
Zero-Shot Detection of LLM-Generated Text via Implicit Reward Model

Runheng Liu, Heyan Huang, Xingchen Xiao, Zhijing Wu*

School of Computer Science and Technology, Beijing Institute of Technology
{rhliu, hhy63, xc Xiao, zhijingwu}@bit.edu.cn

Abstract

Large language models (LLMs) have demonstrated remarkable capabilities across various tasks. However, their ability to generate human-like text has raised concerns about potential misuse. This underscores the need for reliable and effective methods to detect LLM-generated text. In this paper, we propose IRM, a novel zero-shot approach that leverages Implicit Reward Models for LLM-generated text detection. Such implicit reward models can be derived from publicly available instruction-tuned and base models. Previous reward-based method relies on preference construction and task-specific fine-tuning. In comparison, IRM requires neither preference collection nor additional training. We evaluate IRM on the DetectRL benchmark and demonstrate that IRM can achieve superior detection performance, outperforms existing zero-shot and supervised methods in LLM-generated text detection.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across various tasks [1, 2]. In particular, their ability to follow instructions and generate human-like content has made them essential tools in various real-world applications. However, this trend raises concerns about the misuse of LLMs, such as generating fake news [3] and plagiarism [4]. Since LLM-generated text can be indistinguishable from human-written text [5], it is necessary to develop effective methods for detecting such content.

Detection methods can generally be categorized into two paradigms: supervised and zero-shot approaches. Supervised methods typically train a classifier using both human-written and LLM-generated texts. However, such classifiers often fail to generalize well to texts generated by unseen LLMs [6, 7]. In contrast, zero-shot methods provide an alternative with more generalization ability and without the need for training [8]. These methods leverage various LLMs and exploit statistical classification metrics to assess how likely a text is to be generated by an LLM, such as the average log likelihood or the log-rank of each tokens within the detected text [9, 10, 5]. However, in real-world the source LLM of the detected text is often unknown, thus these zero-shot methods have to rely on a proxy LLM and suffer from a performance drop. This highlights the need to identify model-agnostic metrics for detection.

ReMoDetect [11] has recently proposed using the reward score—the output of a reward model that measures the extent to which a text aligns with human values—as a detection metric. The rationale is that powerful LLMs are typically trained to obtain high reward score, making this signal potentially model-agnostic. Specifically, ReMoDetect constructs a preference dataset from human and ChatGPT responses to the same prompt, and fine-tunes a pre-trained reward model for adaptation to the detection task. However, this fine-tuning inherits the drawbacks of supervised methods. As

*Corresponding author.

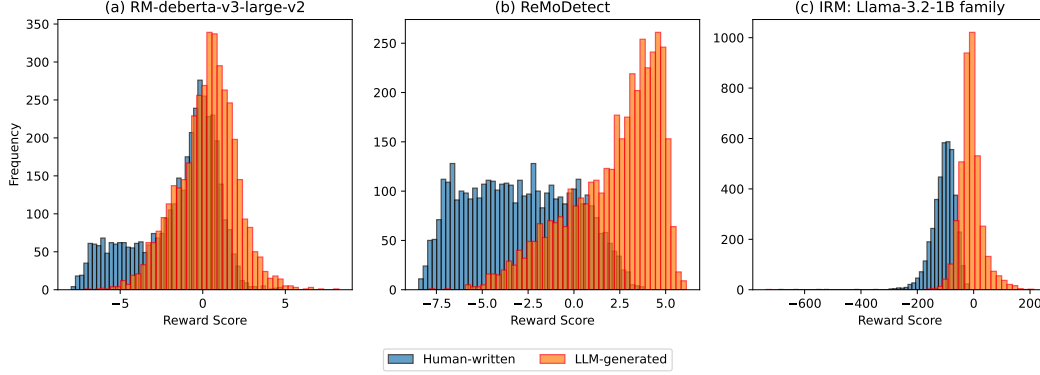


Figure 1: Distributions of reward scores for human-written and LLM-generated texts from the multi-LLM sub-task of the DetectRL benchmark. The source LLMs include GPT-3.5-turbo, PaLM-2-Bison, Claude-Instant, and LLaMA-2-70B. (a) shows distributions under RM-deberta-v3-large-v2, which is a pre-trained reward model. (b) shows distributions under ReMoDetect, which is initialized from RM-deberta-v3-large-v2 and fine-tuned on task-specific preference dataset. (c) illustrates distributions under an implicit reward model derived from Llama-3.2-1B family without additional training.

shown in Figure 1 (b), when ReMoDetect is evaluated on texts generated by unseen LLMs, there remains substantial overlap between the two score distributions of human-written and LLM-generated texts, suggesting the limited generalization ability.

In this paper, we propose IRM, a novel zero-shot method that leverages implicit reward models for LLM-generated text detection. Such implicit reward models can be derived from publicly available instruction-tuned and base models. Given a detected text y , an instruction-tuned model π_θ and a base model π_{ref} , the implicit reward score of y is formulated as $r(y) = \log \frac{\pi_\theta(y)}{\pi_{\text{ref}}(y)}$. This score serves as the detection metric in our method, where a higher score indicates that the input text is more likely to be generated by LLMs. This formulation is built on Direct Preference Optimization (DPO) [12], which provides theoretical support for parameterizing implicit reward models. Compared to ReMoDetect, IRM requires neither preference collection nor additional training. As shown in Figure 1 (c), the two distributions under IRM are more separable than those in (b), suggesting that our zero-shot metric transfers well to unseen LLMs.

We evaluate IRM on the DetectRL benchmark [13] and demonstrate that IRM is an effective zero-shot method for LLM-generated text detection. Notably, with Llama-3.2-1B family, IRM achieves an average score of 91.77%, surpassing previous zero-shot baselines such as Log-Likelihood (77.46%) and Log-Rank (77.42%), as well as Binoculars (87.67%), a method which also utilizes two LLMs during detection. Furthermore, IRM even outperforms the supervised reward-based method, ReMoDetect (85.86%). These results underscore IRM’s potential as a effective zero-shot detector for real-world applications.

2 Preliminaries

The mainstream alignment approach for LLMs is Reinforcement Learning from Human Feedback (RLHF) [14, 15], which relies on a reward model as a proxy for human preferences. In this section, we first introduce the concept of reward modeling, followed by an overview of the RLHF process.

Reward modeling. Typically, the reward model is initialized from a pre-trained language model with an additional linear head that outputs a scalar value. It is trained on a preference dataset, where human annotators compare multiple responses to the same prompt and rank them based on quality such as helpfulness or harmlessness. Formally, the preference is modeled using the Bradley-Terry (BT) model [16]:

$$p(y_1 \succ y_2 | x) = \frac{\exp(r(x, y_1))}{\exp(r(x, y_1)) + \exp(r(x, y_2))}, \quad (1)$$

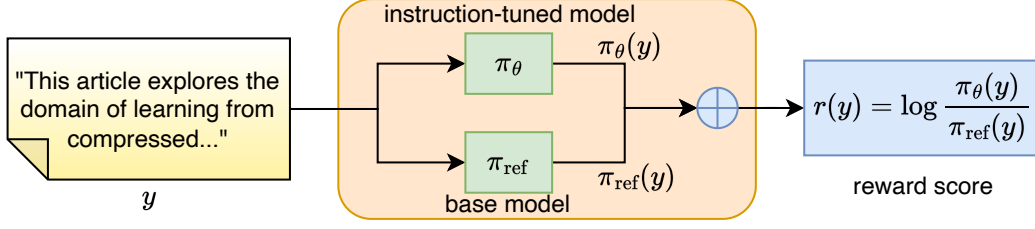


Figure 2: IRM leverages open-source instruction-tuned and base models to construct an implicit reward model. The resulting reward score is used as a detection metric, where a higher score indicates a higher probability that y is generated by an LLM.

where r denotes the reward model, x is the prompt, and y_1, y_2 are two candidate responses. Given a prompt–response pair (x, y) , the reward model outputs a scalar score $r(x, y)$, where a higher score indicates stronger alignment with human values.

RLHF. During RLHF, the learned reward model is used to provide feedback for optimizing the language model. The LLM is framed as a policy model π_θ , which generates responses conditioned on input prompts. The policy is optimized to maximize its expected reward while remaining close to a reference model π_{ref} , typically the original base LLM prior to alignment. This leads to the following optimization objective:

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} [r(x, y)] - \beta \mathbb{D}_{\text{KL}}[\pi_\theta(y|x) || \pi_{\text{ref}}(y|x)], \quad (2)$$

where x denotes the prompt, y the response, r the reward model, and β a hyper-parameter controlling the strength of KL regularization.

3 Method

We introduce IRM, a zero-shot detection method for LLM-generated text detection. We formalize the approach and show how it can be instantiated using publicly available instruction-tuned and base models.

Implicit reward model for detection. Direct Preference Optimization (DPO) [12] offers a closed-form solution to the RLHF objective in Eq. 2:

$$\pi_\theta^*(y|x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right), \quad (3)$$

where $Z(x) = \sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)$ is the partition function that normalizes the distribution. By rearranging this equation, we can express the reward as a function of the policy and reference models, yielding an implicit reward model:

$$r(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x). \quad (4)$$

For simplicity, we let $x = \emptyset$ and y be the detected text. In this case, the term $\beta \log Z(x)$ becomes a constant across all detected texts. Importantly, applying a linear transformation to the reward scores does not affect commonly used classification metrics such as AUROC and F_1 , as long as the threshold is adjusted accordingly. As a result, we define the detection metric as:

$$r(y) = \log \frac{\pi_\theta(y)}{\pi_{\text{ref}}(y)} = \sum_{i=1}^L \log \frac{\pi_\theta(y_i | y_{<i})}{\pi_{\text{ref}}(y_i | y_{<i})}, \quad (5)$$

where L is the length of y and y_i denotes the i -th token. DPO also provides a theoretical justification for the effectiveness of the implicit reward model. As shown in Eq. 4, given a reward model $r(x, y)$

and a reference model π_{ref} , one can analytically derive the optimal policy model π_{θ}^* under the RLHF objective. Conversely, given a learned policy model π_{θ} and a reference model π_{ref} , one can recover a corresponding reward model such that π_{θ} is optimal with respect to that reward under the same reference model. This bidirectional relationship ensures that the implicit reward model is consistent with the behavior of the policy model relative to the reference model. Therefore, if the policy model aligns well with human preferences, the derived implicit reward model is also expected to exhibit strong performance in tasks such as LLM-generated text detection.

Constructing IRM using instruction-tuned and base models. Typically, publicly available LLMs release both instruction-tuned model and its corresponding base model. We leverage the instruction-tuned model as the policy model and the base model as the reference model. In this way, without any fine-tuning, we can obtain an implicit reward model for LLM-generated text detection. Given a detected text y , its reward score can be computed via Eq. 5. Moreover, the same formulation naturally extends to LLMs trained via iterative alignment. Suppose instruction model π_T is obtained from the base model π_0 through T rounds of optimization: $\pi_0 \rightarrow \pi_1 \rightarrow \dots \rightarrow \pi_{T-1} \rightarrow \pi_T$. While the intermediate models $\pi_1, \dots, \pi_{T-1}$ are typically not publicly released, we can still conceptually define the implicit reward at each step i by applying Eq. 4:

$$r_i(x, y) = \beta \log \frac{\pi_{i+1}(y|x)}{\pi_i(y|x)} + \beta \log Z_i(x). \quad (6)$$

By summing the rewards over all iterations, we obtain the total reward:

$$\sum_{i=1}^T r_i(x, y) = \beta \log \frac{\pi_T(y|x)}{\pi_0(y|x)} + \beta \sum_{i=1}^T Z_i(x). \quad (7)$$

Since $x = \emptyset$, the term $\beta \sum_{i=1}^T Z_i(x)$ becomes a constant. As linear transformation does not affect the ranking of detected texts, the overall reward is determined by $\log \frac{\pi_T(y)}{\pi_0(y)}$. Therefore, given only the instruction-tuned and base models, we can construct an implicit reward model for detection without requiring access to intermediate checkpoints.

4 Experiments

4.1 Settings

Datasets. We conduct evaluations on a large benchmark, DetectRL [13], which covers various domains, multiple LLMs and diverse attacks scenarios in real-world settings. We follow the test setting of DetectRL in the evaluation of detection methods. The statistics of the benchmark are reported in Table 6 in Appendix A.1. Notably, the texts in DetectRL are mostly complete, unlike previous works where the texts are often truncated to prefixes of complete texts. This allows for a more realistic evaluation of detection methods, as texts are typically not truncated in real-world scenarios.

Models. We employ Gemma-2 [17], Llama-3.2, Gemma [18], Qwen-2 [19], and Qwen-2.5 [20] model families for implementation. These model families provide both publicly available base models and instruction-tuned models, which are essential for deriving the implicit reward model. Within these families, we focus on lightweight LLMs, such as Gemma-2-2B-it, and Llama-3.2-1B-Instruct, to ensure computational efficiency while maintaining competitive performance.

Baselines. We compare IRM with recent zero-shot methods, including Binoculars [21] and Lastde [22], as well as typical zero-shot baselines such as Log-Likelihood [9, 10], Rank [5], Log-Rank [5], LRR [23] and Fast-DetectGPT [24]. Since we employ reward score as detection metrics, we also include ReMoDetect [11] and its initialization, RM-Deberta-v3-large-v2, as well as reward models from RewardBench [25], such as GRM [26] as baselines. For practical consideration, we exclude methods that require additional LLMs to perturb the original text for metric computation, as the extra cost is too high in real-world settings.

Metrics. Follow the settings of DetectRL, we report AUROC and F_1 Score as the main evaluation metrics. AUROC is widely used for assessing zero-shot detection methods and F_1 Score provides a comprehensive evaluation of detector capabilities by balancing the Precision and Recall.

Table 1: The overall performance of zero-shot detection methods on the DetectRL benchmark, covering robustness and generalization tasks, which include various domains, LLMs and attacks scenarios. It also considers the effects of text length and real-world human writing characteristics. The reported average score is computed across all tasks. For each task, the best results are marked in **bold** and the second best results are marked by underline. More detailed detection results are available in Table 8, 9, 10 and 11 in Appendix A.2.

Tasks Settings →	Multi-Domain		Multi-LLM		Multi-Attack		Generalization			Length		Human Writing		Avg.
Methods ↓	AUROC	F ₁	AUROC	F ₁	AUROC	F ₁	Domain	LLM	Attack	Train	Test	AUROC	F ₁	
<i>Gemma-2-2B-it</i>														
Log-Likelihood	74.75	68.75	74.65	62.81	77.92	70.65	65.76	63.07	64.79	<u>69.78</u>	<u>69.70</u>	92.78	87.54	72.53
Rank	53.56	45.45	53.45	45.64	57.62	46.14	44.08	43.78	36.86	44.92	44.93	79.05	71.77	51.33
Log-Rank	75.78	69.81	75.62	65.36	78.75	71.37	66.79	64.85	66.30	69.50	68.79	<u>93.02</u>	<u>88.05</u>	73.38
LRR	77.43	70.81	76.81	66.62	79.28	71.32	68.20	66.05	68.30	65.75	63.44	91.41	85.75	73.17
Fast-DetectGPT	70.26	63.62	69.51	59.85	73.30	66.02	62.42	59.94	62.63	56.60	61.66	89.56	83.61	67.61
Lastde	60.62	55.83	59.70	54.91	66.11	56.31	54.45	53.61	45.29	47.25	53.76	90.64	85.63	60.32
<i>Gemma-2-2B family (Gemma-2-2B-it+Gemma-2-2B)</i>														
Binoculars	<u>80.85</u>	<u>74.57</u>	<u>80.43</u>	<u>71.24</u>	<u>83.00</u>	<u>76.58</u>	<u>73.11</u>	<u>70.95</u>	<u>73.64</u>	73.03	71.71	93.29	89.07	77.81
IRM	84.65	77.70	86.56	77.43	88.38	79.02	76.97	76.62	78.64	60.30	58.15	81.19	76.04	<u>77.05</u>
<i>Llama-3.2-1B-Instruct</i>														
Log-Likelihood	79.40	73.49	79.67	71.34	82.65	76.31	72.29	70.84	73.32	72.87	70.98	94.02	89.85	77.46
Rank	59.65	53.70	59.50	50.84	63.60	52.35	51.02	49.52	41.91	51.20	49.12	84.37	78.20	57.31
Log-Rank	79.64	73.41	79.78	71.50	82.55	75.37	72.05	70.54	73.32	73.08	71.46	93.94	89.81	77.42
LRR	77.99	70.66	77.79	69.33	79.56	70.65	69.25	68.28	68.38	69.97	68.05	91.62	86.72	74.48
Fast-DetectGPT	68.04	58.44	68.62	58.72	69.58	62.47	56.76	58.40	60.90	45.94	55.84	60.83	53.12	59.82
Lastde	65.11	56.03	59.70	54.91	69.59	69.42	53.40	53.14	59.47	41.59	49.79	87.52	82.00	61.67
<i>Llama-3.2-1B family (Llama-3.2-1B-Instruct+Llama-3.2-1B)</i>														
Binoculars	92.48	<u>85.63</u>	92.08	86.67	93.04	<u>87.02</u>	85.07	83.18	85.60	80.62	81.88	94.63	91.82	87.67
IRM	97.97	93.75	97.24	92.12	97.19	92.01	90.23	90.72	90.87	82.34	83.34	<u>94.48</u>	<u>90.78</u>	91.77

Implementation Details. All experiments are conducted on two NVIDIA RTX 4090 GPUs (24GB each). All model and datasets used in this paper are fully detailed and referenced in Table 7 in Appendix A.1.

4.2 Main Results

4.2.1 Zero-shot LLM-generated Text Detection

In Table 1, we present the overall performance of IRM and other zero-shot detection methods on the DetectRL benchmark. Detailed results for each sub-task are reported in Table 8, 9, 10 and 11 in Appendix A.2. With the Llama-3.2-1B family, IRM achieves the highest average score, improving the best result by 4.1%, from 87.67% to 91.77%. Using the Gemma-2-2B family, IRM also attains a competitive average score of 77.05%, slightly below the best result of 77.81%. All methods, except Fast-DetectGPT, perform better when using the Llama-3.2-1B family than with the Gemma-2-2B family, indicating that Llama-3.2-1B family is more suitable for these zero-shot detection approaches.

4.2.2 Reward-based LLM-generated Text Detection

In Table 2, we present the overall performance of IRM and other reward models on the DetectRL benchmark. With the Llama-3.2-1B family, IRM achieves the highest average score, improving the best result by 5.8%, from 85.86% to 91.77%. Notably, IRM outperforms ReMoDetect, a supervised reward model that fine-tunes a pre-trained reward model (RM-deberta-v3-large-v2) using task-specific data. In addition, IRM also outperforms competitive reward models from the RewardBench benchmark, such as GRM-Llama3.2-3B and GRM-gemma2-2B, when evaluated on DetectRL benchmark. This suggests a potential mismatch between reward models optimized for preference classification and the requirements of LLM-generated text detection, indicating that directly leveraging such models for detection may be suboptimal.

4.3 Analysis

Robustness to domains. As shown in Table 8, IRM outperforms other detection methods on 6 out of 8 evaluation metrics and achieves the best average performance across all domains. While its performance varies slightly across domains—for instance, achieving particularly strong results on the arXiv dataset and relatively lower scores on the Review dataset—it maintains overall superiority throughout. For example, using the Llama-3.2-1B family, IRM achieves an AUROC of 98.56% and

Table 2: The overall performance of reward-based detection methods on the DetectRL benchmark, covering robustness and generalization tasks, which include various domains, LLMs and attacks scenarios. It also considers the effects of text length and real-world human writing characteristics. The reported average score is computed across all tasks. For each task, the best results are marked in **bold** and the second best results are marked by underline. More detailed detection results are available in Table 8, 9, 10 and 11 in Appendix A.2.

Tasks Settings →	Multi-Domain		Multi-LLM		Multi-Attack		Generalization			Length		Human Writing		Avg.
Reward Models ↓	AUROC	F_1	AUROC	F_1	AUROC	F_1	Domain F_1	LLM F_1	Attack F_1	Train F_1	Test F_1	AUROC	F_1	
ReMoDetect	<u>91.99</u>	<u>85.95</u>	<u>90.41</u>	<u>83.17</u>	<u>91.22</u>	<u>82.09</u>	<u>79.08</u>	<u>78.36</u>	<u>80.22</u>	88.53	85.87	<u>91.63</u>	<u>87.65</u>	<u>85.86</u>
RM-Deberta-v3-large-v2	73.08	65.01	70.37	64.91	71.34	63.44	56.70	62.28	61.21	79.68	72.71	81.48	74.62	68.99
GRM-gemma2-2B	51.07	31.20	50.57	45.32	50.69	59.97	27.53	41.58	56.30	45.71	53.97	63.69	60.37	49.07
GRM-Llama3.2-3B	45.47	21.27	46.15	22.85	48.56	30.80	19.12	19.63	29.15	46.81	49.99	60.71	54.64	38.09
<i>IRM</i>														
Gemma-2-2B family	84.65	77.70	86.56	77.43	88.38	79.02	76.97	76.62	78.64	60.30	58.15	81.19	76.04	77.05
Llama-3.2-1B family	97.97	93.75	97.24	92.12	97.19	92.01	90.23	90.72	90.87	<u>82.34</u>	<u>83.34</u>	94.48	90.78	91.77

Table 3: The performance of IRM using the Llama-3.2-1B family on various attacks to human-written texts.

DIPPER Paraphrase				Polish using LLMs				Back Translation			
Pre	Rec	F_1	AUROC	Pre	Rec	F_1	AUROC	Pre	Rec	F_1	AUROC
93.02	83.33	87.91	94.39	77.76	79.76	78.75	85.36	93.11	87.10	90.01	95.54

an F_1 Score of 95.02% on the arXiv dataset, and an AUROC of 97.03% and an F_1 Score of 91.65% on the Review dataset. One possible explanation is that positive reviews are tend to be more neutral or benign compared to negative reviews, making them more likely to receive higher reward scores. This subtle property of the data may slightly reduce the separation margin between human-written and LLM-generated texts. Still, IRM maintains strong performance overall, underscoring its effectiveness across diverse domains.

Robustness to LLMs. As shown in Table 9, IRM outperforms other detection methods on 7 out of 8 evaluation metrics and achieves the best average performance across all LLMs. While IRM demonstrates strong generalization across a wide range of generation models, its performance can be further enhanced when the instruction-tuned and base models used to construct the implicit reward model share a similar model-family background with the generation model. For example, when using the Gemma-2-2B family, IRM performs particularly well in detecting texts generated by PaLM-2-bison, achieving an AUROC of 88.65% and an F_1 Score of 79.41%. Similarly, with the Llama-3.2-1B family, IRM demonstrates outstanding performance in detecting Llama-2-70B’s outputs, achieving an AUROC of 99.35% and an F_1 Score of 96.58%.

Robustness to attacks. As shown in Table 10, IRM outperforms other detection methods on 8 out of 10 evaluation metrics and achieves the highest average performance across all attacks applied to LLM-generated texts. For example, when using Llama-3.2-1B family, IRM achieves an average AUROC of 97.19% and an average F_1 Score of 92.01%. While IRM demonstrates strong robustness against attacks on LLM-generated texts, its performance is relatively lower but still competitive when attacks are applied to human-written texts, as shown in Table 11. For example, with the Llama-3.2-1B family, IRM achieves an average AUROC of 94.48% and an average F_1 Score of 90.78%, which are slightly below than the best results: 94.63% for AUROC and 91.82% for F_1 Score. We further investigate which attack type has the greatest impact on IRM’s performance. As shown in Table 3, with the Llama-3.2-1B family, IRM is particularly affected by the polishing attack, where LLMs are prompted to improve human-written texts. This result is intuitive: since the polished texts are generated by LLMs and typically exhibit improved fluency or coherence, they tend to receive higher reward scores than their original, unpolished counterparts. As a result, the distinction between human-written and LLM-generated texts become less clear, leading to a moderate drop in detection performance for IRM.

Generalization of IRM. In the generalization task, following the DetectRL benchmark settings, we evaluate each sub-task using the optimal threshold from other sub-tasks and calculate the average F_1 Score across all sub-tasks. As shown in Table 1, IRM exhibits strong generalization capability

Table 4: The performance comparison of baseline zero-shot methods using instruction-tuned and base models. Gemma-2-2B-it and Llama-3.2-1B-Instruct are instruction-tuned models, while Gemma-2-2B and Llama-3.2-1B are their corresponding base models.

Model	Log-Likelihood	Rank	Log-Rank	LRR	Fast-DetectGPT	Lastde
Llama-3.2-1B-Instruct	77.46	57.31	77.42	74.48	59.82	61.67
Llama-3.2-1B	66.44	50.80	66.81	65.78	63.80	59.13
Δ	-11.02	-6.51	-10.61	-8.70	+3.98	-2.54
Gemma-2-2B-it	72.53	51.33	73.38	73.17	67.61	60.32
Gemma-2-2B	65.74	47.23	67.24	69.55	63.08	59.25
Δ	-6.79	-4.10	-6.14	-3.62	-4.53	-1.07

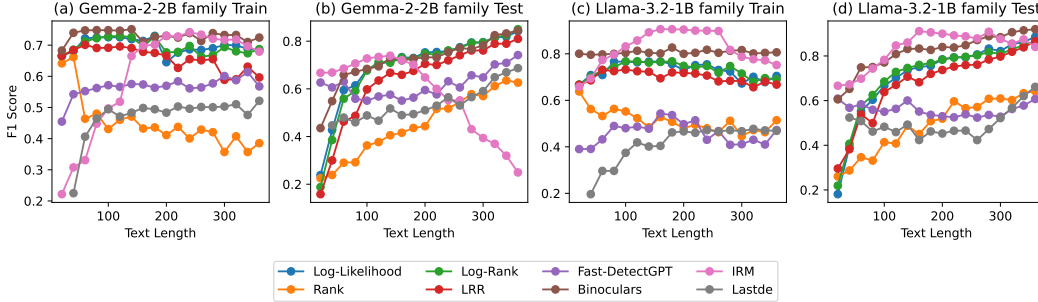


Figure 3: The performance of various zero-shot detection methods across different text lengths during training-time and test-time.

across different domains, LLMs and attacks. For instance, when using Llama-3.2-1B family, IRM achieves F_1 Score of 90.23%, 90.72% and 90.87% on domain, LLM and attack generalization tasks. These results suggest that the decision boundary of the implicit reward score is stable across various conditions, highlighting IRM’s superior generalization ability.

Effectiveness of instruction-tuned models for detection methods. As prior works often utilize base models when implementing detection methods, we follow this practice and implement baseline methods using both Llama-3.2-1B and Gemma-2-2B. As shown in Figure 4, using instruction-tuned models consistently improves the performance of 5 out of 6 baseline methods. For instance, when using Llama-3.2-1B-Instruct, the performance of Log-Likelihood improves from 66.44% to 77.46%. Similarly, with Gemma-2-2B-it, the performance increases from 65.74% to 72.53%. These results suggest that instruction-tuned models are generally more effective for implementing detection methods.

Impact of text length. In the length task, following the DetectRL benchmark settings, we analyze the impact of text length on IRM’s performance. For the train sub-task, we use the optimal threshold derived from the dataset with a length interval of 160-180 and apply it to datasets with other length intervals. Conversely, for the test sub-task, the optimal threshold of each dataset is evaluated on the dataset with a 160-180 length interval. We refer to the dataset used to derive the threshold as the training dataset, and the dataset used to evaluate the threshold as the test dataset. As shown in Table 1, when using Llama-3.2-1B family, IRM achieves the highest F_1 Score of 82.34% and 83.34% on the train and test sub-tasks, respectively. However, when using Gemma-2-2B family, IRM’s performance drops significantly compared to other methods. To further investigate the influence of text length, we present the detailed results of train and test sub-tasks in Figure 3. IRM exhibits a distinct performance pattern trend compared to the other methods. Specifically, IRM tends to perform better than the length interval of the training dataset is similar to that of the test dataset. For example, using Gemma-2-2B family, thresholds derived from texts with lengths of 180-260 words perform better on 160-180-word texts (Figure 3 (a)), while the threshold from texts with 160-180-word is more effective for texts of 80-140 words (Figure 3 (b)). Similarly, using Llama-3.2-1B family, thresholds derived from texts of 140-260 words perform better on 160-180-word texts (Figure 3 (c)), whereas the threshold from 160-180-word texts performs better on texts of 140-200 words (Figure 3 (d)). This phenomenon may

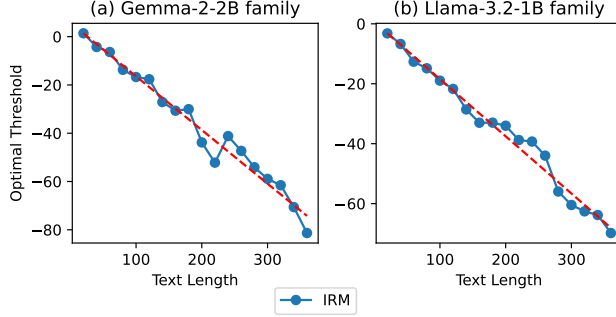


Figure 4: Optimal classification thresholds of IRM across datasets with varying text length intervals. The solid line shows the threshold values for each length interval (from 0 to 360 words), and the red dashed line represents a linear fit to illustrate the overall trend.

Table 5: The performance of IRM with different model versions and sizes.

Tasks Settings →	Multi-Domain		Multi-LLM		Multi-Attack		Generalization			Length		Human Writing		Avg.
Model Family ↓	AUROC	F_1	AUROC	F_1	AUROC	F_1	Domain F_1	LLM F_1	Attack F_1	Train F_1	Test F_1	AUROC	F_1	
Qwen2-0.5B	90.14	83.57	89.43	82.54	90.47	83.55	82.48	81.59	83.34	74.17	70.67	86.11	78.27	82.79
Qwen2-1.5B	89.68	82.49	89.64	82.59	89.44	81.59	81.80	81.82	81.33	76.21	74.52	85.22	77.03	82.57
Qwen2.5-0.5B	79.44	76.28	77.84	74.02	78.48	73.55	74.75	73.35	73.03	66.80	62.30	74.18	68.10	73.24
Qwen2.5-1.5B	76.22	73.51	74.75	69.68	74.70	68.93	69.11	68.56	68.33	65.54	63.17	70.55	62.86	69.69
Gemma-2B	91.84	84.06	92.08	84.77	93.17	85.48	83.19	82.18	85.13	58.06	68.00	92.50	87.90	83.72
Gemma-2-2B	84.65	77.70	86.56	77.43	88.38	79.02	76.97	76.62	78.64	60.30	58.15	81.19	76.04	77.05
Llama-3.2-1B	97.97	93.75	97.24	92.12	97.19	92.01	90.23	90.72	90.87	82.34	83.34	94.48	90.78	91.77
Llama-3.2-3B	97.31	92.51	97.25	92.15	97.21	91.69	91.68	90.67	91.37	78.59	80.97	93.85	88.85	91.08

be attributed to the way the implicit reward score is computed. As shown in Eq. 5, the reward is accumulated over all tokens, so the total reward varies with text length. Consequently, the reward distribution shifts across different length intervals, which affects the optimal threshold value. This trend is illustrated in Figure 4, where the optimal threshold of IRM changes with text length, showing an approximately linear relationship. Notably, for the Gemma-2-2B family, the optimal threshold fluctuates and deviates from the fitted red dashed line within the 160–220-word intervals. Besides, we can also observe that, with Gemma-2-2B family, the optimal threshold of IRM fluctuates and deviates from the red dash within 160-220-words intervals, which can explain the performance drop of IRM when using Gemma-2-2B family. Since this length range corresponds to the train and test dataset in the length task, such instability in threshold values may explains the observed performance drop of IRM when using the Gemma-2-2B family in this task.

Impact of model version and size. We further investigate the impact of model version and size on the performance of IRM by implementing it across various model families, as presented in Table 5. When using the Llama-3.2-1B family, IRM achieves the best average score of 91.77%, slightly higher than the 91.08% achieved with the larger Llama-3.2-3B family. A similar trend is observed in the Qwen2 and Qwen2.5 families: for example, IRM attains 73.24% with Qwen2.5-0.5B family, outperforming Qwen2.5-1.5B family (69.69%). Besides, IRM achieves 83.72% with the Gemma-2B family, compared to 77.05% with Gemma-2-2B family. These results suggest that larger or newer versions of a model do not always lead to better detection performance.

5 Related Works

LLM-generated text detection. Since generative language models have shown remarkable capability in text generation, detecting AI-generated texts has attracted significant research attention. Current research primarily focus on supervised methods and zero-shot methods for LLMs. Supervised methods involve collecting labeled data and training classifiers for distinguishing human-written and machine-generated texts [27–29, 11]. However, it has been shown that many supervised methods often suffer from overfitting to the training set [6, 7], limiting their generalization ability. In contrast, zero-shot methods have shown better generalization ability, making them a promising alternative. Current zero-shot methods are based on pretrained LLMs and utilize heuristic metrics to classify

LLM-generated texts and human-written texts. These metrics include likelihood [9, 10], entropy, rank, log-rank [5], probability curvature [8, 24, 30], etc. Additionally, perturbing texts and measuring the variation of metrics is a commonly used augmentation method for constructing better metrics [22, 31–33], but it incurs extra computational costs. Our method is a zero-shot method that leverages the characteristics of aligned LLMs and derives a metric from LLMs in an off-the-shelf manner, without the need of additional training. Additionally, our method does not require perturbing texts to compute the variation of metrics, avoiding extra computational costs.

Implicit reward model. In LLM alignment, the primary focus is on obtaining a well-aligned LLM, rather than a high-quality reward model. Thus, although DPO and other methods use implicit reward models to implement LLM alignment, the implicit reward models obtained after training are typically not reused. Recently, some research has leveraged implicit reward models to label process rewards for reasoning tasks [34]. Since our method uses the reward score as a metric for distinguishing human-written and LLM-generated texts, we propose leveraging implicit reward models to compute the reward score for texts.

6 Discussion and Conclusion

In this paper, we propose IRM, a novel method for zero-shot detection of LLM-generated text. IRM utilizes implicit reward models to compute reward scores as detection metrics. Experiments on the DetectRL benchmark demonstrate that IRM is an effective zero-shot method for LLM-generated text detection. Notably, with Llama-3.2-1B family, IRM achieves an average score of 91.77%, surpassing previous zero-shot baselines such as Log-Likelihood, Log-Rank and Binoculars. Furthermore, IRM even outperforms the supervised baseline ReMoDetect, showing that IRM remains highly effective without any task-specific adaptation. These results underscore IRM’s potential as a effective zero-shot detector for real-world applications.

Limitations and future works. In our experiments, we focus exclusively on lightweight LLMs, excluding larger LLMs due to computational constraints. Similarly, we do not consider generative reward models as baselines, as they typically require larger LLMs and more computation resources. Besides, we restrict our analysis to deriving IRM from LLMs within the same model family. For cases where only instruction-tuned models are available, deriving corresponding IRM is not feasible. We leave how to effectively leverage LLMs from different model families for future work.

Societal impact. IRM contributes to the responsible use of LLMs by providing a zero-shot method for detecting LLM-generated text. This capability can enhance transparency and help mitigate misuse in high-stakes domains such as education and media. On the other hand, misclassifications may inadvertently suppress human-generated text, highlighting that need for continuous development more powerful detection methods.

Acknowledgments and Disclosure of Funding

We thank all the anonymous reviewers for their insightful and valuable feedback on this paper. This work is supported by the National Natural Science Foundation of China (Grant No.U21B2009).

References

- [1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL <https://arxiv.org/abs/2005.14165>.
- [2] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao,

Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondrasiuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rameez Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>.

- [3] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. Advances in neural information processing systems, 32, 2019.
- [4] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kudipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa

- Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models, 2022. URL <https://arxiv.org/abs/2108.07258>.
- [5] Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. GLTR: Statistical detection and visualization of generated text. In Marta R. Costa-jussà and Enrique Alfonseca, editors, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 111–116, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-3019. URL <https://aclanthology.org/P19-3019/>.
 - [6] Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc’Aurelio Ranzato, and Arthur Szlam. Real or fake? learning to discriminate machine from human generated text, 2019. URL <https://arxiv.org/abs/1906.03351>.
 - [7] Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship attribution for neural text generation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 8384–8395, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.673. URL <https://aclanthology.org/2020.emnlp-main.673/>.
 - [8] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. Detectgpt: zero-shot machine-generated text detection using probability curvature. In Proceedings of the 40th International Conference on Machine Learning, ICML’23. JMLR.org, 2023.
 - [9] Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. Unifying human and statistical evaluation for natural language generation. In Jill Burstein, Christy Doran, and Tamar Solorio, editors, 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 1689–1701, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1169. URL <https://aclanthology.org/N19-1169/>.
 - [10] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. Release strategies and the social impacts of language models, 2019. URL <https://arxiv.org/abs/1908.09203>.
 - [11] Hyunseok Lee, Jihoon Tack, and Jinwoo Shin. Remodetect: Reward models recognize aligned LLM’s generations. In The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024. URL <https://openreview.net/forum?id=pW9Jwim918>.
 - [12] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL <https://openreview.net/forum?id=HPuSIXJaa9>.
 - [13] Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. DetectRL: Benchmarking LLM-generated text detection in real-world scenarios. In The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2024. URL <https://openreview.net/forum?id=ZGMkOikEyv>.
 - [14] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. Advances in neural information processing systems, 33:3008–3021, 2020.

- [15] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [16] Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324, 1952.
- [17] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Matteo Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshtir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL <https://arxiv.org/abs/2408.00118>.
- [18] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikula, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech

- Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. Gemma: Open models based on gemini research and technology, 2024. URL <https://arxiv.org/abs/2403.08295>.
- [19] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL <https://arxiv.org/abs/2407.10671>.
- [20] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.
- [21] Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text. In Forty-first International Conference on Machine Learning, 2024. URL <https://openreview.net/forum?id=axl3FAkpik>.
- [22] Yihuai Xu, Yongwei Wang, Yifei Bi, Huangsen Cao, Zhouhan Lin, Yu Zhao, and Fei Wu. Training-free LLM-generated text detection by mining token probability sequences. In The Thirteenth International Conference on Learning Representations, 2025. URL <https://openreview.net/forum?id=vo4AHjowKi>.
- [23] Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Findings of the Association for Computational Linguistics: EMNLP 2023, pages 12395–12412, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.827. URL <https://aclanthology.org/2023.findings-emnlp.827/>.
- [24] Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-detectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In The Twelfth International Conference on Learning Representations, 2024. URL <https://openreview.net/forum?id=Bpcgcr8E8Z>.
- [25] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024. URL <https://arxiv.org/abs/2403.13787>.
- [26] Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for LLMs. In The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024. URL <https://openreview.net/forum?id=jwh9MHEfmY>.
- [27] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. RADAR: Robust AI-text detection via adversarial learning. In Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL <https://openreview.net/forum?id=QGrkbaan79>.

- [28] Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. Ghostbuster: Detecting text ghost-written by large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 1702–1717, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.95. URL <https://aclanthology.org/2024.naacl-long.95/>.
- [29] Yang Xu, Yu Wang, Hao An, Zhichen Liu, and Yongyuan Li. Detecting subtle differences between human and model languages using spectrum of relative likelihood. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 10108–10121, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.564. URL <https://aclanthology.org/2024.emnlp-main.564/>.
- [30] Guangsheng Bao, Yanbin Zhao, Juncal He, and Yue Zhang. Glimpse: Enabling white-box methods to use proprietary models for zero-shot LLM-generated text detection. In The Thirteenth International Conference on Learning Representations, 2025. URL <https://openreview.net/forum?id=an3fugFA23>.
- [31] Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. DNA-GPT: Divergent n-gram analysis for training-free detection of GPT-generated text. In The Twelfth International Conference on Learning Representations, 2024. URL <https://openreview.net/forum?id=Xlayxj2fWp>.
- [32] Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. Raidar: generative AI detection via rewriting. In The Twelfth International Conference on Learning Representations, 2024. URL <https://openreview.net/forum?id=bQWE2UqXmf>.
- [33] Shixuan Ma and Quan Wang. Zero-shot detection of LLM-generated text using token cohesiveness. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 17538–17553, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.971. URL <https://aclanthology.org/2024.emnlp-main.971/>.
- [34] Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels, 2024. URL <https://arxiv.org/abs/2412.01981>.
- [35] Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12463–12492, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.674. URL <https://aclanthology.org/2024.acl-long.674/>.
- [36] Zhihui Chen, Kai He, Yucheng Huang, Yunxiao Zhu, and Mengling Feng. Divscore: Zero-shot detection of llm-generated text in specialized domains, 2025. URL <https://arxiv.org/abs/2506.06705>.
- [37] Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiakai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-reward-v2: Scaling preference data curation via human-ai synergy, 2025. URL <https://arxiv.org/abs/2507.01352>.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: See Abstract and section 1.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: See Section 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification:

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: See section 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide codes and data with instruction files in supplemental materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: See appendix A.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Our method is deterministic and does not involve stochastic components such as random initialization, sampling, or data splits. Therefore, statistical significance measures or error bars are not applicable.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: See section 4.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We confirm our paper with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: See section 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not release any data or models in this paper, and thus no specific safeguards are required.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have clearly cite all the existing assets we used.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We do not release new asset.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: None.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: None.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Our proposed method IRM leverages publicly available LLMs, including instruction-tuned and base models, to construct an implicit reward model for LLM-generated text detection. These models are integral to the core methodology and are clearly described in the main text.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Details of Main Experiments

A.1 Details of Models and Datasets

Table 6: Statistic of DetectRL benchmark.

Setting	Sub Setting	Size
Multi-Domain	Academic	2008
	News	2008
	Creative	2008
	Social Media	2008
Multi-LLM	GPT-3.5-turbo	2008
	Claude-instant	2008
	PaLM-2-bison	2008
	Llama-2-70b	2008
Multi-Attack	Direct	2016
	Prompt	2032
	Paraphrase	2016
	Perturbation	2016
	Data Mixing	2008
Varing Text Length	-	16200
Human Writing	Direct	2016
	Paraphrase	2016
	Perturbation	2016
	Data Mixing	2012

Table 7: Dataset and model details.

Name Used	Full Name	Author	Source
DetectRL	DetectRL	[13]	GitHub
Llama-3.2-3B-Instruct	meta-llama/Llama-3.2-3B-Instruct	meta	HuggingFace
Llama-3.2-3B	meta-llama/Llama-3.2-3B	meta	HuggingFace
Llama-3.2-1B-Instruct	meta-llama/Llama-3.2-1B-Instruct	meta	HuggingFace
Llama-3.2-1B	meta-llama/Llama-3.2-1B	meta	HuggingFace
Gemma-2-2B-it	google/gemma-2-2b-it	[17]	HuggingFace
Gemma-2-2B	google/gemma-2-2b	[17]	HuggingFace
Gemma-2B-it	google/gemma-2-2b-it	[18]	HuggingFace
Gemma-2B	google/gemma-2-2b	[18]	HuggingFace
Qwen2.5-1.5B-Instruct	Qwen/Qwen2.5-1.5B-Instruct	[20]	HuggingFace
Qwen2.5-1.5B	Qwen/Qwen2.5-1.5B	[20]	HuggingFace
Qwen2.5-0.5B-Instruct	Qwen/Qwen2.5-0.5B-Instruct	[20]	HuggingFace
Qwen2.5-0.5B	Qwen/Qwen2.5-0.5B	[20]	HuggingFace
Qwen2-1.5B-Instruct	Qwen/Qwen2-1.5B-Instruct	[19]	HuggingFace
Qwen2-1.5B	Qwen/Qwen2-1.5B	[19]	HuggingFace
Qwen2-0.5B-Instruct	Qwen/Qwen2-0.5B-Instruct	[19]	HuggingFace
Qwen2-0.5B	Qwen/Qwen2-0.5B	[19]	HuggingFace
RM-Deberta-v3-large-v2	OpenAssistant/reward-model-deberta-v3-large-v2	OpenAssistant	HuggingFace
ReMoDetect	hyunseoki/ReMoDetect-deberta	[11]	HuggingFace
GRM-gemma2-2B	Ray2333/GRM-gemma2-2B-rewardmodel-ft	[26]	HuggingFace
GRM-Llama3.2-3B	Ray2333/GRM-Llama3.2-3B-rewardmodel-ft	[26]	HuggingFace

A.2 Detailed Results of Main Results

Table 8: The performance of various detection methods in the multi-domain task. The best results are marked in **bold** and the second best results are marked by underline.

Metrics →	AUROC	F_1	AUROC	F_1	AUROC	F_1	AUROC	F_1	AUROC	F_1
Domain Settings →	ArXiv		XSum		Writing		Review		Avg.	
<i>Gemma-2-2B-it</i>										
Log-Likelihood	74.48	68.01	59.84	54.29	81.23	74.37	83.44	78.34	74.75	68.75
Rank	50.87	41.75	41.25	28.91	60.26	54.23	61.86	56.92	53.56	45.43
Log-Rank	76.11	70.78	61.23	55.84	81.18	74.43	84.60	78.19	75.78	69.81
LRR	78.98	73.43	65.07	60.53	78.80	70.29	86.86	79.01	77.43	70.81
Fast-DetectGPT	73.25	67.17	70.92	65.08	68.41	61.06	68.49	61.18	70.26	63.62
Lastde	65.63	58.99	58.50	52.64	58.52	52.81	59.84	58.89	60.62	55.83
<i>Gemma-2-2B family (Gemma-2-2B-it+Gemma-2-2B)</i>										
Binoculars	84.07	79.69	69.40	64.60	84.60	75.80	85.34	78.20	80.85	74.57
IRM	<u>97.17</u>	<u>91.24</u>	<u>96.55</u>	<u>90.80</u>	76.47	69.18	68.42	59.60	84.65	77.70
<i>Llama-3.2-1B-Instruct</i>										
Log-Likelihood	79.00	72.61	67.10	61.74	83.15	76.22	88.37	83.38	79.40	73.49
Rank	58.48	51.12	44.80	37.06	59.81	56.76	75.50	69.87	59.65	53.70
Log-Rank	79.36	73.63	67.55	60.60	83.12	76.54	88.56	82.89	79.64	73.41
LRR	77.46	69.80	65.95	59.30	81.25	74.68	87.29	78.85	77.99	70.66
Fast-DetectGPT	85.71	78.26	78.71	72.16	42.42	24.82	65.33	58.50	68.04	58.44
Lastde	80.17	71.95	65.55	59.82	52.15	38.66	62.59	53.68	65.11	56.03
<i>Llama-3.2-1B family (Llama-3.2-1B-Instruct+Llama-3.2-1B)</i>										
Binoculars	93.17	86.51	88.43	81.16	92.03	84.95	96.27	89.91	92.48	85.63
IRM	98.56	95.02	98.45	95.15	<u>97.84</u>	<u>93.19</u>	97.03	91.65	97.97	93.75
<i>Reward Models</i>										
ReMoDetect	92.34	84.63	80.74	75.92	98.12	93.33	<u>96.75</u>	<u>89.92</u>	91.99	<u>85.95</u>
RM-Deberta-v3-large-v2	58.16	45.78	59.37	52.06	95.49	89.59	79.30	72.62	73.08	65.01
GRM-gemma2-2B	33.30	1.18	63.47	57.68	70.16	64.95	37.33	0.98	51.07	31.20
GRM-Llama3.2-3B	45.94	19.45	43.20	3.49	66.08	58.82	26.66	3.31	45.47	21.27

Table 9: The performance of various detection methods in the multi-LLM task. The best results are marked in **bold** and the second best results are marked by underline.

Metrics →	AUROC	F_1	AUROC	F_1	AUROC	F_1	AUROC	F_1	AUROC	F_1
LLM Settings →	GPT-3.5		Claude		PaLM-2		Llama-2		Avg.	
<i>Gemma-2-2B-it</i>										
Log-Likelihood	78.20	69.58	51.97	29.87	80.40	73.01	88.03	78.78	74.65	62.81
Rank	55.17	47.92	48.46	41.12	52.18	44.44	57.99	49.09	53.45	45.64
Log-Rank	78.48	69.46	54.43	38.13	80.76	73.14	88.81	80.73	75.62	65.36
LRR	76.60	68.24	62.97	48.44	79.08	69.57	88.58	80.25	76.81	66.62
Fast-DetectGPT	67.94	62.45	38.21	20.61	81.85	74.71	90.02	81.62	69.51	59.85
Lastde	57.40	58.63	44.49	29.57	65.51	63.65	71.39	67.78	59.70	54.91
<i>Gemma-2-2B family (Gemma-2-2B-it+Gemma-2-2B)</i>										
Binoculars	77.48	69.55	62.21	48.61	89.44	82.36	92.61	84.43	80.43	71.24
IRM	84.03	75.68	<u>86.46</u>	<u>78.44</u>	88.65	79.41	87.11	76.21	86.56	77.43
<i>Llama-3.2-1B-Instruct</i>										
Log-Likelihood	83.26	77.22	59.05	45.91	83.76	76.20	92.63	86.03	79.67	71.34
Rank	63.72	55.82	51.31	36.78	58.10	51.88	64.86	58.90	59.50	50.84
Log-Rank	82.98	74.79	59.73	48.50	83.75	76.31	92.67	86.41	79.78	71.50
LRR	79.73	71.92	60.56	49.61	80.75	72.97	90.13	82.81	77.79	69.33
Fast-DetectGPT	66.27	61.23	44.10	23.85	79.15	72.33	84.96	77.47	68.62	58.72
Lastde	57.91	49.93	54.02	42.85	67.54	60.69	76.18	68.69	63.91	55.54
<i>Llama-3.2-1B family (Llama-3.2-1B-Instruct+Llama-3.2-1B)</i>										
Binoculars	92.26	84.89	81.38	74.40	<u>95.56</u>	90.90	99.10	<u>96.50</u>	<u>92.08</u>	<u>86.67</u>
IRM	97.40	92.22	96.27	89.79	95.95	<u>89.88</u>	99.35	96.58	97.24	92.12
<i>Reward Models</i>										
ReMoDetect	<u>96.14</u>	<u>89.24</u>	82.26	76.74	87.82	79.01	95.41	87.68	90.41	83.17
RM-Deberta-v3-large-v2	78.16	70.24	65.27	51.88	64.61	71.44	73.45	66.08	70.37	64.91
GRM-gemma2-2B	60.93	63.53	53.03	56.21	41.78	1.75	46.52	59.78	50.57	45.32
GRM-Llama3.2-3B	59.63	46.36	36.70	0.98	43.56	25.71	44.69	18.33	46.15	22.85

Table 10: The performance of various detection methods in the multi-attack task. The best results are marked in **bold** and the second best results are marked by underline.

Metrics →	AUROC	F_1	AUROC	F_1	AUROC	F_1	AUROC	F_1	AUROC	F_1	AUROC	F_1
Attack Settings →	Direct		Prompt		Paraph.		Perturb		Mixing		Avg.	
<i>Gemma-2-2B-it</i>												
Log-Likelihood	94.95	88.15	90.97	84.30	70.95	63.77	59.70	54.04	73.01	62.96	77.92	70.65
Rank	79.26	69.84	79.06	70.58	67.04	54.58	14.08	00.00	48.64	35.71	57.62	46.14
Log-Rank	95.25	88.56	91.21	84.38	71.40	64.61	61.72	55.57	74.19	63.71	78.75	71.37
LRR	93.16	85.08	88.92	80.63	70.88	63.38	67.39	59.72	76.07	67.77	79.28	71.32
Fast-DetectGPT	90.59	83.81	84.84	81.34	73.33	68.05	48.55	35.48	69.16	61.41	73.30	66.02
Lastde	91.75	84.12	87.26	81.02	71.32	60.46	18.77	0.00	61.46	55.96	66.11	56.31
<i>Gemma-2-2B family (Gemma-2-2B-it+Gemma-2-2B)</i>												
Binoculars	97.19	91.76	93.27	88.30	79.39	73.27	66.54	57.00	78.59	72.55	83.00	76.58
IRM	89.13	80.01	88.12	77.83	85.50	77.67	85.44	74.45	<u>92.70</u>	<u>85.13</u>	88.38	79.02
<i>Llama-3.2-1B-Instruct</i>												
Log-Likelihood	97.02	91.66	93.84	87.75	76.08	71.06	67.48	60.02	78.81	71.08	82.65	76.31
Rank	86.85	76.62	85.57	76.43	72.80	62.55	18.43	00.00	54.55	46.14	63.60	52.35
Log-Rank	97.06	92.08	93.49	87.83	76.40	70.75	67.16	56.74	78.66	69.46	82.55	75.37
LRR	94.55	87.67	89.94	83.10	74.92	67.65	63.73	52.90	74.67	61.95	79.56	70.65
Fast-DetectGPT	63.94	56.31	60.30	53.76	69.95	63.01	72.12	65.21	81.54	74.05	69.58	62.47
Lastde	87.30	79.51	82.70	74.39	74.29	64.53	31.80	66.69	71.86	61.99	69.59	69.42
<i>Llama-3.2-1B family (Llama-3.2-1B-Instruct+Llama-3.2-1B)</i>												
Binoculars	99.15	96.10	<u>96.52</u>	91.94	90.56	<u>83.01</u>	88.62	<u>81.60</u>	90.37	82.46	93.04	<u>87.02</u>
IRM	<u>98.94</u>	<u>95.52</u>	97.70	93.02	98.00	92.57	95.65	88.65	95.68	90.28	97.19	92.01
<i>Reward Models</i>												
ReMoDetect	97.12	91.00	93.47	84.59	88.59	73.86	<u>88.74</u>	80.44	88.19	80.55	91.22	82.09
RM-Deberta-v3-large-v2	81.88	73.10	76.54	65.28	62.23	45.66	70.96	70.44	65.11	62.74	71.34	63.44
GRM-gemma2-2B	57.93	56.02	56.33	52.33	46.54	66.68	47.42	63.66	45.20	61.15	50.69	59.97
GRM-Llama3.2-3B	54.84	46.79	53.77	41.73	44.56	34.62	39.87	6.92	49.75	23.95	48.56	30.80

Table 11: The performance of various detection methods in the human writing task. The best results are marked in **bold** and the second best results are marked by underline.

Metrics →	AUROC	F_1	AUROC	F_1	AUROC	F_1	AUROC	F_1	AUROC	F_1
Attack Settings →	Direct		Paraph.		Perturb		Mixing		Avg.	
<i>Gemma-2-2B-it</i>										
Log-Likelihood	94.96	88.15	83.10	76.82	99.78	98.56	93.28	86.64	92.78	87.54
Rank	79.27	69.84	70.44	64.51	97.48	93.54	69.03	59.18	79.05	71.77
Log-Rank	95.25	88.56	83.20	78.49	99.81	98.60	93.80	86.53	93.02	88.05
LRR	93.16	85.08	81.79	<u>78.78</u>	99.41	96.66	91.27	82.49	91.41	85.75
Fast-DetectGPT	90.59	83.82	77.95	71.34	98.61	94.92	91.09	84.37	89.56	83.61
Lastde	91.75	84.12	78.53	75.68	99.66	97.53	92.62	85.20	90.64	85.63
<i>Gemma-2-2B family (Gemma-2-2B-it+Gemma-2-2B)</i>										
Binoculars	97.19	91.77	81.44	77.73	99.85	98.51	94.67	88.28	93.29	89.07
IRM	89.13	80.02	64.60	60.92	75.28	74.29	95.77	88.92	81.19	76.04
<i>Llama-3.2-1B-Instruct</i>										
Log-Likelihood	97.03	91.67	83.00	78.12	<u>99.95</u>	<u>99.26</u>	96.11	90.34	94.02	89.85
Rank	86.65	76.62	70.69	70.07	<u>99.55</u>	<u>97.52</u>	80.58	68.60	84.37	78.20
Log-Rank	97.06	92.08	82.78	78.16	99.96	99.45	95.98	89.52	93.94	89.81
LRR	94.55	87.67	80.36	77.37	99.79	97.91	91.80	83.91	91.62	86.72
Fast-DetectGPT	63.95	56.32	52.58	42.72	47.42	40.47	79.36	72.98	60.83	53.12
Lastde	87.30	79.51	73.02	70.37	97.98	93.72	91.79	84.38	87.52	82.00
<i>Llama-3.2-1B family (Llama-3.2-1B-Instruct+Llama-3.2-1B)</i>										
Binoculars	99.16	96.10	81.25	78.81	99.84	98.56	<u>98.26</u>	<u>93.82</u>	94.63	91.82
IRM	<u>98.94</u>	<u>95.52</u>	80.84	76.22	99.35	96.33	98.79	95.06	<u>94.48</u>	<u>90.78</u>
<i>Reward Models</i>										
ReMoDetect	97.12	91.00	76.02	74.91	99.51	97.09	93.89	87.59	91.63	87.65
RM-Deberta-v3-large-v2	81.88	73.10	78.40	75.71	88.37	79.06	77.29	70.60	81.48	74.62
GRM-gemma2-2B	57.94	56.03	68.50	61.71	69.80	65.10	58.53	58.63	63.69	60.37
GRM-Llama3.2-3B	54.84	46.79	63.65	59.71	65.91	59.13	58.44	52.93	60.71	54.64

B Additional Experiments

B.1 Additional Benchmarks

Settings. We further evaluate our method on the RAID [35] and DivScore [36] benchmarks. We adopt LLaMA-3.2-1B as the backbone detector, which demonstrated strong performance on the DetectRL benchmark. For the RAID benchmark, due to the large scale, we randomly sample 512 annotated pairs from the training split for each source model. We report AUROC as the main evaluation metric. For the DivScore benchmark, we use the official test split for evaluation and report both AUROC and F_1 Score. Since the lengths of test samples vary considerably, we compute the mean of implicit rewards across all tokens, instead of the sum (Eq. 5), to normalize the decision threshold, which empirically shows an approximately linear relationship with text length (see Figure 4).

Table 12: Results on RAID benchmark.

Method	chatgpt	gpt4	gpt3	gpt2	llama-chat	mistral	mistral-chat	mpt	mpt-chat	cohere	cohere-chat
Log-Likelihood	98.48	93.51	97.35	68.77	98.93	69.82	97.78	48.84	80.08	89.26	92.42
Log-Rank	98.71	92.95	96.94	71.13	99.12	71.34	98.03	51.83	81.27	87.43	91.89
Binoculars	99.66	96.38	99.12	79.30	99.05	72.45	98.22	51.99	83.41	97.06	98.00
IRM	99.58	99.27	87.43	86.31	99.99	62.25	99.35	57.57	92.42	87.14	92.10
ReMoDetect	99.69	97.82	94.57	57.36	99.33	59.18	97.70	43.06	89.63	85.29	93.02

Table 13: Results on DivScore benchmark.

Method	DeepSeek-R1		DeepSeek-V3		GPT-4o		O3-mini		Avg.	
	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1
Log-Likelihood	88.64	86.11	98.87	97.68	85.96	81.50	90.37	87.47	90.96	88.19
Log-Rank	87.56	84.85	98.91	97.69	86.28	82.34	90.86	88.01	90.90	88.22
Binoculars	98.03	94.18	99.87	99.03	94.84	90.74	95.61	91.88	97.09	93.96
IRM	97.39	93.67	99.27	97.77	99.08	96.74	97.95	94.25	98.42	96.61
ReMoDetect	95.86	90.69	99.73	98.20	99.56	97.46	98.73	94.73	98.47	95.27

Results. As shown in Table 12 and Table 13, IRM performs competitively across a wide range of models, supporting the generalizability of our approach.

B.2 Additional Baselines

Settings. We further include Skywork-Reward-V2 [37] models as additional baselines. These reward models are trained on large-scale preference data and use backbones up to 8B parameters. They achieve SOTA on the RewardBench benchmark, providing a strong reference to evaluate whether reward model scaling enhances generalization in LLM-generated text detection.

Table 14: Comparison of IRM and Skywork-Reward-V2 on the DetectRL benchmark.

Tasks Settings →	Multi-Domain		Multi-LLM		Multi-Attack		Generalization			Length		Human Writing		Avg.
Methods ↓	AUROC	F_1	AUROC	F_1	AUROC	F_1	Domain F_1	LLM F_1	Attack F_1	Train F_1	Test F_1	AUROC	F_1	
<i>Llama-3.1-8B</i>														
Skywork-Reward-V2	71.33	62.97	70.67	66.15	68.23	65.85	59.06	65.59	63.23	45.41	61.78	39.32	23.98	58.74
IRM	90.52	84.67	89.01	83.18	88.70	82.13	81.10	79.36	81.47	74.28	76.78	82.61	76.18	82.31
<i>Llama-3.2-3B</i>														
Skywork-Reward-V2	63.48	48.47	63.33	57.78	60.32	41.35	44.50	56.99	32.58	34.77	49.65	30.52	6.91	45.43
IRM	97.31	92.51	97.25	92.15	97.21	91.69	91.68	90.67	91.37	78.59	80.97	93.85	88.85	91.08
<i>Qwen3-8B</i>														
Skywork-Reward-V2	68.21	57.01	67.94	62.23	66.08	55.31	54.21	61.97	54.88	40.19	44.90	35.90	19.82	52.97
IRM	74.42	58.02	73.08	59.05	73.70	62.13	55.77	57.70	61.15	42.54	41.77	69.61	58.49	60.57

Results. As shown in Table 14, IRM consistently outperforms the reward-model-based methods, even when using the same or smaller backbones. This further supports our claim that standard preference-trained reward models, despite their size and strong performance on preference classification, do not generalize well to the LLM-generated text detection task. In contrast, IRM’s implicit reward modeling is inherently aligned with this detection objective, leading to robust performance across different settings.

Algorithm 1 Inference pipeline of IRM

Inputs: policy model π_θ , reference model π_{ref} , detected text x , decision threshold t .

Output: authenticity label $y \in \{\text{human-written, LLM-generated}\}$

```
1:  $p_{\text{ref}} \leftarrow \pi_{\text{ref}}(x)$ 
2:  $p_\theta \leftarrow \pi_\theta(x)$ 
3:  $r \leftarrow \log p_\theta - \log p_{\text{ref}}$  ▷ Compute implicit reward score
4: if  $r < t$  then
5:    $y \leftarrow \text{human-written}$ 
6: else
7:    $y \leftarrow \text{LLM-generated}$ 
8: end if
9: return  $y$ 
```
