# On the Generalization and Adaptation Ability of Machine-Generated Text Detectors in Academic Writing

Yule Liu*
Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
yliu514@connect.hkust-gz.edu.cn

Zhiyuan Zhong*
Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, Guangdong, China
zhiyuanzhong@hkust-gz.edu.cn

Yifan Liao
National University of Singapore (Chongqing Research Institute)
Chongqing, China
yifan.liao@nus.edu.sg

Zhen Sun
Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, Guangdong, China
zsun344@connect.hkust-gz.edu.cn

Jingyi Zheng
Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, Guangdong, China
jzheng029@connect.hkust-gz.edu.cn

Jiaheng Wei
Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, Guangdong, China
jiahengwei@hkust-gz.edu.cn

Qingyuan Gong
Fudan University
Shanghai, China
gongqingyuan@fudan.edu.cn

Fenghua Tong
Qilu University of Technology
Jinan, Shandong, China
tongfh@qlu.edu.cn

Yang Chen
Fudan University
Shanghai, China
chenyang@fudan.edu.cn

Yang Zhang
CISPA Helmholtz Center for Information Security
Saarbrücken, Saarland, Germany
zhang@cispa.de

Xinlei He†
Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, Guangdong, China
xinleihe@hkust-gz.edu.cn

## Abstract

The rising popularity of large language models (LLMs) has raised concerns about potential abuse and harmful content. As a result, developing a highly generalizable and adaptable machine-generated text (MGT) detection system has become an urgent priority. Given that LLMs are most commonly misused in academic writing, this work investigates the generalization and adaptation capabilities of MGT detectors in three key aspects specific to academic writing: First, we construct MGT-Academic, a large-scale dataset comprising over 336M tokens and 749K samples. MGT-Academic focuses on academic writing, featuring human-written texts (HWTs) and MGTs across STEM, Humanities, and Social Sciences, paired with an extensible code framework for efficient benchmarking. Second, we benchmark the performance of various detectors for binary classification and text attribution tasks in both in-domain and cross-domain settings. This benchmark reveals the often-overlooked challenges of text attribution tasks. Third, we introduce a novel text attribution task in which models must adapt to new classes over time, with little or no access to prior training data, spanning both few-shot and many-shot scenarios. We implement a range of adaptation techniques to enhance performance across these settings. Our findings provide new insights into the generalization ability of MGT detectors and lay the foundation for building robust, adaptive detection systems. The code framework is available at https://github.com/Y-L-LIU/MGTBench-2.0.

## CCS Concepts

- **Computing methodologies → Natural language generation**;
- **General and reference → Evaluation**.

## Keywords

Large Language Model; Machine-Generated Text Detection

**ACM Reference Format:**
Yule Liu, Zhiyuan Zhong, Yifan Liao, Zhen Sun, Jingyi Zheng, Jiaheng Wei, Qingyuan Gong, Fenghua Tong, Yang Chen, Yang Zhang, and Xinlei He. 2025. On the Generalization and Adaptation Ability of Machine-Generated Text Detectors in Academic Writing . In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737408

*Both authors contributed equally to this work.
†Corresponding author.

*KDD '25, Toronto, ON, Canada*

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737408

## 1 Introduction

Recent advancements in large language models (LLMs) showcase their strong ability to tackle a wide range of natural language processing (NLP) tasks [20, 28, 29]. Its versatility and superiority across numerous domains unlock remarkable real-world applications, e.g., education, idea crafting, and context refinement [48]. However, the ease and convenience have opened the door to abuse, particularly in academic writing [42], social media [38], and web search [21], leading to severe ethical and practical challenges. Additionally, recent work [14, 19, 37] reveals multiple security vulnerabilities of LLMs. To audit the potential abuse and harm contents, recent efforts [4, 25] have focused on developing techniques to distinguish machine-generated text (MGT) from human-written text (HWT) and benchmarking their performance [13, 41] over a wide range of datasets. Existing work mainly focuses on detecting whether a given text is MGT or HWT, which is referred to as *binary classification task* in this paper. However, it remains unclear how detectors designed for binary classification perform and generalize in identifying the specific source LLM that generated the text, which is referred to as *text attribution task* in this paper.

To holistically evaluate and understand the generalization and adaptation ability of existing detectors, we construct a large-scale MGT dataset named MGT-Academic focusing on academic writing, comprising over 336M tokens and 749K samples from 5 LLMs and 3 academic domains: STEM, Humanities, and Social Sciences. Within each domain, we collect HWT data from Wikipedia and academic texts sourced, including Arxiv or Project Gutenberg, depending on the scenario. Each HWT has corresponding MGTs generated by five popular LLMs. Further, we build a publicly available, extendable, and user-friendly code framework for the community, which enables fast and effective benchmarking for existing methods in binary classification and text attribution tasks. It covers state-of-the-art methods, including seven metric-based detectors and five model-based detectors.

Leveraging MGT-Academic, we conduct a comprehensive investigation into the performance and generalization ability, especially for text attribution tasks. First, we benchmark the performance of existing detectors in both binary classification and text attribution tasks. The results in the text attribution task bring new insights that metric-based detectors (detecting MGT using statistical metrics such as log-likelihood or rank [8]) fail and achieve only random-guess results. We analyze the underlying reasons for failure and emphasize the need for developing generalizable detectors. Second, we evaluate the transferability of detectors in task attribution to examine how detectors transfer to specific domains, i.e., STEM, Humanities, and Social Sciences[1]. To enhance the transferability, we evaluate important techniques such as adding examples from the target domains.

Third, since new LLMs are continuously released, each with different characteristics and unique stylistics, we propose a new attribution task, where a model should adapt to the new class introduced over time without (or with very limited) access to the original training data for earlier classes. This task is crucial for real-world applications where MGTs from new LLMs become available in stages, and retraining the model from scratch is impractical due to computational or data storage constraints. In this paper, we consider two practical settings for adaptation, i.e., few-shot and many-shot settings, depending on the number of accessible examples in the new class. To the best of our knowledge, we are the first to discuss how the detectors adapt to new MGTs in detection. We benchmark the performance of detectors when adapting to new LLMs in different scenarios and equip the detector with eight different techniques to improve the performance. In summary, our contributions can be listed as follows:

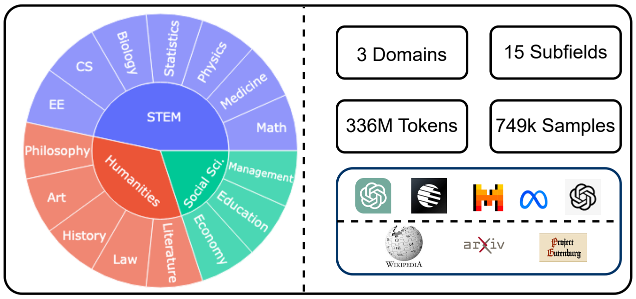

**Figure 1: Overview of** MGT-Academic**. It shows the data source and split domains.**

- We introduce MGT-Academic, a large-scale MGT dataset focused on academic writing, encompassing over 336M tokens and 749K samples from 5 LLMs and 3 academic domains: STEM, Humanities, and Social Sciences. We additionally provide an extensible code framework, which will be made publicly available, to efficiently benchmark existing MGT detectors in different tasks.
- We conduct extensive experiments and benchmark the performance of various detectors for binary classification and attribution tasks in both in-domain and cross-domain settings. The failure of metric-based detectors in the text attribution task demonstrates the inherent complexity of the attribution task. Our study emphasizes the need for developing metric-based detectors that can adapt to text-attribution tasks.
- We introduce a new attribution task where detectors adapt to new classes over time without (or with very limited) access to prior training data in few-shot and many-shot scenarios. We benchmark the performance of model-based detectors with multiple adaptation techniques. Despite the improved performance, the remaining gap to the ideal performance (fine-tuning all data at once) highlights the complexity of this task, underscoring the need for further investigation in the future.

## 2 Related Work

**MGT Detection.** Recent advancements in LLMs have empowered users to tackle a wide range of NLP tasks, demonstrating their versatility and superiority across numerous domains [28, 28, 29]. Exploiting LLMs is especially convenient in academic writing [23, 48], such as generating ideas, drafting articles, or refining content. However, the ease and convenience can be significantly abused, raising concerns about authenticity, as well as ethical questions regarding

[1] Due to the page limit, check results for the binary classification task in our arxiv version.

originality and over-dependence on AI-generated content [7]. To prevent the misuse of MGT data, recent studies [3, 8, 10, 13, 25] have developed a variety of MGT detectors, which can be broadly categorized into metric-based and model-based methods. Metric-based methods [8, 25, 36] leverage proxy LLMs to extract features from processed text and train an additional classifier to model the relationship between features and labels. In contrast, model-based methods [10, 16] typically integrate a classification head into a BERT model and fine-tune the augmented model on supervised datasets. The detectors in this paper are listed in Appendix C.

Several efforts have aimed to benchmark the performance of MGT detectors. For example, MGTBench [13] provided a comprehensive evaluation framework for these detectors, which utilizes existing HWT datasets, including Essay, WP, and Reuters. M4GTBench [42] extended this by benchmarking performance on multilingual and multi-source datasets. While existing studies emphasize transferability across datasets and LLMs in binary classification, they pay less attention to the generalization ability of detectors in attribution tasks.

**Adapting to New Classes.** Adapting to new classes with few-shot settings is related to few-shot learning, aiming to improve the quick adaptation ability. One way is to use the distance between the samples and the representatives of each class for classification [5, 35]. Another way is to train a neural network and learn the relationship between samples and the representatives [39]. Additionally, data augmentation is used to increase the number of training samples and train a classifier [47] Adapting to new classes with many-shot settings is related to class incremental learning (CIL), where the key is to alleviate the forgetting of previous knowledge [22, 50]. One way to improve the performance is to incorporate a distillation loss or regularization term to transfer knowledge from the old model to the updated one, thus reducing forgetting [17, 33]. Another way is to store a small subset of past representative data to enable the model to rehearse earlier tasks [31]. Additionally, some other work explores calibrating the output layer of the classification head to improve the performance [45].

Some efforts have discussed the CIL in classification tasks such as entailment or intent classification [30, 46]. To the best of our knowledge, our work is the first to benchmark the adaptation ability of MGT detectors in both few-shot and many-shot settings.

## 3 Construction of MGT-Academic

### 3.1 MGT-Academic Collection

We collect 749,625 samples with 336,714,335 tokens.

**Human Data.** We collect data in three academic domains, i.e., STEM, Social Sciences, and Humanities, where each domain contains different fine-grained fields. For each specific academic field, we specify the category parameter and query the APIs of the data sources, followed by merging the retrieved data back to the affiliated domain. Specifically, every domain consists of Wiki data and contents collected from Arxiv (LaTeX code of pre-print papers) or Project Gutenberg (published e-books), depending on the scenario. More details are shown in Table A1. This method allows us to systematically map the datasets to the STEM, humanities, and social sciences sub-domains.

**Table 1: Linguistic Metrics of MGT-Academic: FSE shows the readability, with higher scores indicating easier readability. TTR shows the text diversity, with higher scores indicating more unique tokens in the text. ASL shows the syntactic complexity, with higher scores indicating more words within a sentence.**

| Metric | Human | Moonshot | GPT-3.5 | Mixtral | Llama3 | GPT-4omini |
|---|---|---|---|---|---|---|
| FSE | 39.74 | 34.41 | 36.56 | 38.39 | 36.14 | 31.46 |
| TTR | 0.52 | 0.53 | 0.60 | 0.52 | 0.53 | 0.58 |
| ASL | 23.31 | 22.16 | 21.73 | 19.39 | 21.26 | 20.76 |

**Machine Data.** We select five widely used LLMs, including Llama-3.1-70b-Instruct [9], Mixtral-8×7b-Instruct [24], KimiChat [26], ChatGPT, GPT-4omini [27] to generate the MGTs. Llama-3.1-70b-Instruct and Mixtral-8×7b-Instruct are two commonly used open-source LLMs that exploit dense and MoE architecture respectively. Moonshot, ChatGPT, and GPT-4omini are popular proprietary models, with Moonshot known for its long-context understanding and the GPT family recognized for its comprehensive capabilities. We prompt the LLM to be a wiki/paper/book editor and polish the given human text, which is consistent with the previous dataset [25, 42]. The prompts for generating MGT data are listed in Appendix A.

**Linguistic Metrics.** We compute a set of linguistic metrics, including readability, lexical diversity, and syntactic complexity, to assess the quality of the generated text. The results are summarized in Table 1. Regarding readability , we use the Flesch Reading Ease (FSE) score [6], which estimates how difficult a text is to understand. Higher scores indicate easier readability. We find that MGTs are generally more challenging to read, likely due to the complex and varied expressions used by LLMs. The average FSE score of 35 falls within the college-level readability range (30–60) and is close to the graduate-level range (0–30). Regarding lexical diversity , we use Type-Token Ratio (TTR), which reflects the ratio of unique words to total words in a text. A higher TTR indicates richer vocabulary usage. Our results show that MGTs generally exhibit higher TTR than HWTs, implying that machine-generated texts tend to employ a broader lexical range. Regarding syntactic complexity, we calculate the average sentence length (ASL), which serves as a proxy for structural complexity. Higher ASL values indicate longer, more intricate sentences. We observe that MGTs typically have shorter sentence lengths compared to HWTs, suggesting that machine-generated texts favor simpler and more direct syntactic structures.

**Data Quality Concerns.** We have several measures to ensure data quality. First, we ensure the papers and articles collected from Arxiv and Wiki are posted before December 2023 (release date of ChatGPT) to improve the reliability of human data. Second, we utilize SentenceBert [32] to measure the editing distance in each doman and further adopt TSNE [40] to project the high-dimensional representations into two dimensions. The projections are shown in Figure A1, indicating the revised MGTs from HWTs are non-trivial and of high quality. Third, despite the relatively high performance in the binary classification task, the reported data aligns well with the previous work [13, 42]. In contrast, the degraded performance in the other task reveals the complexity of the proposed dataset.

**Table 2: Experiment Result for In-distribution Binary Classification. We train and test the detectors on the same data domain. The results are reported using the F1-score. ST. represents STEM, Hu. represents Humanity, and So. represents Social Science. The larger values with blue colors indicate better performance, and the lower values with red colors indicate weaker performance. For the abnormal results, we use "-" as the placeholder.**

| Method | Llama-3.1-70b | | | Mixtral-8x7b | | | MoonShot-8k | | | GPT-4o-mini | | | GPT-3.5 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2-16 | ST. | Hu. | So. | ST. | Hu. | So. | ST. | Hu. | So. | ST. | Hu. | So. | ST. | Hu. | So. |
| LL | 0.714 | 0.794 | 0.803 | 0.662 | 0.749 | 0.809 | 0.711 | 0.760 | 0.806 | 0.638 | 0.689 | 0.765 | 0.481 | 0.606 | 0.709 |
| Entropy | 0.759 | 0.707 | 0.829 | 0.688 | 0.659 | 0.795 | 0.723 | 0.662 | 0.780 | 0.683 | 0.686 | 0.763 | 0.679 | 0.640 | 0.700 |
| Rank | 0.618 | 0.697 | 0.713 | 0.617 | 0.695 | 0.719 | 0.685 | 0.750 | 0.817 | 0.643 | 0.651 | 0.627 | 0.678 | - | 0.571 |
| Log-Rank | 0.736 | 0.709 | 0.795 | 0.655 | 0.596 | 0.732 | 0.688 | 0.650 | 0.773 | 0.639 | 0.615 | 0.704 | 0.648 | 0.591 | 0.708 |
| Rank-GLTR | 0.720 | 0.759 | 0.802 | 0.655 | 0.701 | 0.808 | 0.600 | 0.734 | 0.795 | 0.658 | 0.693 | 0.713 | 0.620 | 0.679 | 0.694 |
| Fast-DetectGPT | 0.817 | 0.817 | 0.887 | 0.760 | 0.759 | 0.842 | 0.842 | 0.801 | 0.899 | 0.688 | 0.718 | 0.752 | 0.677 | 0.713 | 0.756 |
| Binoculars | 0.881 | 0.897 | 0.911 | 0.833 | 0.845 | 0.890 | 0.923 | 0.867 | 0.916 | 0.710 | 0.772 | 0.800 | 0.680 | 0.803 | 0.792 |
| RADAR | 0.800 | 0.834 | 0.743 | 0.747 | 0.833 | 0.757 | 0.771 | 0.852 | 0.750 | 0.762 | 0.855 | 0.778 | 0.771 | 0.897 | 0.814 |
| ChatGPT-D | 0.557 | 0.552 | 0.712 | 0.452 | 0.526 | 0.642 | 0.531 | 0.643 | 0.743 | 0.280 | 0.320 | 0.454 | 0.458 | 0.679 | 0.625 |
| DistillBert-F | 0.987 | 0.983 | 0.971 | 0.977 | 0.983 | 0.976 | 0.988 | 0.991 | 0.990 | 0.983 | 0.988 | 0.982 | 0.983 | 0.979 | 0.966 |
| Roberta-F | 0.987 | 0.994 | 0.994 | 0.992 | 0.997 | 0.995 | 0.993 | 0.992 | 0.997 | 0.987 | 0.993 | 0.994 | 0.986 | 0.986 | 0.981 |
| DeBETRTa-F | 0.988 | 0.990 | 0.971 | 0.987 | 0.989 | 0.987 | 0.993 | 0.988 | 0.997 | 0.989 | 0.992 | 0.992 | 0.987 | 0.980 | 0.980 |

**Table 3: Experiment Result for In-distribution Text Attribution. We train and test the model on the same data domain. The results are reported using the F1-score. The larger values with blue colors indicate better performance and lower values with red colors indicate weaker performance.**

| | LL | Entropy | Rank | Log-Rank | Rank-GLTR | Fast-Detect | Binoculars | DistillBert-F | Roberta-F | DeBETRTa-F |
|---|---|---|---|---|---|---|---|---|---|---|
| STEM | 0.219 | 0.158 | 0.194 | 0.214 | 0.166 | 0.139 | 0.141 | 0.844 | 0.888 | 0.868 |
| Humanities | 0.148 | 0.118 | 0.142 | 0.225 | 0.181 | 0.148 | 0.133 | 0.784 | 0.815 | 0.790 |
| Social Science | 0.214 | 0.189 | 0.228 | 0.285 | 0.215 | 0.162 | 0.191 | 0.817 | 0.818 | 0.810 |

### 3.2 Code Framework

Our framework follows the factory design pattern and implements *AutoDetector* and *AutoExperiment* for abstraction, which is aligned with the approach used in Huggingface Transformers [43], the most widely used library in the NLP community. It provides an easy-to-use and extendable code framework for the community and is publicly available.

### 3.3 Data Moderation

**Human Split.** To ensure data quality, we apply a rigorous cleaning process to remove noise and irrelevant content. We discard texts with fewer than 50 words and ensure that all entries start and end with complete sentences, preserving their coherence and clarity. As shown in Table A2, we filter out data with the given keywords to avoid duplication and improve readability.

**Machine Split.** To moderate the machine-generated data, we focus on removing the obvious identifiers for text detection. First, we remove the text of short length below 50 words (split by space), which is usually produced by failed or incomplete API queries. Second, we make sure every data entry ends with complete sentences to avoid easy detection. Third, we customize different keywords filtering rules for MGTs in Table A2.

## 4 In-distribution Performance

**Experiment Settings.** Our setting assumes the training and testing data are from the same domain. For each domain, we first sample the same number of HWTs and MGTs from the corresponding domains, then randomly split them into the train/test dataset with an 80%/20% ratio. Regarding the evaluation metric, we report the F1-score, an imbalance-robust metric that balances precision and recall. More experimental details are shown in Appendix D.

**Detectors.** In this setting, we benchmark both metric-based and model-based detectors on MGT-Academic. For metric-based detectors, we evaluate Log-Likelihood, Entropy, Rank, Rank-GLTR [8], LRR [36], Fast-DetectGPT [4], and Binoculars [11]. Specifically, we utilize Llama-2-7B-chat as the metric generator to find an optimal threshold to maximize the F1-score for binary classification and train a logistic regression classifier for text attribution. For model-based detectors, we include RADAR [15], ChatGPT-D [10], DistillBERT [34], RoBERTa [18], and DeBerta-v3 [12]. We use the officially released model weights for RADAR and ChatGPT-D, while fine-tuning DistillBERT, RoBERTa, and DeBerta on MGT-Academic's training data. More details are provided in Appendix C.

**Binary Classification Task.** The performance of detectors in binary classification task is shown in Table 2 and we have several

**Table 4: Experiment Result for Transferring Across Domains in Text Attribution Task. We train the model on data in one domain and test the model on another domain. The results are reported using the F1-score. The larger values with blue colors indicate better performance, and the lower values with red colors indicate weaker performance.**

| | | Hu. | ST. | So. | | | Hu. | ST. | So. | | | Hu. | ST. | So. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LL | Hu. | 0.181 | 0.160 | 0.204 | Binoculars | Hu. | 0.148 | 0.155 | 0.182 | FastDetect | Hu. | 0.225 | 0.221 | 0.250 |
| | ST. | 0.158 | 0.166 | 0.186 | | ST. | 0.195 | 0.219 | 0.244 | | ST. | 0.213 | 0.214 | 0.247 |
| | So. | 0.176 | 0.185 | 0.215 | | So. | 0.178 | 0.191 | 0.214 | | So. | 0.251 | 0.241 | 0.285 |
| DeBERTa-F | Hu. | 0.790 | 0.689 | 0.764 | DistillBert-F | Hu. | 0.784 | 0.592 | 0.715 | Roberta-F | Hu. | 0.815 | 0.674 | 0.762 |
| | ST. | 0.706 | 0.868 | 0.812 | | ST. | 0.651 | 0.844 | 0.781 | | ST. | 0.727 | 0.883 | 0.814 |
| | So. | 0.730 | 0.816 | 0.810 | | So. | 0.718 | 0.828 | 0.817 | | So. | 0.727 | 0.827 | 0.818 |

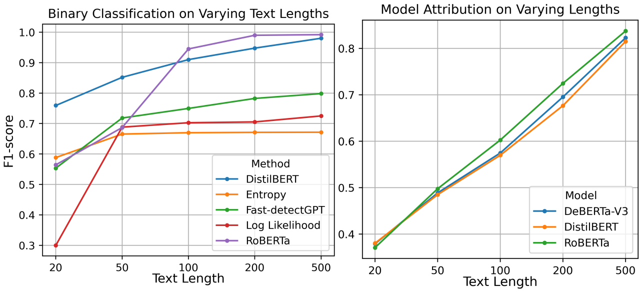

**Figure 2: The F1-score of different detectors under texts of varying lengths.**

observations. First, we find that supervised model-based detectors consistently outperform other methods, achieving F1-scores above 0.98. This advantage is largely due to the availability of extensive supervised training data, enabling these detectors to learn highly effective classification boundaries. Second, the detectors with offically released weights, e.g., RADAR and ChatGPT-D, show relatively poor performance. The degradation demonstrates that detecting MGTs in MGT-Academic is non-trival and further remarks the generalization problem in developing detectors. Third, while the state-of-the-art (SOTA) metric-based detectors, such as Fast-DetectGPT and Binoculars, show very competitive zero-shot performance in most cases, identifying MGTs from GPT-4omini and GPT-3.5 appears to be a challenging task. This indicates Llama-2-7b-chat fails to extract distinguishable features from MGT-Academic.

**Text Attribution Task.** The performance of detectors in text attribution task is shown in Table 3 and we have several observations. Notably, RADAR and ChatGPT-D are excluded from this evaluation because their publicly released classification heads are designed specifically for binary classification. First, we observe that while fine-tuning model-based detectors can yield competitive results, their performance drops significantly compared to their near-perfect accuracy in binary classification tasks. This decline suggests that transitioning from binary to multi-class classification poses a notable challenge, despite being a natural extension.

Second, surprisingly, metric-based detectors exhibit almost no capability in text attribution, performing at a level close to random guessing. Given that the feature space in these detectors is typically one-dimensional (e.g., log-likelihood, logrank, and entropy), it is not surprising that they struggle in multi-class settings. To better understand and address this limitation, we compare the performance of vanilla SOTA metric-based methods (FastDetectGPT and Binocular) with a combined metric approach, which concatenates five different features from existing metric-based detectors, i.e., ll, logrank, entropy, FastDetectGPT, and Binoculars. The results, shown in Table 5, reveal a significant performance improvement, supporting our hypothesis that the low dimensionality of traditional metrics limits their effectiveness and highlighting the potential of richer metric combinations for text attribution. These findings highlight that text attribution is a critical yet underexplored task.

**Table 5: Experiment result for combining the metrics in attribution task. The results are reported using F1-score.**

| Detector | STEM | Humanities | Social Science |
|---|---|---|---|
| Fast-Detect | 0.2137 | 0.2251 | 0.2849 |
| Binoculars | 0.2188 | 0.1483 | 0.2141 |
| Combined | **0.3528**$_{\uparrow 61\%}$ | **0.3560**$_{\uparrow 88\%}$ | **0.3624**$_{\uparrow 36\%}$ |

Additionally, we conduct an ablation study on the effect of text length on detectors' performance in Figure 2. The binary classification requires roughly 100 words to have decent performance, while the attribution tasks require more words.

**Takeaways.** We benchmark the performance of detectors in binary classification and text attribution tasks. For the binary classification task, model-based detectors consistently perform better than metric-based detectors with the same distributions in train and test data. For the attribution task, the model-based detectors show competitive performance while the metric-based detectors perform poorly (near random guessing), which is caused by the low dimension of the features. Future works are encouraged to develop metric-based detectors suitable for attribution tasks.

## 5 Generalization in Domain Transferring

**Experiment Settings.** Our setting assumes that the training and testing data come from different domains, while maintaining an 80%/20% train-test split. Specifically, we only consider the transfer detectors across domains in the text attribution task. Regarding the evaluation metric, we report the F1-score. The detailed experimental details are shown in Appendix D. We select representative detectors,

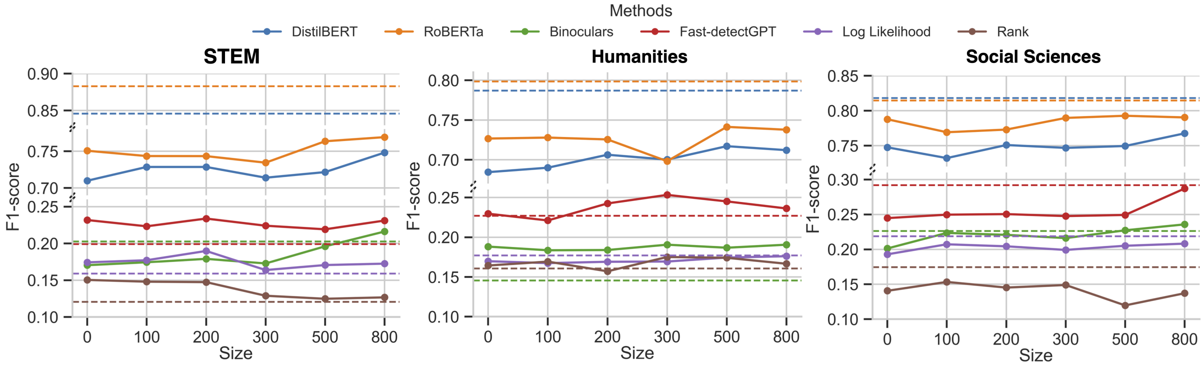

**Figure 3: Mitigation Result for Domain Transferring. The line plots illustrate how detector performance changes as data from the target domains is incrementally added. Dashed lines represent the ideal performance achieved when the detector is trained specifically on the target domain.**

i.e., LL, FastDetectGPT, Binoculars for metric-based detectors, as well as DistillBert, RoBERTa, and DeBerta-v3 in supervised model-based detectors.

**Transferring in Attribution Task.** The performance of transferring across domains in text attribution task is summarized in Table 4, and we have several observations. First, despite the generally poor performance of metric-based detectors, transferring these detectors to other domains occasionally results in improved accuracy. This unexpected outcome may stem from the limited expressiveness or inefficiency of the extracted metrics, which might inadvertently align better with certain domain-specific characteristics. Second, although supervised model-based detectors achieve competitive results compared to metric-based ones, cross-domain transfer remains a significant challenge.

**Mitigating Techniques.** Given the limited transferring ability, we add limited examples from the target domain to the training data. Figure 3 reports how the performance evolves when adding more data. On average, performance improves by 3.7%; however, the gains remain moderate, suggesting that simply increasing the number of target-domain samples (ranging from 100 to 800 in our experiments) is insufficient for achieving substantial improvements. This finding highlights another key challenge in text attribution task: effective adaptation to new domains remains elusive even with additional labeled data.

**Takeaways.** We evaluate the transferability of detectors in text attribution tasks. The results indicate that transferring across different domains in attribution tasks may suffer from severe performance degradation. Adding data from the target domain only shows moderate performance, indicating effective adaptation to new domains remains elusive. Future works are encouraged to develop robust detectors and efficient techniques for domain adaptation.

## 6 Adaptation to New LLMs

Since new LLMs with different characteristics are continuously released, we study how the pre-trained detector (trained on the text attribution task already) would adapt to the new class introduced over time with very limited (or without) access to the original training data for earlier classes. This scenario is common in real-world applications, where models often need to generalize to unseen distributions without retraining on past data. For simplicity, we mainly focus on the scenario where only one new LLM is introduced to the original detector (trained with HWTs and four types of MGTs). Moreover, we also consider the case where the detectors must adapt to two classes. To the best of our knowledge, we are the first to investigate the adaptation ability of MGT detectors.

Since the metric-based detectors generally show very poor performance in text attribution task, our study mainly focuses on the supervised model-based detectors, i.e., DeBERTa-V3, RoBERTa, and DistilBERT, for this more challenging setting.

### 6.1 Few-shot Adaptation

**Experiment Settings.** During the pre-training stage, the objective is to train a five-class classifier, and training arguments are shown in Appendix D. During the adaptation stage, the training data comprises very few MGTs (e.g., 1, 5, or 20) from one or two new LLMs, combined with an equal number of samples from the previously seen classes. The new learning objective is to extend the original five-class classifier to accommodate an additional class, resulting in a six-class classifier. During the evaluation stage, the testing data includes a balanced set of samples from all six classes. Due to page limit, results for RoBERTa and DistilBERT are in our arXiv version

**Techniques.** Few-shot adaptation methods do not require fine-tuning, and we use the pre-trained detector as the feature extractor and evaluate three representative methods.

- ProtNet [35] computes class prototypes $\mathbf{p}_c$ by averaging embeddings in each class (support set):
$$\mathbf{p}_c = \frac{1}{|\mathcal{S}_c|} \sum_{\mathbf{x}_i \in \mathcal{S}_c} f(\mathbf{x}_i),$$
where $\mathcal{S}_c$ is the support set for class $c$, and $f(\mathbf{x}_i)$ is the feature representation of input $\mathbf{x}_i$. Classification is based on Euclidean distance to each prototype $\hat{y} = \arg\min_c \|\mathbf{f}(\mathbf{x}) - \mathbf{p}_c\|_2^2$.
- RelationNet [39] learns a relation score between query samples and class prototypes using a neural network:
$$\text{RelationScore}(\mathbf{x}, c) = g\left(\mathbf{f}(\mathbf{x}) \oplus \mathbf{p}_c\right),$$

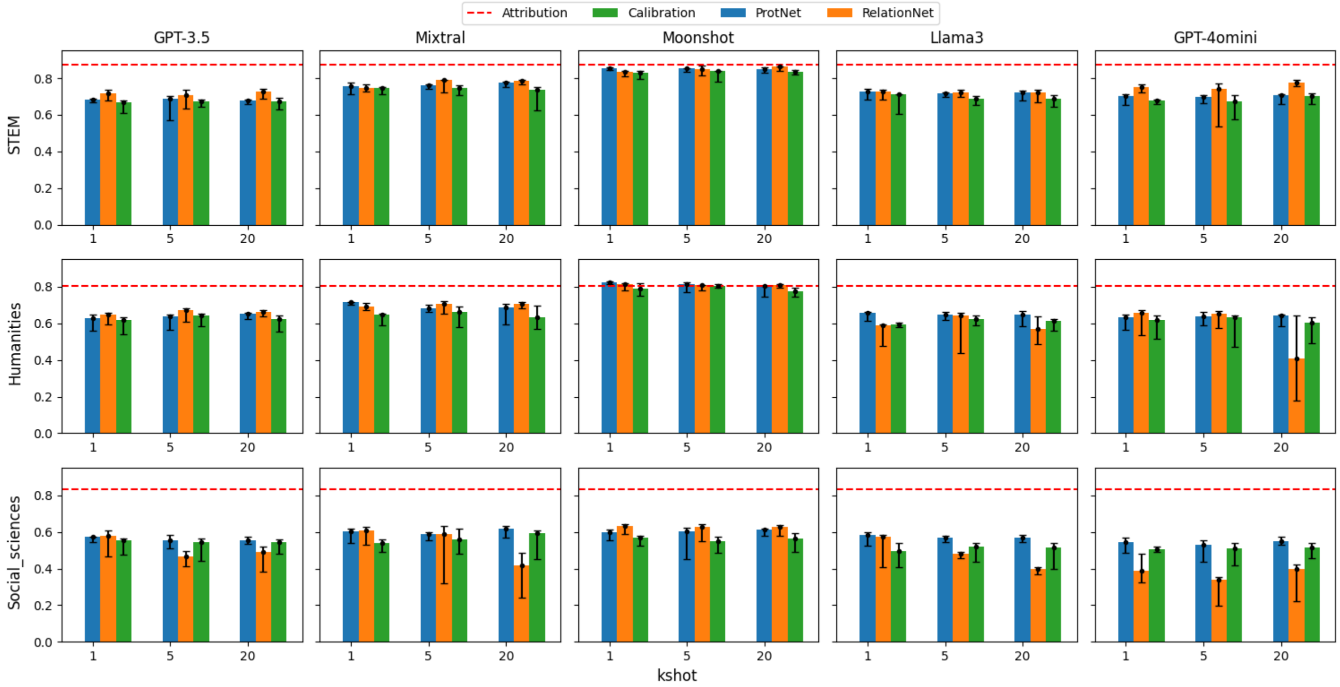

**Figure 4: Experiment results for adapting to new LLM in the few-shot setting (1 new class). Kshot represents the number of examples in the new class. The title of each column represents the newly introduced class. Dashed lines represent the ideal performance achieved when the detector is trained on all data at once. The detector is built on the DeBERTa-V3.**

where ⊕ denotes concatenation and $g$ is a learnable relation module. The class with the highest score is selected as the prediction.

- Distribution Calibration [47] first converts the input feature $x$ into $x'$ using Tukey's Ladder of Powers and subsequently leverages statistics from base classes to calibrate the feature distribution of new classes:

$$\mathbf{f}_{\text{calibrated}}(\mathbf{x}') = \frac{\mathbf{f}(\mathbf{x}') - \mu_{\text{base}}}{\sigma_{\text{base}}},$$

where $\mu_{\text{base}}$ and $\sigma_{\text{base}}$ are the mean and standard deviation computed from base-class features. Classification is similar to ProtNet, which finds the closest class mean of the transformed features as the prediction.

**Few-shot Results.** The results of DeBerta-v3 (the number of new classes is one) are shown in Figure 4, from which we have several observations. First, we find that increasing the support set in each class has limited improvement in performance. This means one representative example for the new class can effectively adapt the detector to new classes. Second, the performance of Social Science is generally poor, which may reflect the inherent similarity of text generated by different LLMs. Third, we analyze the performance of different techniques. ProtNet generally gives the most stable performance with the smallest variance, while other methods suffer from the high dimensionality (768) of extracted features: Regarding RelationNet, it utilizes a multi-layer neural network for classification, as a result, high-dimensional inputs can lead to instability and high variance. Regarding distribution calibration, it first transforms the features into a normal distribution and samples data points from the distribution to augment the classifier. However, the high dimension of extracted features may lead to improper augmentation and relatively poor performance. Furthermore, we study the scenario where two new LLMs are introduced. Specifically, since exhausting all possible cases is resource-consuming and brings limited insights, we only consider a practical case where the two LLMs are the latest released models, i.e., Llama-3 and GPT-4omini. The results of DeBerta-v3-based are shown in Figure 5. The degraded performance (drop from 0.67 to less than 0.6) suggests that extending the number of new classes poses a notably new challenge to the original hard task.

## 6.2 Many-shot Adaptation

**Experiment Setting.** During the pre-training stage, the objective is the same as that in few-shot adaptation to train a five-class classifier. During the adaptation stage, the training data comprises the full data in the new class, with some techniques needing limited access (typically 100 examples) to previously seen classes. During the evaluation stage, the testing data includes a balanced set of samples from all six classes.

To accommodate the new class, we extend the classification head by replacing the final linear layer with one that has an additional output dimension. The weights corresponding to the existing classes are kept unchanged, while the new weight vector for the additional class is initialized appropriately. This strategy, commonly used in previous work [17], allows the detector to adapt to new

**Table 6: Experiment result for adapting to new LLM in many-shot setting (1 new class). The new class is referred to as the Last Model. The results are reported using the F1-score. Attribution represents the ideal performance achieved when the detector is trained on all data at once. The larger values with blue colors indicate better performance, and the lower values with red colors indicate weaker performance.**

| Domain | Last Model | DistilBert | | | | | | RoBERTa | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Normal | LwF | iCaRL | BiC | Combine | Attribution | Normal | LwF | iCaRL | BiC | Combine | Attribution |
| Social Science | GPT-3.5 | 0.5459 | 0.545 | 0.6013 | 0.5923 | 0.5923 | 0.8174 | 0.5380 | 0.5114 | 0.5985 | 0.5960 | 0.5966 | 0.8177 |
| | Mixtral | 0.6099 | 0.6099 | 0.6253 | 0.6341 | 0.6341 | | 0.6168 | 0.6164 | 0.6604 | 0.6588 | 0.6566 | |
| | Moonshot | 0.5401 | 0.5401 | 0.6214 | 0.6215 | 0.6215 | | 0.5844 | 0.5849 | 0.6580 | 0.6383 | 0.6383 | |
| | Llama3 | 0.6595 | 0.6591 | 0.6500 | 0.6585 | 0.6581 | | 0.6108 | 0.6105 | 0.6638 | 0.6686 | 0.6695 | |
| | GPT-4omini | 0.5793 | 0.5798 | 0.5881 | 0.5906 | 0.5906 | | 0.5951 | 0.5946 | 0.6014 | 0.6121 | 0.6117 | |
| | Average | 0.5869 | 0.5868 | 0.6172 | 0.6194 | 0.6193 | | 0.5890 | 0.5835 | 0.6364 | 0.6348 | 0.6345 | |
| STEM | GPT-3.5 | 0.6077 | 0.5999 | 0.6319 | 0.6355 | 0.6348 | 0.8444 | 0.6151 | 0.6595 | 0.6555 | 0.6579 | 0.6585 | 0.8881 |
| | Mixtral | 0.6258 | 0.6262 | 0.6742 | 0.6651 | 0.6668 | | 0.6290 | 0.6286 | 0.6862 | 0.6896 | 0.6894 | |
| | Moonshot | 0.6459 | 0.6455 | 0.7569 | 0.7255 | 0.7309 | | 0.7526 | 0.753 | 0.7975 | 0.7898 | 0.7905 | |
| | Llama3 | 0.6055 | 0.6055 | 0.6710 | 0.6666 | 0.6667 | | 0.5225 | 0.5246 | 0.6920 | 0.6934 | 0.6935 | |
| | GPT-4omini | 0.6219 | 0.6221 | 0.6447 | 0.6498 | 0.6503 | | 0.6124 | 0.6099 | 0.6529 | 0.6692 | 0.6699 | |
| | Average | 0.6214 | 0.6198 | 0.6757 | 0.6685 | 0.6699 | | 0.6263 | 0.6351 | 0.6968 | 0.7000 | 0.7003 | |
| Humanity | GPT-3.5 | 0.5742 | 0.5742 | 0.5707 | 0.5805 | 0.5809 | 0.7835 | 0.5355 | 0.5351 | 0.5913 | 0.5843 | 0.5860 | 0.8151 |
| | Mixtral | 0.5738 | 0.5744 | 0.6558 | 0.6583 | 0.6583 | | 0.5367 | 0.5369 | 0.6733 | 0.6542 | 0.6561 | |
| | Moonshot | 0.5946 | 0.5945 | 0.7328 | 0.7303 | 0.7291 | | 0.6634 | 0.6609 | 0.7431 | 0.7586 | 0.7586 | |
| | Llama3 | 0.5343 | 0.5346 | 0.6437 | 0.6347 | 0.6352 | | 0.5852 | 0.5842 | 0.6602 | 0.6695 | 0.6683 | |
| | GPT-4omini | 0.5713 | 0.5714 | 0.6004 | 0.6042 | 0.6042 | | 0.5839 | 0.5837 | 0.6132 | 0.5935 | 0.5932 | |
| | Average | 0.5697 | 0.5698 | 0.6407 | 0.6416 | 0.6415 | | 0.5809 | 0.5802 | 0.6562 | 0.6520 | 0.6524 | |
| Overall Average | | 0.5927 | 0.5921 | 0.6445 | 0.6432 | 0.6436 | 0.8151 | 0.5988 | 0.5996 | 0.6632 | 0.6623 | 0.6624 | 0.8403 |

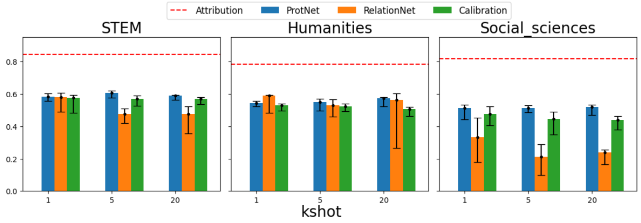

**Figure 5: Experiment results for adapting to new LLM in the few-shot setting (2 new classes: Llama-3 and GPT-4omini). Dashed lines represent the ideal performance achieved when the detector is trained on all data at once. The detector is built on DeBerta-v3-base.**

classes incrementally without retraining the entire model. Due to page limit, results for DeBERTa-V3 are in our arXiv version.

**Techniques.** Since many-shot adaptation methods require fine-tuning the detector, we lower the initial learning rate to 1/4 of that in the pre-training stage to maintain performance.

- LwF [17] distills the logits from the previous model into the new one. The key idea is to preserve the logits of the previous model on old tasks during new tasks by adding a regularization term to the loss function:

$$\mathcal{L} = \mathcal{L}_{\text{new}}(x, y) + \lambda \mathcal{L}_{\text{distill}}(x), \tag{1}$$

where $\mathcal{L}_{\text{new}}$ is the classification loss for new tasks, $\mathcal{L}_{\text{distill}}$ is the distillation loss to keep the results for old tasks, and $\lambda$ balances the two terms.

- iCaRL [31] introduces memory replaying, which maintains a fixed memory buffer to store a subset of examples in the pre-training stage. The memorized samples are subsequently replayed during the adaptation stage to retain knowledge from the previous stage.
- BiC [45] focuses on addressing the bias towards new classes. It first trains the model ($n$ old classes and $m$ new classes) without any correction, followed by adding a bias correction layer that adjusts the logits of new classes.

$$q_k = \begin{cases} o_k, & \text{if } 1 \leq k \leq n, \\ \alpha o_k + \beta, & \text{if } n+1 \leq k \leq n+m, \end{cases} \tag{2}$$

where $q_k$ is the new logit for class $k$, $o_k$ is the original logit, and $\alpha$, $\beta$ are one-dimensional learnable parameters. The bias correction layer ensures that the model's predictions remain balanced across classes to avoid the favor for new classes due to their dominance in the learning process.

- Combine integrates three techniques, i.e., the knowledge distillation loss, memory buffer, and bias correction techniques, to produce a combined method.

**Many-shot Results.** We evaluate the performance of two supervised models in both normal fine-tuning and fine-tuning with four different techniques during the adaptation stage. The results are shown in Table 6, from which we have several observations. First, the many-shot setting still suffers from an obvious performance drop compared to the standard six-class attribution task, primarily

**Table 7: Experiment result for adapting to new LLM in many-shot setting (2 new classes). Normal represents direct fine-tuning without any technique. Attribution represents the ideal performance achieved when the detector is trained on all data at once. The larger values with blue colors indicate better performance, and the lower values with red colors indicate weaker performance.**

| Domain | **Last Model** | DistilBert | | | | | RoBERTa | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Normal | LwF | iCaRL | BiC | Attribution | Normal | LwF | iCaRL | BiC | Attribution |
| Social Science | GPT-4omini | 0.4529 | 0.4791 | 0.4788 | 0.4792 | 0.8174 | 0.4407 | 0.4187 | 0.4891 | 0.4891 | 0.8177 |
| STEM | + | 0.4540 | 0.4712 | 0.4698 | 0.4703 | 0.8444 | 0.4857 | 0.5098 | 0.5000 | 0.5005 | 0.8881 |
| Humanity | Llama3 | 0.4313 | 0.4327 | 0.4657 | 0.4657 | 0.7835 | 0.4538 | 0.4521 | 0.4821 | 0.4833 | 0.8151 |

due to catastrophic forgetting [17]. Second, employing different techniques can effectively mitigate performance degradation. For instance, the performance of RoBERTa improves by 1%, 10.75%, 10.59%, and 10.63%, respectively, using the four shown techniques. ICaRL and BiC techniques demonstrate competitive results, with each excelling in specific domains or LLMs. Despite these improvements, the results remain below the upper-bound performance observed in standard attribution tasks, underscoring the challenges of adapting to new LLMs.

Furthermore, we extend the evaluation to the scenario where two new LLMs are introduced, and we only consider a practical case where the two LLMs are the latest released models, i.e., Llama- 3 and GPT-4omini. The poor performance presented in Table 7 indicates that when two classes are introduced, many-shot fine-tuning can be ineffective. These findings highlight the inherent challenges in many-shot settings and emphasize the need for the exploration of scalable techniques.

**Takeaways.** We introduce a new attribution task setting that adapts a detector to new LLMs with very limited access to the previous data. Two settings, i.e., few-shot and many-shot settings, are established to solve the task. Both few-shot and many-shot techniques show competitive performance in adapting to new LLMs. However, there is still a performance gap between the current methods and the ideal values, which train the detectors on all data at once. Future works are encouraged to explore and develop more techniques to build continuously evolving MGT detectors.

## 7 Discussion and Limitation

**Datasets and Methods.** Despite the collected data being from multiple resources, some specific subject, such as Education or Chemistry, only covers Wiki data. Additionally, the MGTs are generated using one prompt and lack complicated processes, e.g., machine-written machine-humanized [1], or human-edited [2]. To enhance the dataset, future work could include texts generated by more advanced models such as DeepSeek-R1 or QwQ, under varied prompting strategies. As this study primarily focuses on evaluating detector performance in text attribution tasks, we leave the expansion of the dataset to future updates of our open-source codebase.

**Metric-based Detectors for Text Attribution.** Our results show that the metric-based detectors for text attribution only show performance slightly above random guesses, but do not provide a solution to poor performance. Although we find that simply concatenating different metrics may improve performance, there is still a huge gap between the model-based detectors. This is an important yet underexplored topic in developing robust and comprehensive MGT detectors. We leave it as our future work for further exploration.

**Adapting Detectors to New LLMs.** We are the first to introduce this setting in the MGT detection task and evaluate the result on detectors equipped with several adapting techniques. Due to the limitation of research resources, we only collect data from 5 different LLMs, as a result, the number of classes in the pre-training stage and adapting stage is small. We will continuously expand the data resources and add more advanced LLMs.

**Robustness of Evading Attacks.** Recent studies [44, 49] have focused on evaluating the robustness of detectors against adversarial attacks and revealed the vulnerability of current MGT detectors. While robustness is a critical aspect of reliable text attribution, evaluating it is beyond the scope of this work. We therefore leave a thorough investigation of detector robustness under adversarial conditions to future research.

## 8 Conclusion

In this work, we introduce MGT-Academic, a large-scale academic writing dataset containing over 336M tokens and 749K samples from STEM, Humanities, and Social Sciences, including HWTs and MGTs, generated by five different LLMs. First, our findings show that metric-based detectors struggle in attribution tasks due to the low dimensionality of their features. This underscores the need for further development of metric-based approaches suitable for attribution tasks. Second, transferring across domains in attribution tasks is still a hard problem, and simple techniques (adding data) cannot effectively improve the performance. Third, we introduce a novel attribution task setting where detectors must adapt to new LLMs with minimal access to prior data. While techniques for both few-shot and many-shot settings show competitive performance, a gap remains between current methods and the ideal case of training detectors on all available data.

## Acknowledgment

This work is supported by Shanghai Science and Technology Innovation Action Plan Project (No. 24692102500, No. 22510713600) and the Key Laboratory of Computing Power Network and Information Security, Ministry of Education (No.2023ZD029). We thank the valuable feedback from reviewers.

## References

[1] Mervat Abassy, Kareem Elozeiri, Alexander Aziz, Minh Ngoc Ta, Raj Vardhan Tomar, Bimarsha Adhikari, Saad El Dine Ahmed, Yuxia Wang, Osama Mohammed Afzal, Zhuohan Xie, Jonibek Mansurov, Ekaterina Artemova, Vladislav Mikhailov, Rui Xing, Jiahui Geng, Hasan Iqbal, Zain Muhammad Mujahid, Tarek Mahmoud, Akim Tsvigun, Alham Fikri Aji, Artem Shelmanov, Nizar Habash, Iryna Gurevych, and Preslav Nakov. Llm-detectaive: a tool for fine-grained machine-generated text detection, 2024.

[2] Ekaterina Artemova, Jason Lucas, Saranya Venkatraman, Jooyoung Lee, Sergei Tilga, Adaku Uchendu, and Vladislav Mikhailov. Beemo: Benchmark of expert-edited machine-generated outputs, 2025.

[3] Sameer Badaskar, Sachin Agarwal, and Shilpa Arora. Identifying Real or Fake Articles: Towards better Language Modeling. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 817–822. ACL, 2008.

[4] 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, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.

[5] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[6] Rudolf Flesch. Marks of readable style; a study in adult education. *Teachers College Contributions to Education*, 1943.

[7] Yuyou Gan, Yong Yang, Zhe Ma, Ping He, Rui Zeng, Yiming Wang, Qingming Li, Chunyi Zhou, Songze Li, Ting Wang, Yunjun Gao, Yingcai Wu, and Shouling Ji. Navigating the risks: A survey of security, privacy, and ethics threats in llm-based agents, 2024.

[8] Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. GLTR: Statistical Detection and Visualization of Generated Text. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 111–116. ACL, 2019.

[9] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[10] Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *CoRR abs/2301.07597*, 2023.

[11] 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 *Proceedings of the 41st International Conference on Machine Learning*. PMLR, 2024.

[12] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

[13] Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. MGT-Bench: Benchmarking Machine-Generated Text Detection. *CoRR abs/2303.14822*, 2023.

[14] Xinlei He, Guowen Xu, Xingshuo Han, Qian Wang, Lingchen Zhao, Chao Shen, Chenhao Lin, Zhengyu Zhao, Qian Li, Le Yang, Shouling Ji, Shaofeng Li, Haojin Zhu, Zhibo Wang, Rui Zheng, Tianqing Zhu, Qi Li, Chaoxiang He, Qifan Wang, Hongsheng Hu, Shuo Wang, Shi-Feng Sun, Hongwei Yao, Zhan Qin, Kai Chen, Yue Zhao, Hongwei Li, Xinyi Huang, and Dengguo Feng. Artificial intelligence security and privacy: a survey. *Science China Information Sciences*, 2025.

[15] 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.

[16] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic Detection of Generated Text is Easiest when Humans are Fooled. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1808–1822. ACL, 2020.

[17] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

[18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

[19] Yule Liu, Zhen Sun, Xinlei He, and Xinyi Huang. Quantized delta weight is safety keeper. *arXiv preprint arXiv:2411.19530*, 2024.

[20] Yule Liu, Jingyi Zheng, Zhen Sun, Zifan Peng, Wenhan Dong, Zeyang Sha, Shiwen Cui, Weiqiang Wang, and Xinlei He. Thought manipulation: External thought can be efficient for large reasoning models. *arXiv preprint arXiv:2504.13626*, 2025.

[21] Zeren Luo, Zifan Peng, Yule Liu, Zhen Sun, Mingchen Li, Jingyi Zheng, and Xinlei He. The rising threat to emerging ai-powered search engines. *arXiv preprint arXiv:2502.04951*, 2025.

[22] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost Van De Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):5513–5533, 2022.

[23] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2024.

[24] MistralAI. https://mistral.ai/.

[25] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature. *CoRR abs/2301.11305*, 2023.

[26] Moonshot. https://www.moonshot.cn.

[27] OpenAI. https://chat.openai.com/chat.

[28] OpenAI. GPT-4 Technical Report. *CoRR abs/2303.08774*, 2023.

[29] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*. NeurIPS, 2022.

[30] Debjit Paul, Daniil Sorokin, and Judith Gaspers. Class incremental learning for intent classification with limited or no old data. In Francesco Barbieri, Jose Camacho-Collados, Bhuwan Dhingra, Luis Espinosa-Anke, Elena Gribovskaya, Angeliki Lazaridou, Daniel Loureiro, and Leonardo Neves, editors, *Proceedings of the First Workshop on Ever Evolving NLP (EvoNLP)*, pages 16–25, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[31] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

[32] N Reimers. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[33] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018.

[34] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108, 2019.

[35] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017.

[36] Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *CoRR abs/2306.05540*, 2023.

[37] Zhen Sun, Tianshuo Cong, Yule Liu, Chenhao Lin, Xinlei He, Rongmao Chen, Xingshuo Han, and Xinyi Huang. Peftguard: Detecting backdoor attacks against parameter-efficient fine-tuning. *arXiv preprint arXiv:2411.17453*, 2024.

[38] Zhen Sun, Zongmin Zhang, Xinyue Shen, Ziyi Zhang, Yule Liu, Michael Backes, Yang Zhang, and Xinlei He. Are We in the AI-Generated Text World Already? Quantifying and Monitoring AIGT on Social Media. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL, 2025.

[39] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[40] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[41] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3964–3992. Association for Computational Linguistics, 2024.

[42] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, et al. M4GT-Bench: Evaluation Benchmark for Black-Box Machine-Generated Text Detection. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

[43] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR abs/1910.03771*, 2019.

[44] 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. *CoRR*, abs/2410.23746, 2024.

[45] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 374–382, 2019.

[46] Congying Xia, Wenpeng Yin, Yihao Feng, and Philip S. Yu. Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system.

In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1351–1360. Association for Computational Linguistics, 2021.

[47] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. *arXiv preprint arXiv:2101.06395*, 2021.

[48] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2024.

[49] Jingyi Zheng, Junfeng Wang, Zhen Sun, Wenhan Dong, Yule Liu, and Xinlei He. TH-Bench: Evaluating Evading Attacks via Humanizing AI Text on Machine-Generated Text Detectors. In *Proceedings of the 31st ACM International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2025.

[50] Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *arXiv preprint arXiv:2303.07338*, 2023.

# A Generation Prompts

### Prompt Template for Wiki Text

<Background>:
Please act as an expert wiki editor and revise the wiki content from the perspective of a wiki editor to make it fluent and elegant. Here are the specific requirements:
1. You should provide accurate and comprehensive information. Use reliable sources and cross-check your information to ensure its accuracy.
2. Wiki articles should be neutral and unbiased. Avoid expressing personal opinions or promoting a particular viewpoint. Instead, present all relevant information and let the readers form their own opinions.
3. Your writing should be clear and easy to understand. Avoid using complex sentences and unnecessary words. Remember, your goal is to convey information, not to showcase your vocabulary. Please only include the written wiki page in your answer.
Here is the original wiki page:
<text>: //to-be-polished text

### Prompt Template for Arxiv Text

<Background>:
Please act as an expert paper editor and revise a section of the paper to make it more fluent and elegant. Please only include the revised section in your answer. Here are the specific requirements:
1. Enable readers to grasp the main points or essence of the paper quickly.
2. Allow readers to understand the important information, analysis, and arguments throughout the entire paper.
3. Help readers remember the key points of the paper.
4. Please clearly state the innovative aspects of your research in the section, emphasizing your contributions.
5. Use concise and clear language to describe your findings and results, making it easier for reviewers to understand the paper. Here is the original section of the paper:
<text>: //to-be-polished text

### Prompt Template for Gutendex Text

<Background>:
Please act as an expert book editor and revise the book content from the perspective of a book editor to make it fluent and elegant.
1. Clarity: Ensure that your writing is clear and easy to understand. Avoid jargon and complex language that may confuse the reader.
2. Relevance: Make sure that the content you are writing is relevant to the topic at hand. Do not deviate from the main subject.
3. Accuracy: Ensure that all the information you provide is accurate and up-to-date. This includes statistics, facts, and theories.
4. Brevity: Keep your writing concise. Avoid unnecessary words or phrases that do not add value to the content. Here is the original book content:
<text>: //to-be-polished text

# B Editing Distance

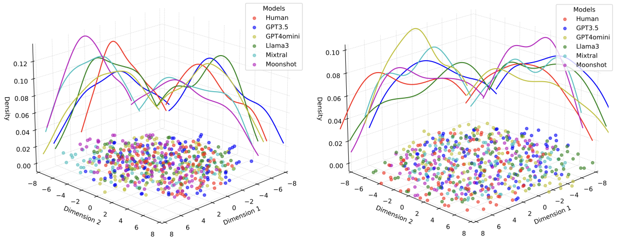

**Figure A1: Humanity (left) and STEM (right) Embeddings Across Models.**

# C Detectors

For zero-shot detectors, metrics were obtained from the white-box model Llama2-7B-Instruct [9], unless stated otherwise. Fast-DetectGPT and Binoculars were evaluated using the optimal settings specified in their respective papers. For model-based detectors, DistilBERT and RoBERTa were fine-tuned with a learning rate of 5e-6, batch size of 64, 3 epochs, and a random seed of 3407. RADAR and ChatGPT-D used their officially released weights without additional fine-tuning. Details of detectors in binary classification and text attribution are provided in Appendix D.

**Log-Likehood [8].** A zero-shot method uses a language model to compute the log probability of each token in a text. A higher average log-likelihood suggests the text is more likely generated by an LLM.

**Rank [8].** A zero-shot method calculates the absolute rank of each token in a text based on its previous context and determines the text's score by averaging these rank values. A smaller average rank score indicates a higher likelihood that the text is machine-generated.

**Rank GLTR [8].** GLTR is designed to assist in labeling machine-generated text. We uses Test-2 features as suggested by Guo et al. [10], evaluating the fraction of words ranked within 10, 100, 1,000, and beyond.

**LRR [36].** The Log-Likelihood Log-Rank Ratio (LRR) combines Log-Likelihood and Log-Rank, with a higher LRR indicating a greater likelihood of text being machine-generated.

**Entropy [8].** A zero-shot method uses entropy to measure text randomness, with lower entropy indicating a higher likelihood of being LLM-generated, as human-written text shows greater unpredictability.

**Fast-DetectGPT [4].** An optimized zero-shot detector improves DetectGPT [25] by replacing perturbation with efficient sampling. We followed

**Table A1: Sources of Data for Machine Generated Text: This table lists the primary paper sources (such as Arxiv and Project Gutenberg) and supplementary resources (Wiki) available for different STEM, Social Science, and Humanity subfields, along with notes on the availability of paper sources where applicable.**

| Domain | Subfield | Source | Human | GPT-4omini | GPT-3.5 | Mixtral-8×7b | Llama-3.1-70b | Moonshot-8k |
|---|---|---|---|---|---|---|---|---|
| STEM | Physics | Arxiv & Wiki | 10.8K / 11,926.6K | 9.1K / 5,485.9K | 8.4K / 2,106.2K | 8.7K / 3,291.4K | 3.5K / 1,059.1K | 2.5K / 891.9K |
| | Math | | 14.1K / 12,338.5K | 12.0K / 5,717.0K | 13.6K / 3,444.1K | 10.3K / 3,425.9K | 6.4K / 1,836.8K | 2.4K / 808.8K |
| | CS | | 14.7K / 12,275.5K | 11.9K / 4,989.4K | 14.2K / 3,415.4K | 9.4K / 3,277.9K | 3.5K / 934.2K | 2.9K / 983.6K |
| | Biology | | 15.7K / 11,485.3K | 13.6K / 5,704.5K | 14.9K / 3,508.6K | 10.9K / 3,716.7K | 3.5K / 924.9K | 3.5K / 1,159.3K |
| | EE | | 19.7K / 13,346.2K | 16.9K / 6,129.3K | 18.6K / 4,378.4K | 13.0K / 4,384.9K | 4.2K / 1,130.5K | 4.1K / 1,350.1K |
| | Statistics | | 9.7K / 11,491.9K | 7.6K / 4,339.8K | 9.5K / 2,545.2K | 6.8K / 2,610.7K | 2.9K / 832.4K | 2.0K / 694.1K |
| | Chemistry | Wiki | 2.4K / 415.4K | 2.2K / 425.2K | 2.8K / 536.8K | 1.6K / 438.6K | 1.0K / 148.6K | 0.5K / 141.2K |
| | Medicine | | 8.7K / 1,668.3K | 8.1K / 1,662.6K | 7.8K / 1,470.6K | 5.1K / 1,419.3K | 2.0K / 454.8K | 1.8K / 528.9K |
| Social Science | Education | Gutenberg & Wiki | 14.2K / 3,831.0K | 13.0K / 2,833.9K | 12.5K / 2,377.2K | 9.2K / 2,704.2K | 3.1K / 925.6K | 2.8K / 905.9K |
| | Economy | Arxiv & Wiki | 12.6K / 6,807.5K | 11.3K / 3,663.5K | 11.7K / 2,531.4K | 6.1K / 2,025.1K | 2.4K / 655.6K | 1.9K / 621.5K |
| | Management | Wiki | 3.5K / 648.5K | 3.3K / 618.0K | 3.3K / 739.4K | 2.1K / 589.0K | 0.6K / 132.5K | 0.7K / 190.9K |
| Humanities | Literature | Gutenberg & Wiki | 18.7K / 13,276.7K | 13.6K / 5,306.9K | 11.4K / 2,077.9K | 13.5K / 5,306.9K | 9.3K / 4,439.6K | 3.9K / 1,823.6K |
| | Law | | 7.5K / 2,695.6K | 6.5K / 1,639.1K | 6.1K / 1,106.6K | 5.2K / 1,727.3K | 2.3K / 780.2K | 1.5K / 522.0K |
| | Art | | 8.4K / 5,899.5K | 6.4K / 2,576.2K | 5.7K / 1,110.1K | 6.0K / 2,411.3K | 4.2K / 2,040.1K | 1.8K / 816.4K |
| | History | | 30.0K / 33,517.6K | 18.4K / 12,848.3K | 14.5K / 3,505.5K | 23.6K / 11,572.2K | 18.4K / 11,019.4K | 6.1K / 3,644.4K |
| | Philosophy | | 3.5K / 1,998.4K | 2.7K / 776.5K | 2.3K / 407.8K | 2.6K / 937.0K | 1.5K / 598.6K | 0.7K / 280.6K |

**Table A2: Detailed Data Moderation Policy**

| | | |
|---|---|---|
| **Machine Split** | Generation Identifier | ‘revised book’,‘revised content’,‘revised version’,‘title’,<br>‘after editing’,‘revised section’ |
| | Special Keywords | ‘book editor’, ‘clarity’, ‘revisions’, ’I apologize’,‘I am sorry’, ‘Unfortunately’, ‘complex language’, ‘revised content’, ‘revised version’, ‘language model’, ‘revised content’, ‘revised version’, ‘accuracy of’, ‘project gutenberg’, ‘reliable information’, ‘ISBN’, ‘PMID’, ‘doi:’, ‘Sure,’, ‘Retrieved from’,‘Category’, ‘http’, ‘As an editor’, ‘As an expert’ |
| | Format Symbols | ‘&’, ‘$’, ‘====’, ‘—’,‘**’, ‘##’, |
| **Human Split** | Special Keywords | ‘ISBN’, ‘PMID’, ‘doi’, ‘vol.’, ‘p.’, ‘https:’, ‘http:’, ‘References External links’ |
| | Format Symbols | ‘\n—’, ‘\n===’, ‘**’, ‘##’, ‘$’ (> 500), ‘&’ (> 150), ‘\’ (> 1000) |

the authors' optimal settings, using GPT-Neo-2.7b as the scoring model and GPT-J-6b as the reference model.

**Binoculars [11].** A zero-shot detection method uses two LLMs to compute the perplexity-to-cross-perplexity ratio. Following the authors' optimal settings, we used Falcon-7B-Instruct for PPL and Falcon-7B for X-PPL.

**RADAR [15].** RADAR uses adversarial training between a paraphraser and a detector. We used the pre-trained RoBERTa detector from Hugging Face without additional training.

**ChatGPT-D [10].** ChatGPT Detector distinguishes human-written from ChatGPT-generated texts by fine-tuning a RoBERTa model on the HC3 [10] dataset.

**DistilBERT [34].** The detector is built by fine-tuning a pre-trained DistilBERT model with an additional classification layer.

**RoBERTa [18].** The detector is built by fine-tuning a pre-trained RoBERTa model with an additional classification layer.

## D Experimental Settings

**In-distribution Task.** For zero-shot detectors, we randomly selected 1,000 training samples to predict the metrics. The classification threshold was set to maximize the F1-score on the training set, and a classifier was trained using the same data. Note that GLTR produces vectors of four rank values, and thus, threshold-based classification is not applicable. For model-based detectors, we fine-tuned the model using at most 10,000 training samples. We used 2000 randomly selected data points for the testing set. Zero-shot detectors employ threshold-based classification and logistic regression for binary human-machine classification tasks.

**Text Attribution Task.** Text attribution tasks use SVM and logistic regression classifiers with default sklearn implementations and a linear kernel for SVM. Model-based detectors have their classification heads adjusted to match the number of classes. Fine-tuning was done with a learning rate of 5e-6, batch size of 64, 3 epochs, and a random seed of 3407.

**Class Incremental Experiment.** To train the original model, we use a setting similar to that in the model attribution task. We train the model for 2 epochs in this stage and set the learning rate to 5e-6 and the batch size to 64. To train the updated model, training data has the same number of data as each class had in the previous stage. We train the model for 1 epoch in this stage and set the learning rate to 2.5e-7 (1/4 of the original lr) and the batch size to 64. For the LwF technique, the regularization parameter is set to 0.2. To maintain the example in iCaRL, we set the cache size for each class to 100. The validation set in BiC is constructed by combining the data in the example together. Specifically, since a small amount of old data is introduced in the training process of iCaRL and BiC, we adopt weighted cross entropy to avoid the side effects of data imbalance.