# Learning to Rewrite: Generalized LLM-Generated Text Detection

**Wei Hao**[1*], **Ran Li**[1*], **Weiliang Zhao**[1], **Junfeng Yang**[1], **Chengzhi Mao**[2]
Columbia University[1], Rutgers University[2]
{rl3424, wh2473, wz2665, jy2324}@columbia.edu, cm1838@rutgers.edu

## Abstract

Detecting text generated by Large Language Models (LLMs) is crucial, yet current detectors often struggle to generalize in open-world settings. We introduce **Learning2Rewrite**, a novel framework to detect LLM-generated text with exceptional generalization to unseen domains. Capitalized on the finding that LLMs inherently modify LLM-generated content less than human-written text when rewriting, we train an LLM to amplify this disparity, yielding a more distinguishable and generalizable edit distance across diverse text distributions. Extensive experiments on data from 21 independent domains and four major LLMs (GPT-3.5, GPT-4, Gemini, and Llama-3) demonstrate that our detector outperforms state-of-the-art detection methods by up to 23.04% in AUROC for in-distribution tests, 35.10% for out-of-distribution tests, and 48.66% under adversarial attacks. Our unique training objective ensures better generalizability compared to directly training for classification, even when leveraging the same amount of tunable parameters. Our findings suggest that reinforcing LLMs' inherent rewriting tendencies offers a robust and scalable solution for detecting LLM-generated text.

## 1 Introduction

Large Language Models (LLMs) demonstrate exceptional capabilities in various tasks (Radford et al., 2019; Brown et al., 2020; Achiam et al., 2023; Touvron et al., 2023; Team et al., 2023; OpenAI, 2020). However, they can be misused for illegal or unethical activities, such as spreading misinformation (Chen and Shu, 2023), scaling phishing campaigns (Hao et al., 2025), manipulating social media (Zhang et al., 2024), and generating propaganda (Pan et al., 2023). LLMs also facilitate academic dishonesty (Zellers et al., 2019; Mvondo et al., 2023), and training foundation models with generated content can lead to irreversible defects in the resulting models (Shumailov et al., 2023).

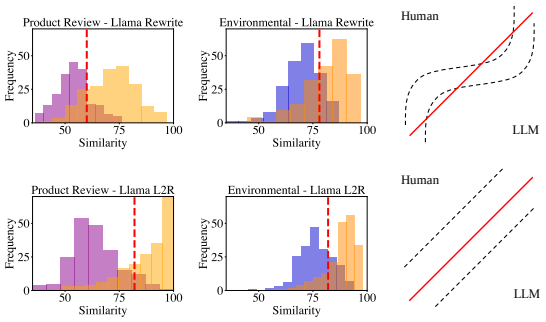


Figure 1: Rewriting for LLM-generated Text Detection. The histograms depict the edit distance distributions for texts generated by human and LLMs, illustrating how fine-tuning a rewrite model enhances their separation. We show two domains: Purple and Yellow represent human and LLM distributions for Product Review texts, while Blue and Orange represent those for Environmental texts. Without fine-tuning the rewrite model, human and LLM distributions are inseparable by a single threshold (red line, above). After fine-tuning, the texts can be separated by one threshold (below). On the right, we conceptualize L2R's intuition by showing that the rugged decision boundary between human and LLM texts, caused by varying data distributions across domains, can be better aligned and divided by a single threshold after fine-tuning. Specifically, the standard deviation in decision thresholds among all domains decreases from 0.7506 to 0.4428 after fine-tuning.

These issues highlight the urgent need for reliable algorithms to detect LLM-generated text.

Various methods for detecting LLM-generated text have been proposed (Solaiman et al., 2019; Mitrović et al., 2023; Mitchell et al., 2023; Su et al., 2023; Liu et al., 2024; Bao et al., 2024; Mao et al., 2024; Verma et al., 2024; Gehrmann et al., 2019). Most of these detectors employ pre-trained models, extracting hand-crafted features and heuristics (e.g., loss curvature (Bao et al., 2024), edit distance from rewriting (Mao et al., 2024)), and apply thresholds to separate LLM from human text. However, these thresholds are highly domain-dependent, hindering a universal detection standard.

[*]Equal contribution, order is sorted by last name.



*Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6421–6434
July 27 - August 1, 2025 

In this paper, we present L2R (Learning-to-Rewrite), which trains an LLM to specifically perform more edits when asked to rewrite human-generated text and fewer edits when rewriting LLM-generated text across a diverse set of domains, thus effectively distinguish LLM-generated text from human-generated one. Unlike traditional detectors designed solely for binary classification, which perform well in-distribution (ID) but struggle to generalize to out-of-distribution (OOD) domains including adversarial attacks, our method reinforces LLMs' inherent reluctance to rewrite their own outputs by using rewriting as an additional training objective to maximize this tendency, thereby enhancing generalizability and enabling a single detection threshold across diverse distributions. Figure 1 illustrates an example of how L2R learns to make LLM and human-generated text more separable across domains, compared to only rewriting using a pre-trained model (Mao et al., 2024).

On a dataset spanning 21 domains (e.g., finance, entertainment, cuisine) and constructed using four major LLMs (GPT-3.5, GPT-4, Gemini, and Llama-3), L2R surpasses the state-of-the-art detectors, achieving up to 19.56% higher AUROC ID and 35.10% higher OOD than Verma et al. (2024), 23.04% higher ID and 9.90% higher OOD than Bao et al. (2024), and 10.39% higher ID and 4.67% higher OOD than Mao et al. (2024). Compared with fine-tuning a Llama-3 model for naive text classification, L2R has 51.35% higher AUROC OOD despite leveraging the same number of tunable parameters. L2R also outperforms the state-of-the-art detectors by up to 48.66% under adversarial attacks. These results demonstrate that our training objective offers superior accuracy and generalizability. Furthermore, our method provides interpretability by highlighting the rewritten portions of the text. Our codebase is open-sourced[1] and our contributions are summarized as follows:

- While binary classifiers often learn spurious, domain-specific features for LLM-generated text detection, we propose L2R, which learns a proxy based on the minimal edit distance on LLM content, yielding a more robust and invariant detection threshold.
- We build a diversely generated dataset (21 domains) and design a calibration loss function to make fine-tuning both effective and stable.
- We conduct comprehensive evaluations on ID, OOD datasets and against different adversarial attacks (Decoherence and Rewrite bypassing), showing that L2R surpasses state-of-the-art detection methods.

[1] https://github.com/ranhli/l2r_data

## 2 Related Work

Various LLM-generated text detectors have been proposed over the years. One set of detectors trains a model on the input text (Solaiman et al., 2019; Mitrović et al., 2023; Liu et al., 2023). These methods excel in their training domains, but struggle under OOD evaluation (Pu et al., 2023), i.e., detection with text from different domains or unfamiliar models. The second set of detectors utilize the raw output, i.e., logits, from pre-trained LLMs to assign probability score for detection. GLTR (Gehrmann et al., 2019) utilizes statistical features like log probability, word rank, and entropy to assign score, Ghostbuster (Verma et al., 2024) utilizes log probability and unigram and bigram probability, DetectGPT (Mitchell et al., 2023) employs the delta in log probability of the input text after token perturbation to estimate AI likelihood, PECOLA (Liu et al., 2024) selectively applies perturbation for enhanced accuracy, and Fast-DetectGPT (Bao et al., 2024) simplifies the process by exploiting conditional probability curvature. This family of detectors shows improved generalizability, but all require access to the raw output of an LLM. Since the main targets, namely the commercial LLMs, are not open sourced, this poses a challenge for accurate probability estimation using proxy models. Lastly, RAIDAR (Mao et al., 2024) is a detection method based on the observation that LLMs, when asked to rewrite a given text, tend to produce a higher number of edits for human-written text compared to LLM-generated text. Despite the attempt to capture the edit distance from rewrite as a domain-agnostic feature, the amount of edits still varies across distributions, and the threshold of edit amount between human and LLM texts learned on training domains does not generalize to OOD domains, which limits its full potential.

## 3 Method

### 3.1 Rewriting for LLM Detection

Rewriting input with LLM and then measuring the edits proves to be a successful way to detect LLM-generated text. Given an held-out input text set $\mathbf{X}_{train}$ with LLM and human generated texts, and its corresponding label set $\mathbf{Y}_{train}$, an LLM $F(\cdot)$ is

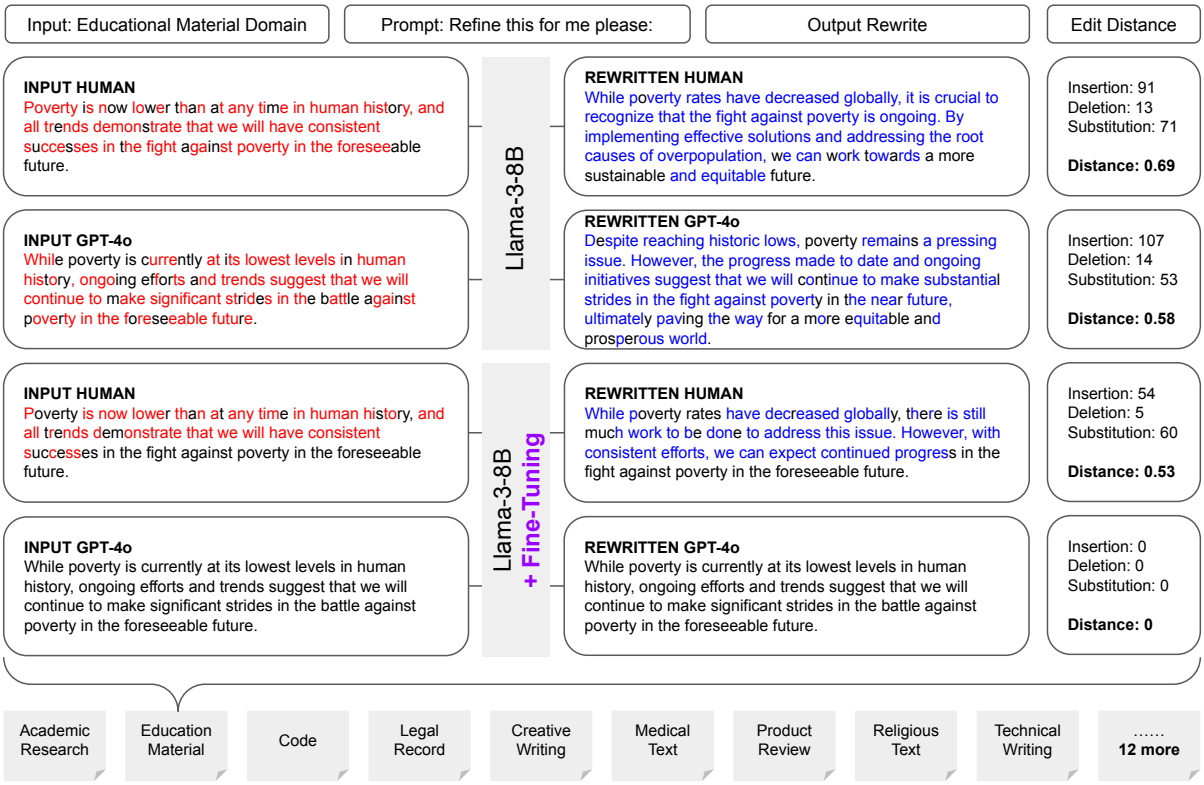


Figure 2: Rewriting examples with edits. Deleted characters are marked in red, added characters are marked in blue, and unmodified characters are in **black**. We exploit the difference in edit distance between human and LLM-generated text for classification. While the pre-trained Llama-3 model give different amount of edits for human and LLM-generated text (above), rewrites from our fine-tuned model are much more separable (below).

prompted to rewrite the input $\mathbf{x} \in \mathbf{X}_{train}$ using a prompt $\mathbf{p}$. The rewriting output is $F(\mathbf{p}, \mathbf{x})$. Particularly, the prompt $\mathbf{p}$ can be set to: "`Refine this for me please`".

The edit distance between the input text and the rewritten output, $D(\mathbf{x}, F(\mathbf{p}, \mathbf{x}))$, is then computed for all $\mathbf{x} \in \mathbf{X}_{train}$. Mao et al. (2024) adopts the Levenshtein score (Levenshtein et al., 1966), which measures the minimum number of insertions, deletions, or substitutions required to transform one text into the other. A higher score denotes the two strings are more similar. With the Levenshtein score, an edit distance used for classification is calculated as:

$$D_k(\mathbf{x}, F(\mathbf{p}, \mathbf{x})) = 1 - \frac{\text{Levenshtein}(F(\mathbf{p}, \mathbf{x}), \mathbf{x})}{\max(len(F(\mathbf{p}, \mathbf{x})), len(\mathbf{x}))}. \quad (1)$$

Mao et al. (2024) trains a classifier, such as logistic regression or decision tree, to threshold the edit distance and predict if a text is written by an LLM. However, as shown in Figure 1, the threshold of rewriting with a pre-trained LLM often varies from one domain to another, causing RAIDAR to fail to generalize to new domains.

### 3.2 Fine-Tuning the Rewrite Model

L2R works on the premise that human-written and LLM-generated text would cause a different amount of edits and a boundary can be drawn to separate both distributions. Thus we can finetune such a rewrite model $F'(\cdot)$, that gives as much edits as possible for human texts, while leaving the LLM texts unmodified, demonstrated in Figure 2. Given some human text $\mathbf{x}_h \in \mathbf{X}_{train}$ and LLM text $\mathbf{x}_{llm} \in \mathbf{X}_{train}$, our objective becomes:

$$\max\{D(\mathbf{x}_h, F'(\mathbf{p}, \mathbf{x}_h)) - D(\mathbf{x}_{llm}, F'(\mathbf{p}, \mathbf{x}_{llm}))\} \quad (2)$$

Since the edit distance is not differentiable, we use the cross-entropy loss $L(\cdot)$ assigned to the input $\mathbf{x}$ by $F'(\cdot)$ as a proxy to the edit distance. As a result, for each of input $\mathbf{x}$ with label $y = 1$ (LLM) or 0 (human), our objective becomes:

$$\min\{L(\mathbf{X}_{\text{train}}) \cdot y_{\text{train}}\}, \quad y_{\text{train}} = \begin{cases} 1 & \text{(LLM)} \\ -1 & \text{(human)} \end{cases} \quad (3)$$

In this way, we flip the sign of the loss when the inputs are written by human. Since the overall loss would be minimized, this effectively encourages

the rewrites to be different from human input and identical to the LLM input.

### 3.3 Calibration Loss during Fine-Tuning

When fine-tuning the rewrite model on Equation 3, the rewrite model aims to minimize the edits on LLM-generated text and maximize the edits on human-generated text. However, without posting regularization and constraint on the unbounded loss, the rewrite model takes the risk of being corrupted (e.g., verbose output for all rewrite and over-fitting with more edits on human-generated text rewrite) which we evaluated in §A.4.

Therefore, we propose a calibration loss, which prevents the over-fitting problem by imposing a threshold value $t$ on the loss for each given input. For human text $\mathbf{x}_h$, we apply gradient backpropagation only if $L(\mathbf{x}_h) < t$. For LLM text $\mathbf{x}_{llm}$, we apply backpropagation only if $L(\mathbf{x}_{llm}) > t$. Otherwise, the gradient is set to 0. We show a pseudocode for the algorithm below:

**Algorithm 1** Calibration Loss Calculation

**Require:** Threshold $t$, loss $L(\cdot)$, human text $x_h$, LLM text $x_{llm}$

$L_h \leftarrow L(x_h)$, $L_{llm} \leftarrow L(x_{llm})$, $L \leftarrow 0$

$L \leftarrow L - L_h$ if $L_h < t$

$L \leftarrow L + L_{llm}$ if $L_{llm} > t$

**return** $L$

Therefore, rather than minimizing the loss proxy, our objective becomes separating the distributions of the edit distance, for rewrites on human and LLM inputs, to two ends of the threshold $t$. Concretely, this enables the model to only optimize against the hard examples, and leave those that are already classified correctly unchanged, to prevent overfitting. This is similar to DPO (Rafailov et al., 2023), where we fine-tune the rewrite model using only preference data, namely the rewrites that are not yet separated by the existing boundary. This process is depicted by the graphical illustrations in Figure 1.

To determine the threshold $t$, we perform a forward pass using the rewrite model before fine-tuning on $\mathbf{X}_{train}$ and train a logistic regression model on all loss values. The threshold $t$ can be derived from the weight and the intercept of the logistic regression model. In practice, applying the calibration loss improves detection performance by 4.54% in AUROC among the 21 domains, from 0.8555 to 0.9009.

## 4 Dataset

Existing detectors are often evaluated on public datasets such as SQuAD (Rajpurkar et al., 2016), XSum (Narayan et al., 2018), and Writing Prompts (Fan et al., 2018) or self-constructed ones. However, these datasets typically represent a narrow subset of available data, both in terms of timeliness and domain coverage. This limitation raises concerns about over-fitting and uncertainty regarding how these detectors would perform when deployed in real-world scenarios, highlighting the necessity in creating a dataset of diversely-distributed texts for training.

### 4.1 Data Collection

To ensure the robustness and generalizability of our detection model, we construct a dataset consisting of human-written text from 21 distinct domains, including finance, entertainment, cuisine, etc. For each domain, we collect the texts either by crawling online platforms like Wikipedia or by sampling from publicly available datasets. From these collections, we randomly select 200 complete paragraphs as text snippets which yields an average length of 120 words among the samples. For each of the human-written samples, we create four LLM-generated counterparts using four state-of-the-art LLMs: GPT-4o (OpenAI, 2024), GPT-3.5-Turbo (OpenAI, 2020), Gemini 1.5 Pro (Reid et al., 2024), and Llama-3-70B-Instruct (Meta, 2024). The entire process results in a total of 21,000 text samples. Details on data generation are listed in Table 5, and descriptions of the domains and their sources are provided in §A.1.

### 4.2 Prompt Diversity

Conventionally, LLM-generated text is created by prompting LLMs to either rewrite a given text or continue writing from a given prefix, often using a single, static prompt for the entire process (Mitchell et al., 2023; Bao et al., 2024; Verma et al., 2024; Mao et al., 2024). However, real-world text generation involves a wide variety of prompts, which can significantly alter the distribution of the generated text. Previous work (Mao et al., 2024) has shown that one straightforward way to bypass the RAIDAR detector is by using the prompt "`Help me rephrase it, so that another GPT rewriting will cause a lot of modifications`", which suggests that data generated by different prompts are different in distribu-

tion, indicating the importance of prompt diversity. To address this, we curate a dataset of 200 rewrite prompts, each containing slight variations in phrasing and instructions. For each generated text, a prompt is randomly sampled from this dataset. Examples of the prompts we use are provided below:

- Refine this for me please:
- Please rewrite this content in your own words:
- Make this text more formal and professional:
- Make this text more casual and friendly:
- Rephrase this text in a more elaborate way:
- Reframe this content in a more creative way:
- Rewrite this text to emphasize the key points:
- Help me rephrase it, so that another GPT rewriting will cause a lot of modifications:

For a RAIDAR detector, training on a diversely-prompted dataset compared with a single-prompted dataset can increase its testing AUROC from 0.7302 to 0.7566 (detailed in A.2). This shows that diverse prompts enables the model to better capture the distribution of LLM texts in the real world, whose generation prompts are expected to vary significantly.

### 4.3 Data Cleaning

In collecting human-written text, we ensure that no data is generated after November 30, 2022, the release date of ChatGPT (OpenAI, 2020), avoiding contamination of human dataset with LLM-generated content. Instead of manually introducing any truncations, we split all texts into natural paragraphs, yielding an overall average length of 120 words with a standard deviation of 108 words. For LLM-generated text, we carefully remove any extraneous suffixes, such as "`Sure, here is a...`", to avoid them from being detected in this way.

## 5 Evaluation

This section answers the following questions:

**Q1:** How does L2R compare with other detectors? (§5.3)

**Q2:** How does L2R perform when OOD? (§5.4)

**Q3:** How does L2R perform under adversarial attacks? (§5.5)

**Q4:** How does L2R's training objective compare with directly training for binary classification? (§5.6)

**Q5:** How does training on our proposed dataset contribute to L2R's performance? (§5.7)

### 5.1 Experiment Setup

We perform all experiments on one NVIDIA A100 GPU with 40GB RAM. We use "meta-Llama/Meta-Llama-3-8B-Instruct" (AI@Meta, 2024) as the open-sourced rewrite model in all experiments. To fine-tune the Llama model with 8B parameters, we employ 4-bit QLoRA (Dettmers et al., 2024), with parameter `r` set to 16, `lora_alpha` set to 32, and `lora_dropout` set to 0.05, unless otherwise noted. We use an initial learning rate of 5e-6, a weight decay of 0.01, and a batch size of 32 to train until convergence. We set the sampling temperature to 0 when using Llama for rewriting during training and detection for deterministic and reproducible results, therefore taking the results from a single run for the experiments. We use 70% of the dataset for training and the rest for testing in all experiments. Training on the 21 domains takes around six GPU hours and rewriting a single text of 120 words takes an average of 13.5 seconds.

### 5.2 Baselines

Our baseline detectors consist of Fast-DetectGPT (Bao et al., 2024), Ghostbusters (Verma et al., 2024), RAIDAR (Mao et al., 2024), and a custom approach named "Llama Logits", which involves training a Llama-3-8B model together with a classifier (same size as RAIDAR and L2R) on its logits output to perform naive text classification. For Ghostbuster, RAIDAR and Llama Logits, we train and test these detectors on the identical training and testing sets as L2R. For Fast-DetectGPT, we use its local version available at Fast-DetectGPT (2024). For Llama Logits, we train its Llama model using the same QLoRA configurations as the rewrite model in L2R for a fair comparison. We also experiment on using a close-sourced model, Gemini 1.5 Pro (Reid et al., 2024) (referred to as Gemini Rewrite), as the rewrite model for RAIDAR in addition to Llama.

### 5.3 Compare L2R with Other Detectors

We compare the performance of L2R with Fast-DetectGPT, Ghostbusters, and RAIDAR (Llama Rewrite and Gemini Rewrite), by measuring the Area Under the Receiver Operating Characteristic Curve (AUROC) scores. The resulting scores for each domain along with their average and standard deviation can be found in Table 1. L2R constantly outperforms both configurations of RAIDAR in all domains; outperforms Fast-DetectGPT in 20 of 21 domains by an average of 23.04% in AU-

| Domain | Fast-DetectGPT | Ghostbusters | RAIDAR (Gemini Rewrite) | RAIDAR (Llama Rewrite) | Llama L2R |
|---|---|---|---|---|---|
| AcademicResearch | 0.4664 | 0.6597 | 0.7911 | 0.8311 | **0.8406** |
| ArtCulture | 0.6292 | 0.6781 | 0.7711 | 0.6750 | **0.8328** |
| Business | 0.6829 | 0.8331 | 0.8153 | 0.8369 | **0.9156** |
| Code | 0.6808 | 0.3770 | 0.5670 | 0.3840 | **0.8383** |
| EducationalMaterial | 0.7474 | 0.8506 | 0.9339 | **0.9675** | 0.9644 |
| Entertainment | 0.8392 | 0.8600 | 0.7836 | 0.8319 | **0.9494** |
| Environmental | 0.8382 | 0.8447 | 0.9081 | 0.9228 | **0.9786** |
| Finance | 0.6879 | 0.7828 | 0.6917 | 0.8153 | **0.9400** |
| FoodCuisine | 0.7425 | 0.6703 | 0.7181 | 0.7831 | **0.9547** |
| GovernmentPublic | 0.7100 | 0.6833 | 0.7375 | 0.7619 | **0.8675** |
| LegalDocument | **0.8365** | 0.5453 | 0.5528 | 0.6594 | 0.7803 |
| LiteratureCreativeWriting | 0.7928 | **0.9456** | 0.8056 | 0.9161 | 0.9294 |
| MedicalText | 0.5693 | 0.6242 | 0.7614 | 0.7700 | **0.7857** |
| NewsArticle | 0.5808 | 0.6800 | 0.7714 | 0.8547 | **0.9242** |
| OnlineContent | 0.6292 | 0.5922 | 0.7408 | 0.8231 | **0.8881** |
| PersonalCommunication | 0.5392 | 0.7042 | 0.6783 | 0.7233 | **0.8239** |
| ProductReview | 0.6467 | 0.7364 | 0.7150 | 0.8075 | **0.9689** |
| Religious | 0.6314 | 0.6111 | 0.7772 | 0.8397 | **0.9775** |
| Sports | 0.6015 | 0.6561 | 0.6917 | 0.7869 | **0.8742** |
| TechnicalWriting | 0.6075 | 0.7242 | 0.8269 | 0.8575 | **0.9369** |
| TravelTourism | 0.6210 | 0.7517 | 0.8492 | 0.8897 | **0.9475** |
| AVERAGE | 0.6705 | 0.7053 | 0.7566 | 0.7970 | **0.9009** |
| STD | 0.1015 | 0.1259 | 0.0928 | 0.1212 | **0.0634** |

Table 1: Comparison of detection performance measured with AUROC scores. For Ghostbuster and all rewrite-based detectors, we train a single classifier on the training set of all domains, then test the model's performance on the test set of each individual domain. **AVERAGE** measures a detector's average performance among all independent domains, and **STD** measures the standard deviation across domains.

ROC; and outperforms Ghostbusters in 20 of 21 domains by an average of 19.56% in AUROC. L2R achieves a 5.62% lower AUROC score than Fast-DetectGPT on legal document domain, and a 1.62% lower AUROC score than Ghostbusters on literature creative writing domain. This may stem from the distinct distributions of these domains: legal documents demand a rigorous writing style, leaving little room for rewriting even with human input, whereas creative writing is more casual, allowing greater rewrite flexibility even for LLM input, thereby making it harder for L2R to distinguish.

In general, the fluctuating AUROC scores indicate the challenging nature of our dataset and the diversity and independence of the distributions across domains. These results also show that L2R has better knowledge of the intricate differences between human and LLM-generated text in various domains compared with the baselines, and is thus more capable in the real-world setting.

### 5.4 OOD Dataset Evaluation

We showed that L2R outperforms the state-of-the-art detectors ID in terms of AUROC scores, but it is equally important to assess its robustness under OOD conditions, as training-based detectors are prone to overfitting to familiar domains and generator models. We first evaluate this by showing its performance on OOD datasets.

To assess L2R's performance on OOD data, we adopt the M4 dataset (Wang et al., 2024), an OOD dataset that is different from our training data in multiple dimensions, including data generation models, text length, decoding strategy, and domains. We show a detailed comparison in Table 2.

The results of the OOD evaluation are presented in Table 3. We include both ID and OOD results to highlight the degree of overfitting for each detector. While the Llama Logits method achieves the highest ID AUROC, its OOD result is the lowest, indicating significant overfitting to the training data. Similarly, Ghostbuster shows overfitting with its OOD AUROC being roughly half of its ID

| Dataset | Ours | M4 |
|---|---|---|
| Generator | GPT-3.5-Turbo, GPT-4o, Llama-3-70B, Gemini 1.5 Pro | BLOOMz, ChatGPT, Davinci, Cohere, Dolly V2 |
| Text Length | Mean: 765 chars, STD: 654 chars | Mean: 1365 chars, STD: 244 chars |
| Decoding Strategy | Nucleus Sampling, Temperature = 1, top_p = 1 | Varies |
| Domains | 21 English domains | 5 English domains |

Table 2: Comparison of characteristics of our dataset and M4 dataset, which we use for OOD evaluation.

| Model | In-Distribution | Out-of-Distribution |
|---|---|---|
| Ghostbusters | 0.7053 | 0.3888 |
| Fast-DetectGPT | 0.6705 | 0.6408 |
| Llama Logits | **0.9774** | 0.1426 |
| Llama Logits (Reduced Params) | 0.8016 | 0.3450 |
| Llama Rewrite | 0.7970 | 0.6931 |
| Llama L2R | 0.9009 | 0.6561 |
| Llama L2R (Reduced Params) | 0.8315 | **0.7398** |

Table 3: ID and OOD performance measured in AUROC scores. For L2R and Llama Logits, the "Reduced Params" models are tuned with approximately 1/4 of the parameters for better generalizability. With reduced parameters, L2R has the highest OOD AUROC, outperforming the naive Llama Rewrite both ID and OOD by 3.45% and 4.67% respectively, suggesting its generalizability through fine-tuning.

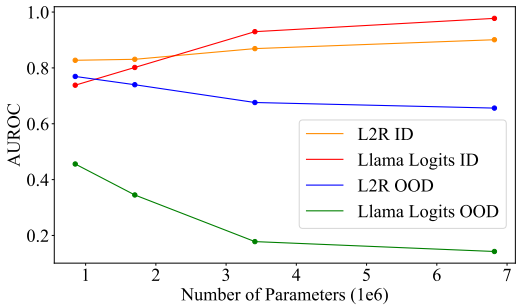


Figure 3: Relationship between the number of trainable parameters and ID and OOD AUROC scores for L2R and RAIDAR. As the number of parameters increase from $1 \times 10^6$ to $7 \times 10^6$, both L2R and Llama Logits show higher ID performance and lower OOD performance, showing how the effect of overfitting emerges as we increase the LLM's trainable parameters. However, L2R continuously outperforms Llama Logits either in OOD setting or in both ID and OOD settings, showing the superior robustness and accuracy of L2R.

performance. The naive rewrite-based approach shows superior robustness compared with these other methods, but L2R trained with reduced parameters, i.e. rank `r` set to 4 and `lora_alpha` set to 8, outperforms Llama Rewrite by 3.45% ID and 4.67% OOD. This demonstrates that our fine-tuning does not simply overfits the rewrite model to the training data, but enhances its classification performance across diverse distributions.

We notice that reducing the number of training parameters make the model more generalizable, and further investigate the impact of fine-tuning parameters on L2R's performance ID and OOD. By adjusting the LoRA parameters `r` and `lora_alpha`, we define four fine-tuning configurations with the number of trainable parameters ranging from 851,968 to 6,815,744, with details listed in §A.3. Figure 3 illustrates the results, where we observe a consistent increase in ID AUROC, accompanied by a decline in OOD AUROC as the number of parameters grows. This suggests that the model becomes increasingly overfitted to the training distribution. L2R either outperforms Llama Logits OOD or both ID and OOD, and all four configurations outperform Ghostbusters and Fast-DetectGPT both ID and OOD. Also, the first two configurations surpass Llama Rewrite in terms of AUROC across both settings.

### 5.5 Adversarial Attack

We employ two distinct types of attack to assess L2R's robustness against the baseline detectors. For both experiments, we apply the attack to all LLM-generated text in the testing set across all domains, while training L2R and the baselines on the unmodified training set and evaluating it on the modified testing set.

#### 5.5.1 Decoherence Attack

Bao et al. (2024) introduces the decoherence attack where two adjacent, randomly selected words are transposed in all sentences longer than 20 words within a paragraph for LLM texts. Bao et al. (2024) demonstrated that this simple attack can be highly effective in degrading the performance of sate-of-the-art detectors, without affecting the core meaning of the input. We present the results of this attack in Table 4, where L2R achieves the highest AUROC on samples subjected to this attack, indicating its superior robustness compared to other models. This is because our rewrite-based objective function for fine-tuning teaches the model the innate distributions of human and LLM text, in-

| Model | No Attack | Decoherence Attack | Rewrite Attack |
|---|---|---|---|
| Ghostbusters | 0.7053 | 0.4730 | 0.4061 |
| Fast-DetectGPT | 0.6705 | 0.4984 | 0.5100 |
| Llama Logits | **0.9774** | 0.7281 | 0.6563 |
| Llama Rewrite | 0.7970 | 0.7681 | 0.7944 |
| Llama L2R | 0.9009 | **0.8746** | **0.8927** |

Table 4: Adversarial attack results. While all detectors show performance degradation, L2R has the highest AUROC in both settings, suggesting its robustness.

stead of relying on brittle statistical features that are easily altered through this simple attack.

#### 5.5.2 Rewrite Attack

Previous work finds that paraphrase or rewrite attacks can degrade the performance of LLM-generated content detectors (Lu et al., 2024; Krishna et al., 2023; Mao et al., 2024). Mao et al. (2024) introduces the rewrite attack in which a GPT-3.5-Turbo model is prompted to refine an input paragraph generated by LLMs, in such a way that a subsequent rewrite by another LLM would result in significant changes (e.g., using the prompt "`Help me rephrase it, so that another GPT rewriting will cause a lot of modifications`"). Mao et al. (2024) showed that this type of attack is effective against RAIDAR and we further show it can affect other types of detectors as well in Table 4, even with our diverse training set. However, L2R still achieves the highest AUROC among all detectors, demonstrating its robustness against this attack. This is because its fine-tuning objective induces a sufficiently large separable gap between rewrites of human and LLM-generated text, allowing adversarially perturbed samples to remain within the LLM distribution. Before attack, the average edit distance is 0.3019 for human rewrites, and 0.1394 for LLM rewrites. After attack, the average edit distance for LLM increases to 0.1614, indicating that the rewrite attack partially shifts the LLM distribution toward the human distribution. However, a clear gap remains between the two, resulting in only a marginal degradation in L2R 's classification performance.

### 5.6 Compare L2R with Direct Fine-Tuning

A valid concern regarding L2R's superior performance is whether it is due to our fine-tuning objective, which enhances model's rewriting ability, or it stems solely from the fact that we exploit the vast parameters of an LLM. To answer this question, we compare L2R with the "Llama Logit" baseline in Table 3 and 4. The Llama Logits detector involves fine-tuning a Llama-3-8B model not for rewrite, but directly for binary classification.

In §5.4, we show that despite the Llama Logits has the highest ID AUROC score among all detectors, surpassing L2R by 7.65%, it has the lowest AUROC when evaluated OOD, up to 51.35% lower than L2R, suggesting that its performance ID is due to overfitting. This highlights the importance of our fine-tuning objective function in ensuring domain-agnostic detection accuracy. Also, the Llama Logits is inferior under adversarial attacks, with 14.65% and 23.64% lower AUROC for decoherence and rewrite attacks, respectively. This again shows L2R's robustness in capturing the true underlying distributions of human and LLM texts.

### 5.7 Effectiveness of the Diverse Dataset

While there exists public datasets that emphasize data diversity, including RAID (Dugan et al., 2024), RuTAD (Maloyan et al., 2022), and MAGE (Li et al., 2024), the contribution of our proposed dataset lies in its ability to help train a robust and generalizable L2R model. We show this by training L2R on MAGE using the same number of texts and under the same training configurations, then test its performance ID and OOD on the M4 dataset. We compare the results in Table 5, where the L2R model trained on our dataset has 15.98% higher OOD AUROC, suggesting that the diverse text distributions in our dataset is effective in training a robust and generalizable L2R model.

| Training Dataset | In-Distribution | Out-of-Distribution |
|---|---|---|
| MAGE | 0.8333 | 0.4963 |
| Ours | **0.9009** | **0.6561** |

Table 5: Comparison of L2R's ID and OOD performance when trained on MAGE and our dataset. The superior OOD AUROC from the model trained on our dataset suggests its training effectiveness.

## 6 Conclusion

We present L2R, a method designed to enhance the detection of LLM-generated text by learning to rewrite more on LLM-generated inputs and less on human-generated inputs. L2R excels in identifying LLM-generated text collected across various models and 21 unique domains, both ID and OOD, and under adversarial attacks. Our work demonstrates that LLMs can be trained to detect text generated by other LLMs, surpassing previous detection methods in accuracy and generalizability.

## 7 Limitations

A limitation of ours is the relatively slow inference runtime. As most detectors only requires a forward pass from the LLM being used, we need to call generate to create a rewrite. Nevertheless, this problem would be well alleviated considering the rapid improvement in LLM efficiency and computing power.

## 8 Acknowledgments

We thank Asaf Cidon for supporting Wei Hao's participation in this project. We thank anonymous reviewers for their insightful feedback that helped improve our work.

## References

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

AI@Meta. 2024. Llama 3 model card.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Canyu Chen and Kai Shu. 2023. Combating misinformation in the age of llms: Opportunities and challenges. *AI Magazine*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

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.

Fast-DetectGPT. 2024. GitHub Repository. [link].

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

Wei Hao, Van Tran, Vincent Rideout, Zixi Wang, An-Mei Dasbach-Prisk, M. H. Afifi, Junfeng Yang, Ethan Katz-Bassett, Grant Ho, and Asaf Cidon. 2025. Do spammers dream of electric sheep? characterizing the prevalence of llm-generated malicious emails. In *Proceedings of the 25th ACM Internet Measurement Conference*, IMC '25, New York, NY, USA. Association for Computing Machinery.

IMDb. 2024. Imdb non-commercial datasets. https://developer.imdb.com/non-commercial-datasets/.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*, volume 10, pages 707–710. Soviet Union.

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.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*.

Shengchao Liu, Xiaoming Liu, Yichen Wang, Zehua Cheng, Chengzhengxu Li, Zhaohan Zhang, Yu Lan, and Chao Shen. 2024. Does detectgpt fully utilize perturbation? bridging selective perturbation to fine-tuned contrastive learning detector would be better. In *Proceedings of the 62nd Annual Meeting of the*

*Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1874–1889.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2023. Coco: Coherence-enhanced machine-generated text detection under low resource with contrastive learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16167–16188.

Ning Lu, Shengcai Liu, Rui He, Yew-Soon Ong, Qi Wang, and Ke Tang. 2024. Large language models can be guided to evade AI-generated text detection. *Transactions on Machine Learning Research*.

Narek Maloyan, Bulat Nutfullin, and Eugene Ilyshin. 2022. Dialog-22 ruatd generated text detection. In *Computational Linguistics and Intellectual Technologies*, page 394–401. RSUH.

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

Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 897–908.

Meta. 2024. Llama 3. https://llama.meta.com/llama3/.

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.

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *arXiv Preprint arXiv:2301.13852*, abs/2301.13852.

Edoardo Mosca, Mohamed Hesham Ibrahim Abdalla, Paolo Basso, Margherita Musumeci, and Georg Groh. 2023. Distinguishing fact from fiction: A benchmark dataset for identifying machine-generated scientific papers in the llm era. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 190–207, Toronto, Canada. Association for Computational Linguistics.

Gustave Florentin Nkoulou Mvondo, Ben Niu, and Salman Eivazinezhad. 2023. Generative conversational ai and academic integrity: A mixed method investigation to understand the ethical use of llm chatbots in higher education. *Available at SSRN 4548263*.

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.

Olympics. 2024. Olympics. https://olympics.com/en/.

OpenAI. 2020. Chatgpt. https://openai.com/chatgpt.

OpenAI. 2024. Gpt-4o. https://openai.com/index/hello-gpt-4o/.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Yang Wang. 2023. On the risk of misinformation pollution with large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023. Deepfake text detection: Limitations and opportunities. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1613–1630. IEEE.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv Preprint arXiv:1606.05250*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv Preprint arXiv:2403.05530*.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, volume 6, pages 199–205.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. 2023. The curse of recursion: Training on generated data makes models forget. *arXiv Preprint arXiv:2305.17493*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for

semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv Preprint arXiv:1908.09203*.

Daniel Spokoyny, Tanmay Laud, Tom Corringham, and Taylor Berg-Kirkpatrick. 2023. Towards answering climate questionnaires from unstructured climate reports. *arXiv Preprint arXiv:2301.04253*.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: A family of highly capable multimodal models. *arXiv Preprint arXiv:2312.11805*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv Preprint arXiv:2302.13971*.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. In *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.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in Neural Information Processing Systems*, 32.

Yizhou Zhang, Karishma Sharma, Lun Du, and Yan Liu. 2024. Toward mitigating misinformation and social media manipulation in llm era. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1302–1305.

Lucia Zheng, Neel Guha, Brandon R Anderson, Peter Henderson, and Daniel E Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 159–168.

# A Appendix

## A.1 Dataset Details

Our dataset encompasses 21 independent English domains. Table 6 shows the source and license for each domain. For all domains, we manually verify that no personal or offensive content is included. For domains taken from third-party datasets, we use the data consistent with their intended use (detection of LLM-generated text).

## A.2 Effectiveness of the Diverse Prompt in Data Preparation

The construction of our dataset involves 200 generation prompts, resembling more real-world use cases compared with traditional evaluation datasets which are usually constrained to one single generation prompt. To prove the superiority of our dataset in training more capable detection models, we create a parallel non-diverse dataset which is created on the same number of domains and source LLMs, but the LLM data is generated only using the prompt "`Rewrite this for me please.`" Then, we train two RAIDAR detectors without fine-tuning, on the non-diverse dataset, and evaluate it on the diverse dataset. As shown in Table 8, the diverse prompts yields to 2.64% increase in AUROC score if the rewrite model is Gemini 1.5 Pro, and 0.82% increase in AUROC score if the rewrite model is Llama-3 8B. This validates the effectiveness of the diverse prompts we were using, and suggests that such diversity could help the detector to capture more information about real world data distributions.

## A.3 LoRA Configurations for Fine-Tuning

We leverage QLoRA when fine-tuning L2R and the baselines. Despite the same quantization precision, Table 9 lists the four different LoRA configurations that we use for fine-tuning in §5.4.

## A.4 Effectiveness of the Calibration Loss

An important contribution of ours that improves the fine-tuning performance is thresholding on the calibration loss, as proposed in §3.4. Without this method, the model tends to overfit during fine-tuning as shown in Figure 4, where the model loss drastically decrease after 1500 steps, resulting in verbose rewrite even for LLM-generated text. We find the overfitting harms L2R's performance from an ablation study on five domains where the AUROC score is only 0.62 after the model overfits.

The calibration loss can benefit model learning because the threshold effectively prevents further modification to model weights once an input, labeled either LLM or human, falls in its corresponding distribution already. Since our purpose is simply to draw a boundary rather than separate the distributions as much as possible, halting further weight updates on already correctly classified inputs allows the model to focus parameter updates only on misclassified examples, leading to more efficient and effective convergence. Concretely, applying the calibration loss improves the L2R's performance by 4.54% in AUROC among 21 domains, even comparing to a model tuned without the calibration and before it overfits.

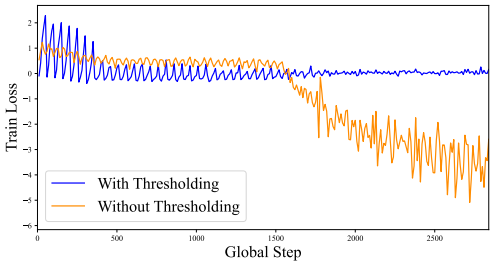


Figure 4: Training loss curves for the rewrite model. The orange line plots the loss trained without the calibration method, and the blue line plots the loss trained with calibration. The later one exhibits faster convergence and higher stability than the former one.

## A.5 Different Ways to Generate OOD Data

There exists a variety of ways to generate OOD data, including using different generation models, decoding strategies, text lengths, and writing styles. While we show how M4, the OOD dataset we use for evaluation, deviates from our training domain in all above aspects in Table2, we conduct two additional ablation studies on how different text length and decoding strategy alone could influence detection performance.

We use 200 randomly selected human-written texts from our dataset for both studies. For the study on decoding strategy, we use greedy decoding for GPT and Gemini models and beam search with num_beams=5 for the Llama model during the construction of the LLM-generated counterparts. For the study on text length, we chunk the input texts to an average length of 60. We test different detectors on these two datasets and show results in Table 7, where L2R outperforms the others across both settings. These results further confirm L2R's

| Category | Source | License |
|---|---|---|
| AcademicResearch | Arxiv abstracts (Mao et al., 2024) | Various CC licenses |
| ArtCulture | Wikipedia | CC BY-SA |
| Business | Wikipedia | CC BY-SA |
| Code | Code snippets (Mao et al., 2024) | MIT |
| EducationalMaterial | Ghostbuster essays (Verma et al., 2024) | CC BY 3.0 |
| Entertainment | IMDb dataset (IMDb, 2024), Stanford SST2 (Socher et al., 2013) | IMDb terms of use, CC Zero |
| Environmental | Climate-Ins (Spokoyny et al., 2023) | CC Zero |
| Finance | Hugging Face FIQA (Thakur et al., 2021) | CC BY-NC |
| FoodCuisine | Kaggle fine food reviews (McAuley and Leskovec, 2013) | CC Zero |
| GovernmentPublic | Wikipedia | CC BY-SA |
| LegalDocument | CaseHOLD (Zheng et al., 2021) | Apache 2.0 |
| CreativeWriting | Writing Prompts (Fan et al., 2018) | MIT |
| MedicalText | PubMedQA (Jin et al., 2019) | MIT |
| NewsArticle | XSum (Narayan et al., 2018) | MIT |
| OnlineContent | Hugging Face blog authorship (Schler et al., 2006) | Non-commercial |
| PersonalCommunication | Hugging Face daily dialogue (Li et al., 2017) | CC-BY-NC-SA 4.0 |
| ProductReview | Yelp reviews (Mao et al., 2024) | Yelp terms of use |
| Religious | Bible, Buddha, Koran, Meditation, and Mormon | N/A |
| Sports | Olympics website (Olympics, 2024) | Olympics terms of use |
| TechnicalWriting | Scientific articles (Mosca et al., 2023) | CC Zero |
| TravelTourism | Wikipedia | CC BY-SA |

Table 6: Source and license for each of the 21 domains in our dataset.

| Avg Length | Decoding Strategy | Fast-DetectGPT | RAIDAR | L2R |
|---|---|---|---|---|
| 120 | Nucleus Sampling | 0.6833 | 0.8186 | **0.9213** |
| 60 | Nucleus Sampling | 0.6500 | 0.7635 | **0.8632** |
| 120 | Greedy Decoding & Beam Search | 0.6897 | 0.8009 | **0.8750** |

Table 7: Comparison of AUROC scores across the three methods under various OOD settings. L2R consistently outperforms the baselines, achieving the highest AUROC in all scenarios.

| Dataset | Rewrite Model | AUROC |
|---|---|---|
| Single-Prompt | Gemini | 0.7302 |
| Multi-Domain Dataset | Llama | 0.7888 |
| Multi-Prompt | Gemini | **0.7566** |
| Multi-Domain Dataset | Llama | **0.7970** |

Table 8: Comparison of AUROC scores of the Gemini and Llama rewrite models when trained on non-diverse versus diverse datasets. Incorporating diverse prompting during training leads to improved detection performance for both models on the same diverse test set.

| r | lora_alpha | Trainable Parameters |
|---|---|---|
| 2 | 4 | 851,968 |
| 4 | 8 | 1,703,936 |
| 8 | 16 | 3,407,872 |
| 16 | 32 | 6,815,744 |

Table 9: Parameter settings for LoRA fine-tuning.

robustness against different types of OOD data distributions.

### A.6 Rewrite Examples

We show the text examples before and after L2R's rewriting, among multiple domains, in Figure 5.

| | | |
|---|---|---|
| **Academic Research** | **INPUT HUMAN**<br>In recent years, several techniques have been proposed for increasing robustness to adversarial examples --- and yet most of these have been quickly shown to be vulnerable to future attacks. For example, over half of the defenses proposed by papers accepted at ICLR 2018 have already been broken. | **REWRITTEN HUMAN**<br>Despite recent efforts to increase robustness, most proposed techniques have been quickly shown to be vulnerable to future attacks. For instance, over half of the defenses presented at ICLR 2018 have already been breached. |
| | **INPUT LLAMA-3-70B**<br>Despite numerous attempts to develop robustness against these threats, most proposed defenses have been rapidly compromised by subsequent attacks. In fact, a staggering 50% of defenses presented at ICLR 2018 have already been breached. | **REWRITTEN LLAMA-3-70B**<br>Despite numerous attempts to develop robustness against these threats, most proposed defenses have been rapidly compromised by subsequent attacks. In fact, a staggering 50% of defenses presented at ICLR 2018 have already been breached. |
| **Code**<br><br>Indentation hidden | **INPUT HUMAN**<br>def solve(s): flg = 0 idx = 0 new_str = list(s) for i in s: if i.isalpha(): new_str[idx] = i.swapcase() flg = 1 idx += 1 s = "" for i in new_str: s += i if flg == 0: return s[len(s)::-1] return s | **REWRITTEN HUMAN**<br>def solve(s): result = [] for char in s: if char.isalpha(): result.append(char.swapcase()) else: result.append(char) return "".join(result) if not any(char.isalpha() for char in s) else "".join(result)[::-1] |
| | **INPUT GPT-4o**<br>def solve(s): has_letter = any(char.isalpha() for char in s) if has_letter: return ''.join(char.swapcase() if char.isalpha() else char for char in s) else: return s[::-1] | **REWRITTEN GPT-4o**<br>def solve(s): has_letter = any(char.isalpha() for char in s) if has_letter: return ''.join(char.swapcase() if char.isalpha() else char for char in s) else: return s[::-1] |
| **Legal Document** | **INPUT HUMAN**<br>This type of information may constitute trade secrets. See G.L.c. 266, §30 (defining “trade secret” as used in G.L.c. 93, §42 | **REWRITTEN HUMAN**<br>This type of information may constitute trade secrets. Massachusetts General Laws, chapter 266, section 30 |
| | **INPUT GEMINI 1.5 PRO**<br>This type of information, such as customer data and proposals, can be legally protected as trade secrets under Massachusetts law (G.L.c. 266, §30). | **REWRITTEN GEMINI 1.5 PRO**<br>This type of information, such as customer data and proposals, can be legally protected as trade secrets under Massachusetts law (G.L. c. 266, § 30). |
| **Creative Writing** | **INPUT HUMAN**<br>A voice in the sky will tell you when you've left a place for the last time. You tell your SO goodbye and head out to [ location of your choice ]. Halfway there you hear "You have visited a location for the last time." John wrinkled his nose and cleared his throat. Pausing, he put down his briefcase and adjusted his tie. It was a red tie with white stripes. This was one of his favorite ties. | **REWRITTEN HUMAN**<br>As John bid his SO farewell, he headed out to [location of his choice]. Halfway there, a voice in his mind whispered, "You have visited this place for the last time." John's expression faltered, and he paused to adjust his tie, a favorite red tie with white stripes. |
| | **INPUT LLAMA-3-70B**<br>As John bid farewell to his partner and headed out to the city, a mysterious voice in the sky announced, "You have visited a location for the last time." He paused, adjusting his favorite red tie with white stripes, and cleared his throat. | **REWRITTEN LLAMA-3-70B**<br>As John bid farewell to his partner and stepped out into the city, a mysterious voice in the sky announced, "You have visited a location for the last time." He paused, adjusting his favorite red tie with white stripes, and cleared his throat. |

Figure 5: Examples of texts in our proposed dataset along with the amount of edits L2R model gives for human and LLM-generated text. Deleted characters are marked in red, inserted characters are in blue, and unmodified characters are in **black**. The examples demonstrate the diverse domains and source LLMs available in the dataset, as well as L2R's ability in separating human and LLM-generated text via rewriting.