. The average F1 score difference is thus (94.82% + 95.35% + 95.31% - 83.58% - 88.34% - 94.70% + 97.60% + 98.86% - 97.55% - 98.85%) / 5 = 3.78%.

C DPIC (decouple prompt and intrinsic characteristics) Prompt Extraction Zero-shot Prompts

"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>.".

D Linguistic Difference Examples

Figure 3 shows examples where HWT and LGT prompts with different sentence complexity. Figure 4 shows the results of analytical thinking level statistics. Figure 5 shows examples of using different personas and different analytical thinking levels.

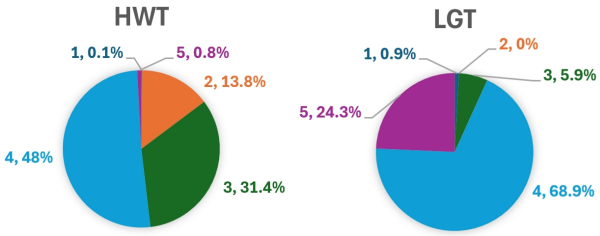

Figure 4: Comparison of different analytical thinking levels, with LGT has higher percentage of level 4 and level 5.

HWT Prompt	LGT Prompt
Write a persuasive essay to convince **your** school to allow students to bring their own phones to school and use them during lunchtime and other free periods. However, the school should be responsible for confiscating the phones if they are used during class or disrupt the learning environment.	Discuss the advantages of permitting students to bring phones to school for use during breaks, with the understanding that the school will confiscate phones if they are used in class or disrupt learning.
Explain your opinion on the Electoral College and its role in the election of the President of the United States, citing evidences to support your argument.	Discuss the advantages and disadvantages of relying on popular votes versus the Electoral College in the election of the president of the United States. **Consider** the **potential for errors** in vote counting, the **possibility** of a tie, and the **impact** on voter turnout. **Evaluate** the **fairness** of the Electoral College system and **propose** potential **solutions** to address any issues that arise.

Figure 5: Examples that use different persona usage (above), and different analytical thinking levels (below left has level 2, and below right has level 5, they are prompts generated by the same problem statements).

E User Study

Figure 6 7 and 8 shows the screenshots of online AI detectors. Figure 9 shows the questionnaire questions. Figure 10 shows the user guide.

E.1 Online AI Detectors Screenshots

E.2 Questionnaire questions

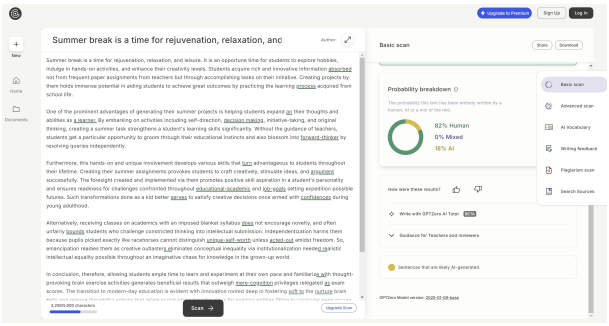

Figure 6: GPTZero Online Detector Screenshot

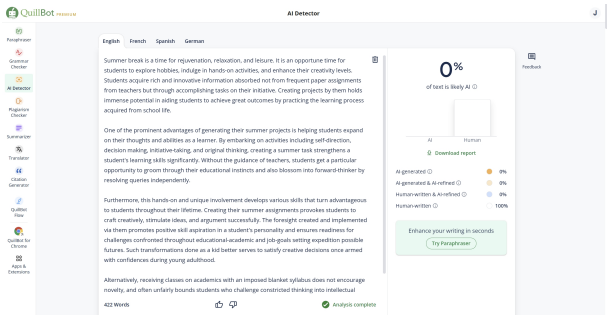

Figure 7: Quillbot Online Detector Screenshot

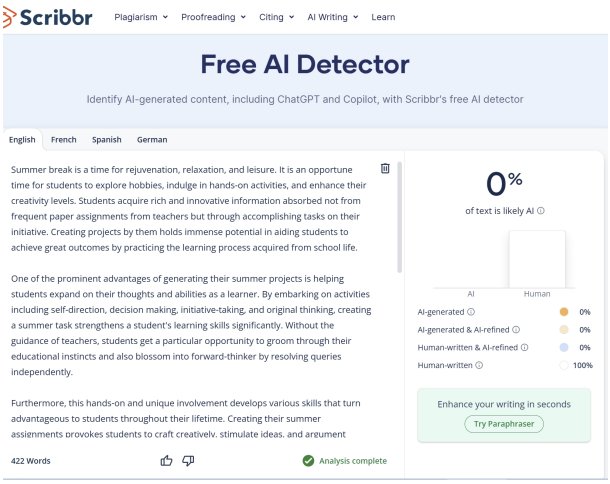

Figure 8: Scribbr Online Detector Screenshot

Aspects	Questions	Rates(1-5)
1 **Transparency, Scrutability, and Education**	🔹 **Do you think IPAD provides clearer and more understandable explanations for its decisions?**	
2 **Trust and Persuasiveness**	🔹 **Do you find IPAD's outputs more trustworthy and convincing?**	
3 **Satisfaction, Effectiveness, and Efficiency**	🔹 **Do you think IPAD is more effective and efficient in performing its detection tasks?**	
4 **Debugging and Error Handling**	🔹 **Does IPAD allow you to identify and correct its mistakes more easily?**	

Figure 9: Questionnaire questions

IPAD User Study Participant Guide

Thank you for participating in our research! The goal of this study is to evaluate the performance of our AI detection framework (IPAD) in various scenarios. Below are the important details you need to know:

1. Purpose of the Study

The purpose of this study is to evaluate how well the IPAD framework can distinguish between human-written and AI-generated texts. As a participant, you will be asked to provide feedback after reviewing the following contents..

2. Recruitment and Payment

You were invited to participate in this study through student recruitment. As a token of appreciation for your time and effort, you will receive a payment of **$5** upon completing the study. The payment will be made via online payment after you have completed all tasks.

3. Data Use and Consent

The data you provide (including any information you input during the study) will be used solely for the purpose of this research. All data will be anonymized, and any personally identifiable information will be removed. Your data will not be used for any commercial purposes.

Before starting, you will be asked to sign a consent form, confirming that you understand your data will be used for this study and that you voluntarily agree to participate. You are free to withdraw from the study at any time, and your decision to withdraw will not affect your payment or any other aspect of the study.

4. Ethics Review and Approval

This study has been approved by an Institutional Review Board (IRB), ensuring that all ethical guidelines are followed. We take your privacy and data security seriously, and all data collection procedures comply with strict privacy protection standards.

5. Voluntary Participation and Withdrawal

Your participation is entirely voluntary, and you may withdraw from the study at any time without any negative consequences. If you choose to withdraw, your data will no longer be used for analysis.

Figure 10: User Study User guide