How Robust is AI Texts to Human Texts in the Wild? A Realistic Benchmark for Multi-Dimensional Evaluation and Detection

Anonymous ACL submission

Abstract

Recent advancements in large language models (LLMs), especially with emerging technologies like reasoning, have increased challenges for detecting LLM-generated text. This highlights the need for a robust benchmark to evaluate detectors in real-world scenarios. Here we introduce BR^{2}ED, a new benchmark for LLM-generated text detection, focusing on three dimensions: data, generator, and detector categorization, ensuring and offering comprehensive evaluation. Through extensive evaluation of baseline detectors, we find that (1) texts generated by reasoning models are more difficult to detect, and (2) operation combinations tend to reduce detection performance. BR^{2}ED provides a standardized platform for assessing detector robustness and advancing LLM-generated text detection.

1 Introduction

Since the release of ChatGPT(OpenAI, 2022), Large language models (LLMs) have rapidly advanced, with techniques like deep reasoning(Wei et al., 2022) enabling broader and more effective use. However, these advancements have posed significant challenges in detecting LLM-generated text, raising two key questions:

(1) Do existing detection tools remain effective when confronted with text generated by newly emerging LLM and generation strategies?

(2) Can existing benchmarks accurately reflect the complexity of real-world scenarios across LLM/operations?

Most existing benchmarks (He et al., 2024; Saha and Feizi, 2025; Dugan et al., 2024) focus on static, single-step generation and overlook the dynamic nature of text generation in real-world applications. In practice, however, generated texts often result from complex, evolving procedures. This limitation leads to the systematic neglect of three critical dimensions in evaluation design, which may result in significant performance degradation in complex real-world scenarios:

(1) **Data diversity**. Existing benchmarks lack **a) more complex and realistic generation scenarios arising from *combinations* of diverse operations or LLMs**; b) comprehensive coverage across both different domains and generation operations.

(2) **LLM variety**. Current benchmarks mostly focus on differences across LLM series, neglecting the influence caused by factors like model mechanisms within the same model family, such as differences in detection capability due to **instruction alignment** and **reasoning ability**.

(3) **Detectors categorization**. Existing detection frameworks generally fail to provide a more detailed and systematic categorization of models according to their characteristics.

To address these issues, we propose **BR^{2}ED**, an evaluation framework for LLM-generated text detection, and perform a comprehensive benchmarking of various detectors. BR^{2}ED is designed to reflect dynamic, real-world text generation scenarios, and systematically covers 3 key dimensions: data diversity, LLM variety, and detector categorization. The detailed framework will be discussed in Section 3. Based on our experiments and discussions, we have the following key insights:

(1) Operation combinations generally tend to disrupt structural consistency and increase the stylistic variety of generated texts, making them harder to identify.

(2) In LLM combination, although the generated content merges traits from both models, detectability is primarily influenced by the secondary LLM's detection difficulty.

(3) Texts produced by reasoning models are harder to identify due to their chain-of-thought, making them appear more human-like.

2 Related Work

2.1 LLM-generated Text Detection

We focus on black-box detectors, which can be mainly classified into two categories: zero-shot and supervised methods.

Zero-shot methods do not require any training data, as pretrained models directly capture the syntactic and statistical properties of the text. The extracted feature vectors are then used to compute metrics for classifying. A representative work is DetectGPT(Mitchell et al., 2023), which relies on log-probability curvature. Building on the architecture of DetectGPT, Fast-DetectGPT(Bao et al., 2023) optimized the sampling process and the statistical metric, resulting in improved detection accuracy and efficiency.

Supervised methods can be further categorized into three types: (1) *Linguistic-based detectors* use pretrained LLMs to extract linguistic features for classifier training. A recent example is Ghostbuster(Verma et al., 2023), which combines structured search with linear classification. (2) *Direct LM-based detectors* leverage pretrained LLMs to extract text embeddings directly for classification. Early work fine-tuned RoBERTa(Liu et al., 2019; Guo et al., 2023), While LLMs like Llama can also be used as binary classifiers. (3) *Modified LM-based detectors* fine-tune pretrained LLMs with a small amount of training data, and then use the augmented LLs for feature extraction or direct classification. The recent ImBD(Chen et al., 2025) detector further refines the logits output of LLMs through Style Preference Optimization.

2.2 LLM-generated Text Detection Benchmark

Table 1 compares BR^{2}ED with widely used benchmarks. Among these, HC3 (Guo et al., 2023) is one of the earliest and most widely adopted benchmarks, containing human-written and GPT-3-generated texts in both Chinese and English, primarily collected from QA tasks and encyclopedic sources. In contrast, CUDRT (Tao et al., 2026) provides a more fine-grained categorization of operations, introducing five text manipulation types: Create, Update, Delete, Rewrite, and Translate. As shown in the table, most existing benchmarks exhibit limited coverage and are generally confined to relatively simple text generation scenarios.

3 BR^{2}ED

Figure 1 presents an overview of data composition and evaluation framework of BR^{2}ED. We identify three fundamental elements in detection-oriented text generation: data, LLMs, and detectors. BR^{2}ED is constructed by systematically combining these elements to form benchmark datasets and evaluation tasks. The following sections detail the design of BR^{2}ED along these three dimensions, and detailed statistics are shown in Section A.1.

3.1 Data Diversity

3.1.1 Comprehensiveness

Domains. To better reflect real-world scenarios, we categorize target data based on its formality. *Formal texts* include news articles from XSum(Narayan et al., 2018), Wikipedia entries from SQuAD(Rajpurkar et al., 2016, 2018), and academic abstracts from PubMedQA(Jin et al., 2019). Informal texts include creative writing from WritingPrompts(Fan et al., 2018), blog articles(Schler et al., 2006) and short Twitter posts(Go et al., 2009). Additionally, to account for the increasing use of LLMs in peer review, we include review comments from OpenReview[1], restricted to those published before 2019.

Operations. Based on CUDRT(Tao et al., 2026), we employ seven operations, which are grouped into three main categories. Detailed descriptions are shown in Section A.3.

(1) *Creative Reformation* includes 'create', 'rewrite', and 'summary', which correspond to text continuation, meaning-preserving rephrasing, and abstractive summarization at the document level.

(2) *Linguistic Enhancement* includes 'polish', 'expand', and 'refine', which correspond to stylistic improvement, moderate content expansion, and sentence-level simplification.

(3) *Cross-lingual Transfer* involves translating text between languages. Since our dataset is in English, each text is translated into an intermediate language and then back to English. Six intermediate languages are used, including four high-resource languages (Chinese, German, French, and Spanish) and two low-resource languages (Turkish and Greek).

3.1.2 Complexity

In real-world scenarios, LLM text generation often involves a secondary generation process with

[1] https://openreview.net/

Benchmark	Instruct	Rs	LLM-Co	Op-Co	Domains	Operations	CD	CM	CO
MGTBench(He et al., 2024)	×	×	×	×	3	1	✓	✓	×
DetectRL(Wu et al., 2024)	×	×	×	×	4	1	✓	✓	×
RAID(Dugan et al., 2024)	×	×	×	×	8	1	✓	✓	×
MixSet(Zhang et al., 2024)	×	×	×	×	6	5	×	✓	✓
M4(Wang et al., 2023)	×	×	×	×	5	1	✓	✓	×
APT-Eval(Saha and Feizi, 2025)	×	×	×	×	6	1	✓	✓	×
CUDRT(Tao et al., 2026)	×	×	×	×	4	7	×	✓	✓
BR^{2}ED	✓	✓	✓	✓	**7**	**7**	✓	✓	✓

Table 1: Comparison of data coverage and evaluation tasks across different benchmarks. 'RS' and 'Instruct' refer to reasoning and typical instruction-tuning tasks in LLM variations, 'LLM-Co' and 'Op-Co' denote LLM/operation combination tasks, and 'CD'/'CM'/'CO' represent Cross-Domain/Model/Operation experiments.

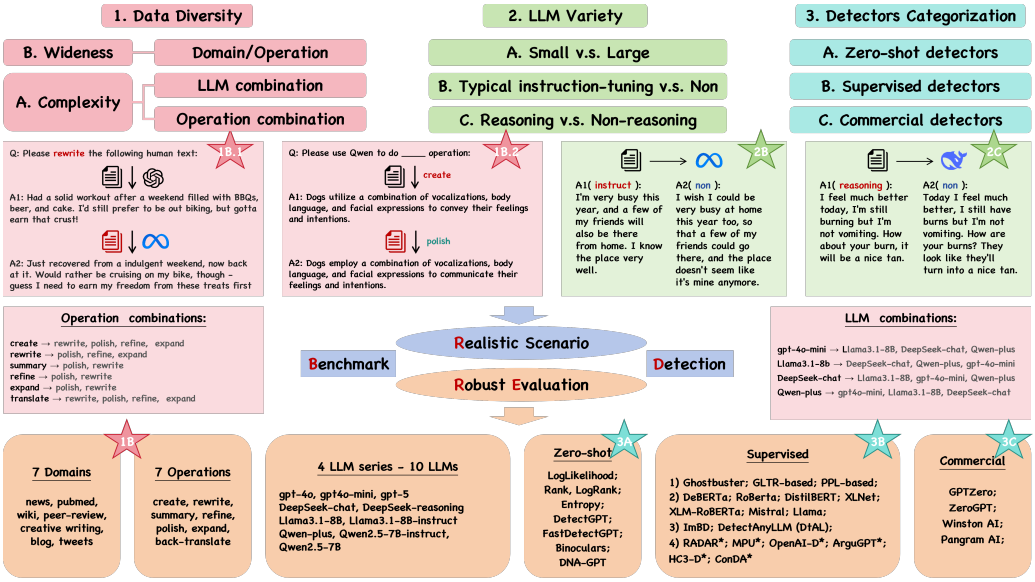


Figure 1: The framework of the BR^{2}ED benchmark.

various combinations of LLMs and operations. To capture this complexity, we constructed two sub-datasets: LLM-Co and Op-Co.

LLM-Co simulates scenarios in which text generated by one LLM is subsequently reprocessed by another. For this, we constructed 12 distinct model combinations as shown in Figure 1. In each case, the same operation was applied during both the initial generation and the reprocessing stage to ensure a controlled comparison.

Op-Co focuses on sequences of different operations applied to the same text. We defined 17 distinct operational combinations, using the same LLM for both generation steps to isolate the effect of the operations themselves.

3.2 LLM Variety

To systematically examine the differences in texts generated by different LLM variants within the same model family, and to assess the influence of model architecture and underlying mechanisms on text quality, we consider three representative dimensions: (1) **Model scale**, contrasting small($\leq$ 8B) and large($\geq$ 70B) models; (2) **Typical instruction-tuned**, contrasting models with/without typical instruction-tuned variants; and (3) **Reasoning capability**, contrasting non-reasoning models with reasoning models. We select a set of representative LLMs across these dimensions and use one-step generation to ensure comparison fairness, which are shown in Figure 1.

3.3 Detectors Categorization

3.3.1 Zero-shot detectors

We employ four logits-based methods, including log-likelihood(Ippolito et al., 2019), rank(Gehrmann et al., 2019), log-rank(Solaiman et al., 2019a), and entropy(Gehrmann et al., 2019). In addition, to comprehensively cover different evaluation perspectives, we select four state-of-the-art zero-shot detectors, DNA-GPT(Yang et al., 2023), Binoculars(Hans et al., 2024), DetectGPT, and Fast-DetectGPT, and assess their performance across diverse datasets.

3.3.2 Supervised detectors

We select several supervised detectors from three main categories.

Linguistic-based detectors include GLTR(Guo et al., 2023), perplexity(PPL)-based(Beresneva, 2016) classifiers, and Ghostbuster. *Direct embedding-based detectors* include DistilBERT, RoBERTa-large, XLM-RoBERTa-large, DeBERTa-v3-large, Llama3.1-8B-v2, Mistral-7B, and XL-Net. *Modified embedding-based detectors* include ImBD and DetectAnyLLM(DtAL)(Chen et al., 2025).

We also evaluate several **open-access** supervised detectors using their publicly released pre-trained weights only. Due to differences in training data, these classifiers are analyzed separately from the *previously mentioned supervised models*. These detectors include RADAR(Hu et al., 2023), MPU(Tian et al., 2023), RoBERTa-HC3(Guo et al., 2023), OpenAI-RoBERTa-based Detector(Solaiman et al., 2019b), ConDA(Bhattacharjee et al., 2023), and ArguGPT(Liu et al., 2023).

3.3.3 Commercial detectors

Since commercial detection tools are more commonly encountered in real-world scenarios, we also incorporate four widely used commercial detectors: GPTZero[2], ZeroGPT[3], Winston AI[4], and the latest commercial detector Pangram AI[5].

4 Key Findings

4.1 Discussion on Data Diversity

4.1.1 Detection difficulty differs across domains, generators, and operations

To examine detector generalization across domains, generators, and operations, we conducted Cross-Domain, Cross-Model, and Cross-Operation experiments. For supervised detectors, training samples were randomly drawn from all subsets except the testing one, which consisted of samples from a specific domain, generator, or operation (see Table A.4 for statistics). The results in Table 2 highlight several key findings, with detailed results provided in Section B.4.

Informal texts are generally easier to detect. Statistical analysis shows that detection performance is consistently higher on informal texts across different detector types. Overall, creative writings achieve the highest detection performance, with the overall AUROC achieving 73.41% , while formal texts such as essays and peer reviews are more challenging with detection accuracy of 64.61% and 93.09%, respectively. We attribute this trend to the generation style of LLMs. LLM-generated text tends to be more formal and standardized in wording and structure, reducing the stylistic gap in formal domains. In contrast, informal texts exhibit clearer stylistic differences, making detection easier.

Llama-generated texts exhibit higher resistance to detection. It can be observed that across all detector types, Llama3.1-8B consistently achieves the lowest detection accuracy among the tested models, with an average AUROC of 73.13%. Regarding this observation, we hypothesize that the informal nature of Llama3.1-8B's output effectively obscures these "AI-like" signatures, thereby making detection more challenging, and we will conduct further text analysis in Section 4.1.2.

Cross-lingual operations are the most challenging to detect. As shown in the results, texts generated via translation achieve the lowest detection accuracy across nearly all detectors, with an average of only 65.46%. This may be due to two factors: (1) translated texts often retain the original meaning with only superficial modifications, which increases the difficulty of feature extraction; and (2) existing detectors are generally not optimized for translated texts. However, translation is a common practice in academic misconduct, such as using AI to translate English papers into Chinese for plagiarism, which poses a significant challenge for AI text detection.

In contrast, compared to word/sentence-level modifications, **article-level modifications are generally easier to detect**, with the detection performance of the 'create' operation achieving the highest accuracy, which indicates that current detection models are less accurate for more fine-grained operations.

4.1.2 LLM combination does not consistently reduce detectability.

To examine the impact of LLM combinations, we conducted in-domain experiments on the LLM-Co dataset, using a train/test split of 0.2/0.8.

As shown in Figure 2, the influence on detectability depends on the intrinsic detectability of both the initial and secondary LLMs. Specifically, content

[2]https://gptzero.me
[3]https://www.zerogpt.com
[4]https://gowinston.ai
[5]https://www.pangram.com

(a) Cross-Domain

AUROC	Formal				Informal			Avg.	Var.
	xsum	pubmedqa	squad	review	writingprompts	blog	tweets		
Zero-shot Avg.	0.4307	0.4551	0.4904	0.4798	**0.5537**	**0.5483**	**0.6134**	0.5102	0.0169
Supervised Avg.	**0.8493**	0.8204	0.7150	0.7734	**0.8238**	**0.8719**	0.7779	0.8045	0.0158
Open-access Avg.	0.5335	0.5520	0.5311	0.5473	**0.6106**	**0.7063**	**0.6942**	0.5964	0.0229
Overall Avg.	0.6476	0.6461	0.6035	0.6309	**0.6915**	**0.7341**	**0.7079**	–	–

(b) Cross-Model

AUROC	Llama3.1-8B	DeepSeek-chat	gpt4o-mini	Qwen-plus	Avg.	Var.
Zero-shot Avg.	0.3990	0.5275	**0.5544**	0.5346	0.5039	0.0093
Supervised Avg.	0.8244	**0.9203**	0.9132	0.9125	0.8926	0.0027
Open-access Avg.	0.5452	0.5815	**0.6511**	0.5361	0.5784	0.0053
Overall Avg.	0.7313	0.8074	**0.8259**	0.7870	–	–

(c) Cross-Operation

AUROC	Creative Reformation			Linguistic Enhancement			Cross-lingual	Avg.	Var.
	create	rewrite	summary	polish	refine	expand	translate		
Zero-shot Avg.	**0.5549**	0.5061	0.4352	**0.5141**	0.4581	**0.5408**	0.5011	0.5015	0.0097
Supervised Avg.	**0.9284**	**0.9270**	**0.9249**	0.9133	0.9047	0.9099	0.8016	0.9014	0.0027
Open-access Avg.	**0.6156**	0.5502	**0.6761**	0.5583	0.5532	0.5139	**0.5654**	0.5761	0.0048
Overall Avg.	**0.7413**	**0.7105**	**0.7168**	0.7085	0.6862	0.7050	0.6546	–	–

Table 2: Average detection performance of different detectors in Cross-Domain/Model/Operation experiments. Bold values indicate the operation where each detector performs best/top3. Yellow colorboxes indicate the most detectable text types, while blue/red highlight the highest/lowest overall average performance, respectively.

generated by Llama generally exhibits increased detectability after secondary generation by another LLM, while content from other LLMs often shows reduced detectability after reprocessed by Llama.

In order to further explore this effect, we conducted a text analysis on LLM combinations that demonstrated statistically significant changes in detection difficulty, comparing the features of the text before and after secondary generation. The results shown in Table B.2a and B.2b indicate that Llama-generated content includes more quotation marks and fewer function words, displaying a more human-like stylistic pattern, which makes the text harder to detect. This provides a plausible explanation for the observed result. In contrast, secondary generation using LLMs that follow more standardized and formal generation patterns, such as gpt4o-mini, tends to increase detectability.

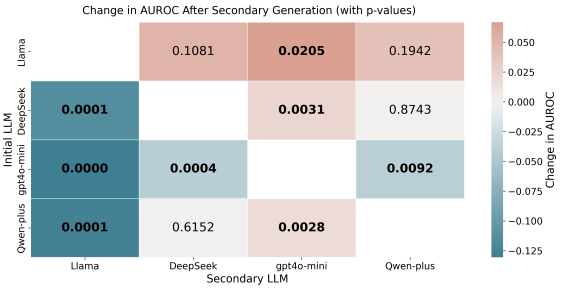
Based on the above analysis, we can further conclude that secondarily generated content tends to blend characteristics from both LLMs, with a pronounced alignment towards the patterns of the secondary LLM, which explains why final detectability primarily reflects the intrinsic detection difficulty of the secondary LLM. Consequently, combinations of more divergent LLMs lead to greater changes in detectability, while combinations of more similar LLMs produce comparatively minor variations that are statistically insignificant.

4.1.3 Operation combination tend to reduce detection accuracy.

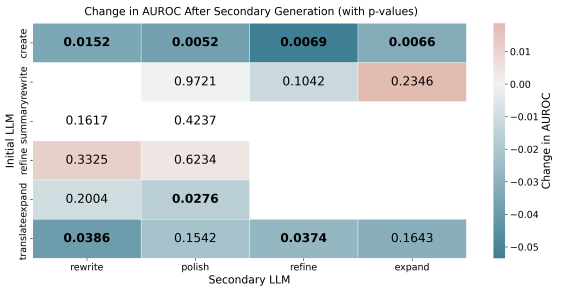
Following the same experimental setup as the LLM combination experiment, we conducted tests on the operation combination scenario, with the results shown in Figure 2.

Overall, operation combinations generally lead to a reduction in detection accuracy. Only a handful of combinations, such as refine–rewrite and refine–polish, show marginal performance gains, but none of which reach statistical significance. As depicted in the heatmap, most combinations exacerbate the detection difficulty, resulting in significant fluctuations in AUROC, with the majority of these changes being statistically significant. Notably, combinations that begin with the 'create' operation lead to a marked decline in performance after regeneration. Specifically, the create–refine sequence results in a 5.37% reduction in accuracy.

Further textual analysis shown in Table B.2c and B.2d indicates that combined operations tend to disrupt structural regularity while simultaneously enhancing stylistic diversity, thereby increasing the



(a) LLM combination



(b) Operation combination

Figure 2: Heatmap of AUROC change after secondary generation (LLM/operation combinations) relative to initial accuracy, with p-values indicating significance (where p-value < 0.05 bolded). 'Llama' presents Llama3.1-8B, and 'DeepSeek' presents DeepSeek-chat.

human-likeness of the generated text. In addition, these outputs frequently exhibit heightened emotionality, as reflected in the increased use of interrogative and exclamatory punctuation, which are characteristic of human writing. This potential correlation might present an interesting direction for future research.

4.2 Discussion on LLM variants.

In these experiments, in order to ensure fair comparison for supervised methods, an OOD training strategy across LLM families is adopted to standardize training data, similar to the settings in Cross-Model experiment. Table 3 summarize the average detection performance and one-tailed t-test results. Detailed results are provided in Section B.3.

4.2.1 Increased model scale leads to greater detection challenges

To investigate the impact of model scale, we selected five models from two LLM families for text generation: GPT (small: gpt4o-mini; large: gpt4o) and Qwen (small: Qwen2.5-7B-Instruct, Qwen3-8B; large: Qwen-plus).

The results show that within the GPT series, gpt4o is consistently harder to detect than gpt4o-mini across all conditions. In the Qwen series, Qwen2.5-7B-Instruct achieves the highest detection performance, with an AUROC of 73.34%, further supporting this observation. We hypothesize that as LLMs increase in scale, their enhanced language modeling capabilities enable them to produce text that more closely resembles human writing, thereby reducing detection effectiveness and increasing detection difficulty.

4.2.2 Texts from reasoning models are harder to detect

In this experiment, we selected four models from three LLM families: gpt5-mini, DeepSeek-chat, DeepSeek-reasoning, and Qwen-plus. For gpt5-mini and Qwen-plus, reasoning capabilities were activated by adjusting their cognitive configurations. Specifically for gpt5-mini, we compared low and high reasoning effort settings as a proxy for non-reasoning and reasoning modes. The training strategy was consistent with that used in the scaling experiment.

Overall, texts generated by reasoning modes are generally harder to detect, with statistically significant differences across all three model families. To further investigate the underlying causes, we conducted a textual analysis of the outputs generated by each model. Detailed results presented in Table B.1b. We observed that texts produced by reasoning models exhibit a notable increase in the use of quotation marks and a tendency toward higher lexical richness, while the proportion of function words is generally lower compared to non-reasoning modes. These patterns suggest that reasoning models generate text that is more conversational and closer to human-like expressions. This may be due to the model engaging in more extensive internal deliberation before generating text, thereby effectively reducing AI-like patterns in the output and increasing the difficulty for detectors to identify them.

Currently, as reasoning techniques become more common in modern LLMs, existing detection models largely fail to explicitly account for this phenomenon, highlighting an important direction for future research.

4.2.3 Typical instruction-tuning may make LLM texts easier to detect.

To investigate this issue, we selected four models from the open-source Llama and Qwen families:

(a) Model Scale

AUROC	gpt		Qwen		
	gpt4o-mini	gpt4o	Qwen2.5-7B-instruct	Qwen3-8B	Qwen-plus
Zero-shot Avg.	**0.5544**	0.5247	0.5124	**0.5406**	0.5346
Open-access Avg.	**0.6511**	0.6282	**0.6478**	0.6314	0.5361
Supervised Avg.	**0.9176**	0.9138	**0.9203**	0.8942	0.9061
Overall Avg.	**0.7467 (p=0.003)**	0.7306	**0.7334 (p=0.045)**	0.7274	0.7092

(b) Reasoning

AUROC	gpt		Qwen		DeepSeek	
	gpt5-mini (low)	gpt5-mini (high)	Qwen-plus	Qwen-plus-reasoning	DeepSeek-chat	DeepSeek-reasoning
Zero-shot Avg.	**0.5027**	0.4918	**0.5353**	0.4708	**0.5275**	0.5001
Open-access Avg.	**0.5206**	0.5083	**0.5361**	0.4921	**0.5815**	0.5767
Supervised Avg.	**0.8918**	0.8852	**0.9160**	0.9095	**0.9222**	0.9210
Overall Avg.	**0.6864 (p=0.0012)**	0.6772	**0.7111 (p=0.0021)**	0.6689	**0.7222 (p=0.0201)**	0.7120

(c) Typical instruction-tuning

AUROC	Llama		Qwen	
	Llama3.1-8B	Llama3.1-8B-instruct	Qwen3-8B	Qwen2.5-7B-instruct
Zero-shot Avg.	0.3990	**0.4007**	**0.5406**	0.5124
Open-access Avg.	0.5452	**0.5473**	0.6314	**0.6478**
Supervised Avg.	0.8299	**0.8321**	0.8023	**0.8496**
Overall Avg.	0.6316	**0.6337 (p=0.0629)**	0.6823	**0.6993 (p=0.0217)**

Table 3: Average detection performance across LLM variations. Yellow and blue colorboxes mark the most detectable LLM and best overall average performance, respectively, with bold values indicating the highest accuracy per group. Parenthetical values report one-sided t-test p-values.

Llama3.1-8B, Llama3.1-8B-instruct, Qwen2.5-7B-instruct, and Qwen3-8B.

The results show that Llama3.1-8B-instruct performs slightly worse than Llama3.1-8B, with overall average AUROCs of 63.16% and 63.37%, respectively. This may because typical instruction-tuning improves instruction following, yielding more controllable and consistent outputs. The improved structural consistency helps the model handle noise better and generate standardized text, which can enhance detectability. Further textual analysis shown in Table B.1c confirms this hypothesis, showing that instruction-tuned model generates relatively more structured text with a higher average number of clauses.

To further support our conclusion, we evaluated the detection performance of instruction-tuned models within the Qwen series. As a result, Qwen2.5-7B-instruct achieved a 1.7% higher detection rate than Qwen3-8B, indicating that text generated by instruction-tuned models is generally easier to detect than that produced by knowledge-distilled models. Unlike structured output generation, strong-to-weak knowledge distillation primarily facilitates knowledge transfer, allowing the smaller model to acquire semantic understanding and reasoning capabilities similar to the larger model, which makes its outputs comparatively harder to detect. Supporting this interpretation, further textual analysis shows that Qwen2.5-7B-instruct generates text with greater sentence-level depth, while Qwen3-8B tends to produce more conversational, human-like expressions.

4.3 Discussions on Detectors

4.3.1 Generalization

As shown in Table 4, **supervised models demonstrate the strongest overall generalization**, achieving high detection accuracy across all three tasks. Notably, *direct LM-based detectors*, such as DeBERTa and RoBERTa, maintain stable performance under distribution shifts, achieving AUROCs exceeding 96.00%. The results suggest that supervised models, even in few-shot settings, can learn the essential characteristics that distinguish LLM-generated text from human-written text. Consequently, increasing the quantity and diversity of training data enables the development of supervised detectors with both high accuracy and strong

AUROC		CD	CM	CO
Zero-shot	Logll	0.5202	0.4926	**0.3418**
	Rank	**0.5705**	**0.5574**	**0.2864**
	LogRank	0.5142	0.4933	0.2609
	Entropy	0.4903	0.5089	0.2587
	DetectGPT	0.5314	0.5036	0.2620
	Fast-D	**0.5459**	**0.5349**	0.2799
	Binoculars	0.4158	0.4212	0.2137
	DNA-GPT	0.4931	0.5193	0.2554
Supervised	Ghostbuster	0.5802	0.7723	0.7382
	GLTR	0.5831	0.6913	0.6861
	Perplexity	0.6361	0.7518	0.7470
	DeBERTa	**0.9702**	**0.9698**	**0.9854**
	RoBERTa	**0.8769**	**0.9688**	**0.9857**
	DistilBERT	0.9384	0.9517	0.9553
	XLM-R	0.9448	0.9664	0.9777
	XLNet	0.9456	0.9517	0.9752
	Mistral	0.7872	0.9595	0.9613
	Llama	0.8450	0.9372	0.9588
	ImBD	0.7798	0.8575	0.8895
	DtAL	0.7670	0.9337	0.9566
Open-access (Supervised)	RADAR	0.6589	0.6057	0.6044
	MPU	0.6512	0.6090	0.6088
	OpenAI-D	0.5210	0.5101	0.5145
	ArguGPT	**0.7550**	**0.6948**	**0.6953**
	HC3-D	0.6351	0.6309	0.6307
	ConDA	0.3571	0.4203	0.4029
Commercial	GPTZero	0.8431	0.8307	0.8430
	ZeroGPT	0.7298	0.6681	0.6800
	Winston	0.9400	0.9338	0.9525
	Pangram	**0.9608**	**0.9510**	**0.9610**

Table 4: The average detection performance for detectors in CD/CM/CO tasks. Yellow color boxes highlight the best detector in each category, while the blue color box and bold text emphasize the best/top 2 performances across all detectors.

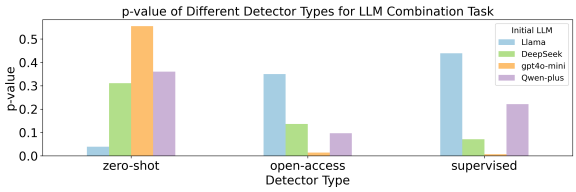
generalization.

In contrast, zero-shot methods show weaker generalization. They are more sensitive to generation settings and achieve relatively limited detection accuracy. While rank-based detector and Fast-DetectGPT perform somewhat better, they still lag behind supervised models. This limitation likely stems from their reliance on simple statistical signals, which are insufficient for capturing deeper patterns across LLM-generated texts. As LLMs evolve and generation scenarios become more complex, the performance of zero-shot methods may decline further.

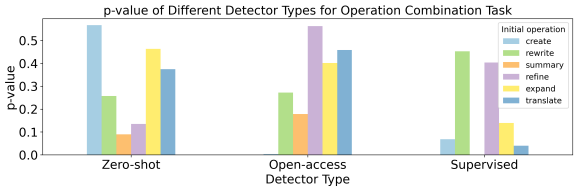
4.3.2 Robustness

To investigate the robustness of different types of detectors to perturbations, we further analyze the results of LLM/operation combination experiments, performing a detailed significance analysis on the changes in detection accuracy before and after secondary generation. The results presented in Figure 3 lead to the following conclusions.

All detectors exhibit relatively poor robustness when faced with LLM combinations, showing considerable fluctuations in detection performance. This highlights the need for developing more robust detectors that specifically address the challenges posed by LLM combination-based secondary generation, a problem that current models have largely overlooked.



(a) LLM combination



(b) Operation combination

Figure 3: Average significance (p-value) of detection performance changes for secondary generation under LLM/operation combination conditions.

On the other hand, **zero-shot models demonstrate relatively strong robustness to combinations of operations**, maintaining stable detection performance across different scenarios. Supervised models, by comparison, remain largely underexplored in this context, offering significant opportunities for further development.

4.3.3 Commercial Detectors

Commercial detectors generally achieve high detection accuracy. As shown in Table 4, among all detectors, Pangram delivers the best overall performance, with the highest accuracy across tasks, exceeding 95%. In contrast, ZeroGPT performs the worst across all tasks.

5 Conclusion

In this work, we introduce BR^{2}ED, a comprehensive benchmark for evaluating the detection of LLM-generated text across various domains, operations, and model architectures. Through a large-scale, multi-faceted dataset and extensive empirical analysis, BR^{2}ED facilitates systematic comparisons and provides valuable insights for advancing AI-generated text detection.

6 Limitations

BR^{2}ED has two key limitations. Firstly, its current scale is primarily suited for small- to medium-sized applications, and the evaluation of larger datasets is necessary for validating industrial use. Secondly, it excludes watermark-based detectors due to incompatible evaluation protocols. These constraints highlight areas for further development.

References

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

Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *Natural Language Processing and Information Systems*, pages 421–426, Cham. Springer International Publishing.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. Conda: Contrastive domain adaptation for ai-generated text detection. *arXiv preprint arXiv:2309.03992*.

Jiaqi Chen, Xiaoye Zhu, Tianyang Liu, Ying Chen, Chen Xinhui, Yiwen Yuan, Chak Tou Leong, Zuchao Li, Long Tang, Lei Zhang, and 1 others. 2025. Imitate before detect: Aligning machine stylistic preference for machine-revised text detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23559–23567.

Liam Dugan, Alyssa Hwang, Filip Trhlik, Josh Magnus Ludan, Andrew Zhu, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. Raid: A shared benchmark for robust evaluation of machine-generated text detectors. *arXiv preprint arXiv:2405.07940*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009.

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

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. *arXiv preprint arXiv:2401.12070*.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2251–2265.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 15077–15095. Curran Associates, Inc.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2019. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*.

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.

Yikang Liu, Ziyin Zhang, Wanyang Zhang, Shisen Yue, Xiaojing Zhao, Xinyuan Cheng, Yiwen Zhang, and Hai Hu. 2023. Argugpt: evaluating, understanding and identifying argumentative essays generated by gpt models. *arXiv preprint arXiv:2304.07666*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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.

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. *Preprint*, arXiv:1808.08745.

OpenAI. 2022. Introducing chatgpt.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *Preprint*, arXiv:1806.03822.

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

Shoumik Saha and Soheil Feizi. 2025. Almost ai, almost human: The challenge of detecting ai-polished writing. *arXiv preprint arXiv:2502.15666*.

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.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, and 1 others. 2019a. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019b. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203.

Zhen Tao, Yanfang Chen, Dinghao Xi, Zhiyu Li, and Wei Xu. 2026. Toward reliable detection of llm-generated texts: A comprehensive evaluation framework with cudrt. *ACM Transactions on Intelligent Systems and Technology*, 17(2):1–35.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, Qinghua Zhang, Ruifeng Li, Chao Xu, and Yunhe Wang. 2023. Multiscale positive-unlabeled detection of ai-generated texts. *arXiv preprint arXiv:2305.18149*.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2023. Ghostbuster: Detecting text ghostwritten by large language models. *arXiv preprint arXiv:2305.15047*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, and 1 others. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *arXiv preprint arXiv:2305.14902*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Junchao Wu, Runzhe Zhan, Derek Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia Chao. 2024. Detectrl: Benchmarking llm-generated text detection in real-world scenarios. *Advances in Neural Information Processing Systems*, 37:100369–100401.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *arXiv preprint arXiv:2305.17359*.

Qihui Zhang, Chujie Gao, Dongping Chen, Yue Huang, Yixin Huang, Zhenyang Sun, Shilin Zhang, Weiye Li, Zhengyan Fu, Yao Wan, and 1 others. 2024. Llm-as-a-coauthor: Can mixed human-written and machine-generated text be detected? *arXiv preprint arXiv:2401.05952*.

A Detailed information of Datase

A.1 Dataset descriptions

This section provides detailed descriptions of the datasets used across various tasks in BRED, outlining the specific data sources and their characteristics. The collected data underwent a cleaning process, which included the following steps: 1) removal of escape characters in the text; 2) elimination of sections containing non-English text; 3) removal of obvious LLM prompt phrases such as "Here is the translation" and "note." Detailed statistics are shown in Table A.1 and A.2.

LLM	Description	Avg text length	Total
Llama3.1-8B	~200 texts per (domain, operation)	287.89	9066
Llama3.1-8B-instruct		294.54	9058
DeepSeek-chat		255.41	9190
DeepSeek-reasoning		252.01	9196
gpt4o-mini		265.63	9356
gpt4o		253.27	9260
gpt5-mini (low)		272.96	9200
gpt5-mini (high)		272.28	9168
Qwen2.5-7B-instruct		242.22	9350
Qwen3-8B		266.99	9444
Qwen-plus		267.11	9236
Qwen-plus (reasoning)		241.8	9066
Total	–		110590

Table A.1: The statistics of texts generated by one-step generation.

Initial	Description	Avg text length	Total
Llama3.1-8B	~1200 texts per combination	303.05	3420
DeepSeek-chat		283.89	3495
gpt4o-mini		288.43	3435
Qwen-plus		270.95	3513
create	~1200 texts per combination	305.61	4671
rewrite		308.78	3480
summary		205.04	2310
refine		254.69	2310
expand		354.36	2352
translate		279.94	4632
Total	–		33618

Table A.2: The statistics of texts generated by two-step generation.

A.2 Human-written text collection

Xsum Our news dataset is based on the XSum(Narayan et al., 2018) dataset, which consists of BBC articles and ac companying single sentence summaries. The dataset contains 226,711 BBC articles archived via the Wayback Machine, spanning nearly a decade (from 2010 to 2017) and covering a wide range of domains, including news, politics, sports, weather, business, technology, science, health, family, education, entertainment, and the arts.

Writingprompts Our creative writing dataset is constructed based on the WritingPrompts(Fan et al., 2018) dataset, which comprises prompts and corresponding stories collected from the Reddit's r/WritingPrompts[1] before 2018. The dataset has been preprocessed to remove noise, with story lengths constrained to a minimum of 30 words and a maximum of 1,000 words.

Pubmedqa Our academic article data is sourced from the publicly available PubMedQA dataset(Jin et al., 2019), which is constructed from PubMed[2] abstracts and includes abstract samples along with human-generated question-answer pairs. In accordance with the generation operation defined in this study, we primarily selected abstracts from the PubMedQA dataset for data collection.

Squad Our Wikipedia data is sourced from the publicly available SQuAD(Rajpurkar et al., 2016, 2018) dataset released by Stanford University, which is a reading comprehension dataset comprising over 100,000 questions based on a set of Wikipedia articles. Similar to our approach with PubMedQA, we selected the context as part of the human-written text in our data collection process.

Review Recent studies have documented a growing trend of peer reviewers employing LLMs to generate or refine review comments. To account for this phenomenon in our dataset construction, we collected peer reviews from Openreview[3], limiting our selection to those published before 2019 in order to minimize potential LLM involvement in the generation of the comments.

Blog To enrich the variety of text types in our dataset, we also collected blog articles, which are sourced from a publicly available dataset(Schler et al., 2006), consisting of posts from 19,320 bloggers gathered from Blogger website[4] in August 2004.

Tweets Finally, considering the challenges of short-text detection in the current field of LLM-generated text detection, we collected 2,800 tweets from Twitter dataset(Go et al., 2009). After data cleaning, the average text length is approximately 30 words.

[1] https://www.reddit.com/r/WritingPrompts/
[2] https://pubmed.ncbi.nlm.nih.gov/
[3] https://openreview.net/
[4] https://www.blogger.com/

Prompt	Description
Part I	This section defines the role and context of the task. It clarifies the type of operation to be performed.
	example: 'You are a (e.g., news writer), tasked with improving and continuing a `[text type]`.'
Part II	This section defines the operation you are performing. This part will be one of the following: 1) create, 2) rewrite, 3) summary, 4) refine, 5) polish, 6) expand, 7) translate. Each option specifies the type of modification to the text.
	example: 'Your task is to continue the article in a consistent tone, adding depth and seamlessly transitioning from the provided content...'
Part III	This section includes the provided content that needs to be worked on the given text.
	example: 'The given text is as follows: `[insert the original article or content]` '
Part IV	This section contains any additional notes, instructions, or specific clarifications for the task.
	example: 'Note: Your rewriting needs to conform to the standards and expressions of `[text type]`. '

Table A.3: Prompt templates for text generation.

A.3 Generation operations

create The 'create' operation of LLMs involves generating new content based on user prompts. Specifically, we adopt complete operation, which the first 25 percent of a human-written text is used as input, and the LLMs are prompted to complete the remaining parts based on the initial fragment.

rewrite The 'rewrite' operation involves prompting a LLM with an existing text, instructing it to reorganize paragraphs to change the writing style, producing a new text that preserves the original meaning while altering its linguistic form. This operation is applied at the **document level**, often resulting in structural changes to the original text.

summary The 'summary' operation refers to the process in which LLMs condense and rephrase the original text after receiving it, extracting the core ideas and key information to produce a clear and concise summary, similar to the process of summarizing an abstract. *The text produced after the summary operation typically does not retain the original content.*

refine The 'refine' operation involves simplifying, condensing, or improving the text after receiving it, making it more concise, clear, and easier to comprehend. This operation is applied at the **sentence level**, where the refined text *retains some parts of the original content but presents it in a more concise manner*.

polish The 'polish' operation refers to the process by which a LLM enhances the sentence-level quality of a given text while preserving its original content and meaning. This operation typically involves correcting grammatical errors, refining word choices, improving sentence fluency, and ensuring stylistic consistency. Unlike rewrite or refine which may restructure or compress the input, *the polish operation maintains the original sentence structure and focuses only on* ***linguistic clarity*** *and readability.*

expand The 'expand' operation denotes the procedure through which a LLM adds relevant content to a given text, such as additional explanations, examples, or background information. While keeping the original meaning intact, the model makes the text more detailed and informative. This operation is also performed at the sentence level and does not alter the overall structure of the original text.

translate The 'translate' operation denotes the sequence-to-sequence transformation process in which a language an LLM converts a text from one language to another while preserving the original meaning, tone, and style as much as possible. In this study, since our dataset is composed of English texts, we apply the translate operation in two stages. First, each human-written text is translated from English to another, and then translated back into English.

A.4 Prompt templates for text generation

The example prompts we used in the generation of LLM texts are shown below in Table A.3.

A.5 Experimental Setup

We define several tasks for performance evaluation, including 1) Cross-Domain (CD) experiment, Cross-Model (CM) experiment, Cross-Operation (CO) experiment, 2) LLM/Operation-Crossover (LLM/Op-Co) experiment, and 3) LLM-Variant (LLM-Va) experiment. In particular, the detailed train/test splits of the three categories of tasks are shown below in Table A.4.

Tasks	Training strategy	Train/test split
CD/ CM/ CO	OOD	4000/-
LLM-Co/ Op-Co	in-domain	0.2/0.8
LLM-Va	OOD	4000/-

Table A.4: Train/ Test dataset splits

(a) Scale

Average Statistics	gpt5-mini (low)	gpt5-mini (high)	DeepSeek-chat	DeepSeek-reasoning	Qwen-plus	Qwen-plus-reasoning
.	0.43	0.439	0.44	0.434	0.446	0.438
,	0.36	0.358	0.355	0.352	0.334	0.329
’	0.081	0.092	0.091	0.099	0.092	0.102
"	0.035	0.054	0.044	0.048	0.046	0.055
:	0.027	0.024	0.028	0.028	0.042	0.035
?	0.029	0.024	0.027	0.025	0.026	0.026
!	0.017	0.019	0.014	0.014	0.014	0.014
Sentence count	16.18	16.08	15.77	15.59	16.83	14.77
Sentence length	24.6	24.74	23.51	23.41	24.31	24.83
Word length	4.84	4.83	4.84	4.87	4.87	4.98
Type-Token Ratio	0.607	0.61	0.616	0.623	0.62	0.63
Function word ratio	0.357	0.353	0.357	0.354	0.353	0.336
Syntactic depth	2.51	2.49	2.46	2.46	2.51	2.5

(b) Reasoning

Average Statistics	gpt4o-mini	gpt4o	Qwen2.5-7B-instruct	Qwen-plus
.	0.4228	0.4067	0.4193	0.446
,	0.3562	0.3377	0.3484	0.334
’	0.0997	0.1163	0.1073	0.092
"	0.0507	0.0602	0.0602	0.046
:	0.0252	0.039	0.0262	0.042
?	0.0261	0.0239	0.0249	0.026
!	0.0198	0.0161	0.0136	0.014
Sentence count	15.64	15.41	15.29	16.83
Sentence length	24.01	23.43	23.84	24.31
Word length	4.85	4.91	4.91	4.87
Type-Token Ratio	0.601	0.607	0.618	0.62
Function word ratio	0.369	0.359	0.356	0.353
Syntactic depth	2.51	2.49	2.69	2.51

(c) Typical instruction-tuning

Average Statistics	Qwen2.5-7B-instruct	Qwen3-8b	Llama3.1-8B-instruct	Llama3.1-8B
.	0.4193	0.426	0.4011	0.4005
,	0.3484	0.3432	0.3782	0.3786
’	0.1073	0.1139	0.1185	0.118
"	0.0602	0.0551	0.0457	0.0465
:	0.0262	0.0219	0.0211	0.0214
?	0.0249	0.0252	0.0232	0.0232
!	0.0136	0.0144	0.0121	0.0118
Sentence count	15.29	16.1	17.59	17.25
Sentence length	23.84	24.3	24.18	23.97
Word length	4.91	4.8	4.84	4.87
Type-Token Ratio	0.618	0.601	0.607	0.61
Function word ratio	0.356	0.368	0.358	0.357
Syntactic depth	2.69	2.5	2.44	2.43

Table B.1: Textual analysis of LLM variation. The terms `initial` and `secondary` refer to texts generated in the first and second generation of AI, respectively. The core punctuation marks considered in this analysis include: `.`, `,`, `’`, `"`, `:`, `?`, and `!`, and the first seven rows of the table present the average proportions of these core punctuation marks across the examined texts.

B Detailed results of experiments

B.1 Textual Analysis

To further analyze the results of LLM variation (scale/reasoning/typical instruction) and LLM/operation combination experiment, we performed a textual analysis on the dataset, with the results shown in Tables B.1 and B.2.

B.2 Results of LLM-Co and Op-Co Experiments

To make the results more intuitive, we visualized the outcomes of different detectors as heatmaps, as shown in Figures B.1 and B.2.

B.3 Results of LLM Variety Experiments

Detailed results of all detectors are shown in Figure B.3, B.4, and B.5.

B.4 Results of CD/CM/CO Experiments

Detailed results of all detectors are shown in Figure B.6, B.7, and B.8.

(a) LLM combination

Average Statistics	Llama3.1-8B/gpt4o-mini		DeepSeek-chat/gpt4o-mini		Qwen-plus/gpt4o-mini	
	Initial	Secondary	Initial	Secondary	Initial	Secondary
.	0.3613	0.3657	0.4538	0.4090	0.4739	0.4455
,	0.4471	0.4589	0.4225	0.4331	0.4062	0.4231
'	0.0977	0.0486	0.0114	0.0209	0.0049	0.0127
"	0.0528	0.0654	0.0256	0.0373	0.0343	0.0279
:	0.0159	0.0209	0.0318	0.0319	0.0321	0.0300
?	0.0177	0.0193	0.0418	0.0464	0.0394	0.0436
!	0.0076	0.0212	0.0131	0.0214	0.0091	0.0172
Sentence count	16.76	15.77	15.04	15.21	21.44	20.19
Sentence length	22.50	21.48	20.02	20.83	20.38	20.87
Word length	5.23	5.28	5.10	5.20	5.12	5.20
Type-Token Ratio	0.6480	0.6193	0.6429	0.6200	0.6310	0.6119
Function word ratio	0.3241	0.3262	0.3224	0.3257	0.3140	0.3206
Syntactic depth	2.419	2.461	2.316	2.434	2.295	2.374

(b) LLM combination

Average Statistics	DeepSeek-chat/Llama3.1-8B		gpt4o-mini/Llama3.1-8B		gpt4o-mini/DeepSeek-chat		gpt4o-mini/Qwen-plus		Qwen-plus/Llama3.1-8B	
	Initial	Secondary	Initial	Secondary	Initial	Secondary	Initial	Secondary	Initial	Secondary
.	0.4598	0.3308	0.2886	0.3045	0.3027	0.2749	0.3163	0.2937	0.4689	0.3426
,	0.4251	0.4250	0.2919	0.4099	0.3086	0.2339	0.3231	0.2512	0.4257	0.4237
'	0.0145	0.1117	0.0387	0.1049	0.0036	0.0019	0.0342	0.0017	0.0049	0.1047
"	0.0258	0.0804	0.0503	0.0586	0.0558	0.0059	0.0433	0.0099	0.0219	0.0727
:	0.0287	0.0271	0.0553	0.0273	0.0161	0.0283	0.0205	0.0285	0.0249	0.0231
?	0.0330	0.0166	0.0247	0.0185	0.0236	0.0125	0.0190	0.0054	0.0249	0.0241
!	0.0132	0.0084	0.0279	0.0155	0.0236	0.0125	0.0151	0.0054	0.0121	0.0086
Sentence count	42.5058	42.5020	29.1918	40.9869	30.8610	14.7775	19.4301	15.1908	20.8572	23.6604
Sentence length	20.0226	20.8287	20.3767	20.8690	19.6658	14.7775	19.4301	15.1908	20.8572	23.6604
Word length	4.9185	4.9856	4.9359	5.1413	4.8881	5.2367	4.9359	5.1413	5.0791	5.2586
Type-Token Ratio	0.5912	0.6473	0.5933	0.5595	0.5933	0.5595	0.5806	0.5549	0.6543	0.6541
Function word ratio	0.3469	0.3035	0.3446	0.3021	0.3446	0.3021	0.3479	0.2928	0.3183	0.3005
Syntactic depth	2.3334	2.4090	2.3695	2.3690	2.4215	2.1273	2.3778	2.1999	2.3645	2.4134

(c) Operation combination

Average Statistics	create/rewrite		create/polish		create/refine		create/expand	
	Initial	Secondary	Initial	Secondary	Initial	Secondary	Initial	Secondary
.	0.4421	0.4512	0.4286	0.4382	0.4374	0.4733	0.4211	0.3936
,	0.4413	0.4371	0.4575	0.4317	0.4332	0.3963	0.4496	0.5096
'	0.0327	0.0310	0.0421	0.0359	0.0403	0.0395	0.0448	0.0288
"	0.0191	0.0108	0.0249	0.0253	0.0289	0.0229	0.0298	0.0154
:	0.0336	0.0363	0.0257	0.0341	0.0320	0.0377	0.0315	0.0261
?	0.0269	0.0287	0.0192	0.0310	0.0256	0.0272	0.0194	0.0233
!	0.0043	0.0049	0.0020	0.0038	0.0027	0.0030	0.0037	0.0032
Sentence count	16.26	17.35	14.30	15.12	16.67	13.91	16.04	27.89
Sentence length	20.09	21.12	20.58	21.56	21.09	19.57	20.59	24.13
Word length	5.41	5.54	5.37	5.51	5.41	5.63	5.36	5.45
Type-Token Ratio	0.6478	0.6475	0.6658	0.6597	0.6388	0.6854	0.6497	0.5584
Function word ratio	0.3237	0.3082	0.3276	0.3060	0.3255	0.2888	0.3243	0.3228
Syntactic depth	2.4350	2.4548	2.4778	2.4389	2.4940	2.3519	2.4710	2.6553

(d) Operation combination

Average Statistics	expand/polish		translate/rewrite		translate/refine	
	Initial	Secondary	Initial	Secondary	Initial	Secondary
.	0.3612	0.3912	0.4085	0.4332	0.4171	0.4325
,	0.5174	0.4764	0.3451	0.4125	0.3355	0.4118
'	0.0447	0.0368	0.0860	0.0294	0.0790	0.0419
"	0.0241	0.0236	0.0819	0.0408	0.0907	0.0384
:	0.0159	0.0227	0.0313	0.0364	0.0283	0.0264
?	0.0272	0.0386	0.0341	0.0375	0.0373	0.0424
!	0.0043	0.0049	0.0020	0.0038	0.0027	0.0032
Sentence count	16.26	17.35	14.30	15.12	16.67	13.91
Sentence length	20.09	21.12	20.58	21.56	21.09	19.57
Word length	5.41	5.54	5.37	5.51	5.41	5.63
Type-Token Ratio	0.6478	0.6475	0.6658	0.6597	0.6388	0.6854
Function word ratio	0.3237	0.3082	0.3276	0.3060	0.3255	0.2888
Syntactic depth	2.7236	2.5187	2.3657	2.2999	2.5037	2.3630

Table B.2: Textual analysis of LLM/Operation combinations. In the first row of the table, `A/B` indicates that A represents the first LLM/operation and B denotes the secondary LLM/operation. The terms `initial` and `secondary` refer to texts generated in the first and second generation of AI, respectively. The core punctuation marks considered in this analysis include: `.`, `,`, `'`, `"`, `:`, `?`, and `!`, and the first seven rows of the table present the average proportions of these core punctuation marks across the examined texts.

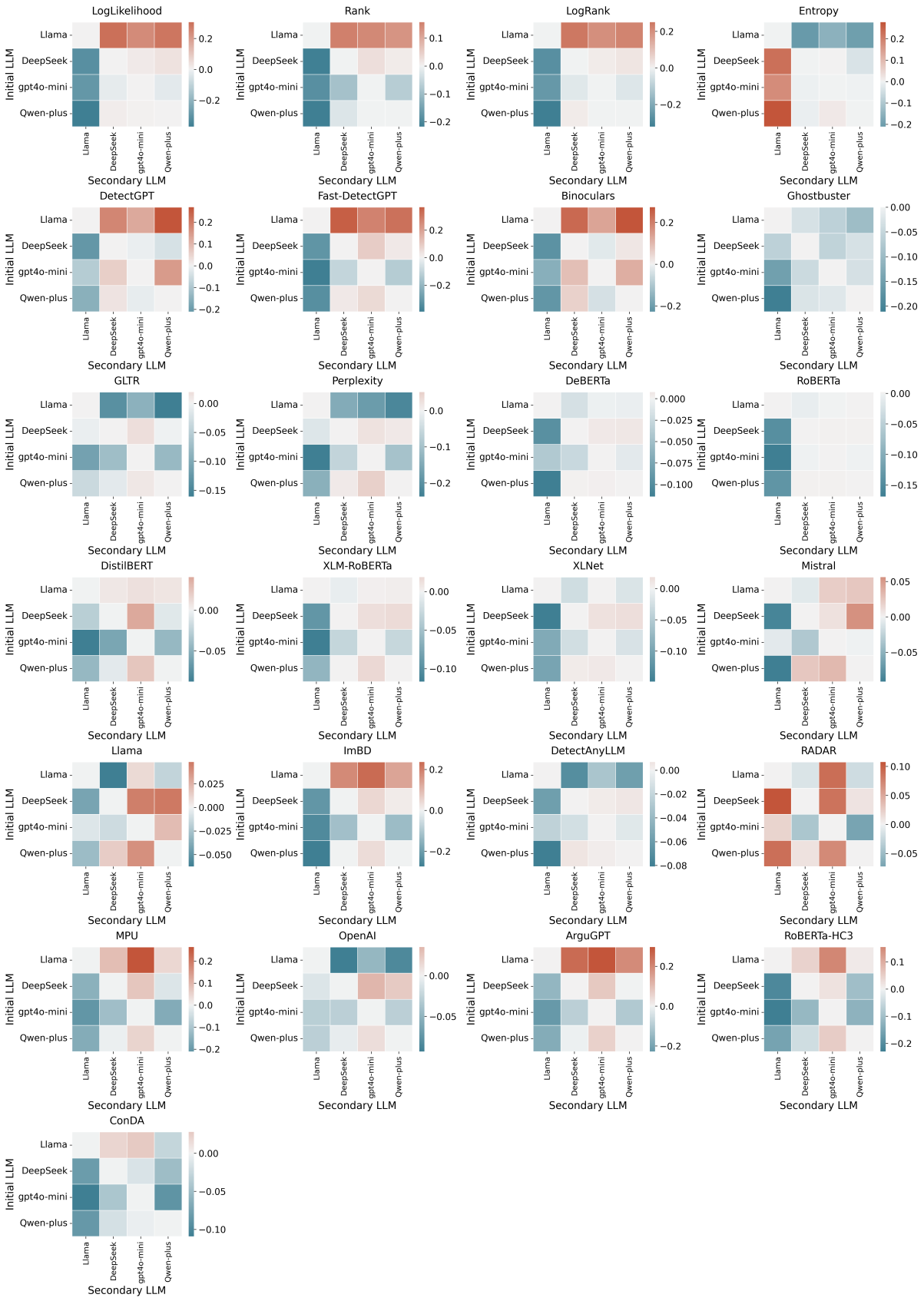


Figure B.1: Heatmap of AUROC changes for all detectors after secondary generation (LLM combinations), relative to their initial accuracy. 'Llama' presents Llama3.1-8B and 'DeepSeek' presents DeepSeek-chat.

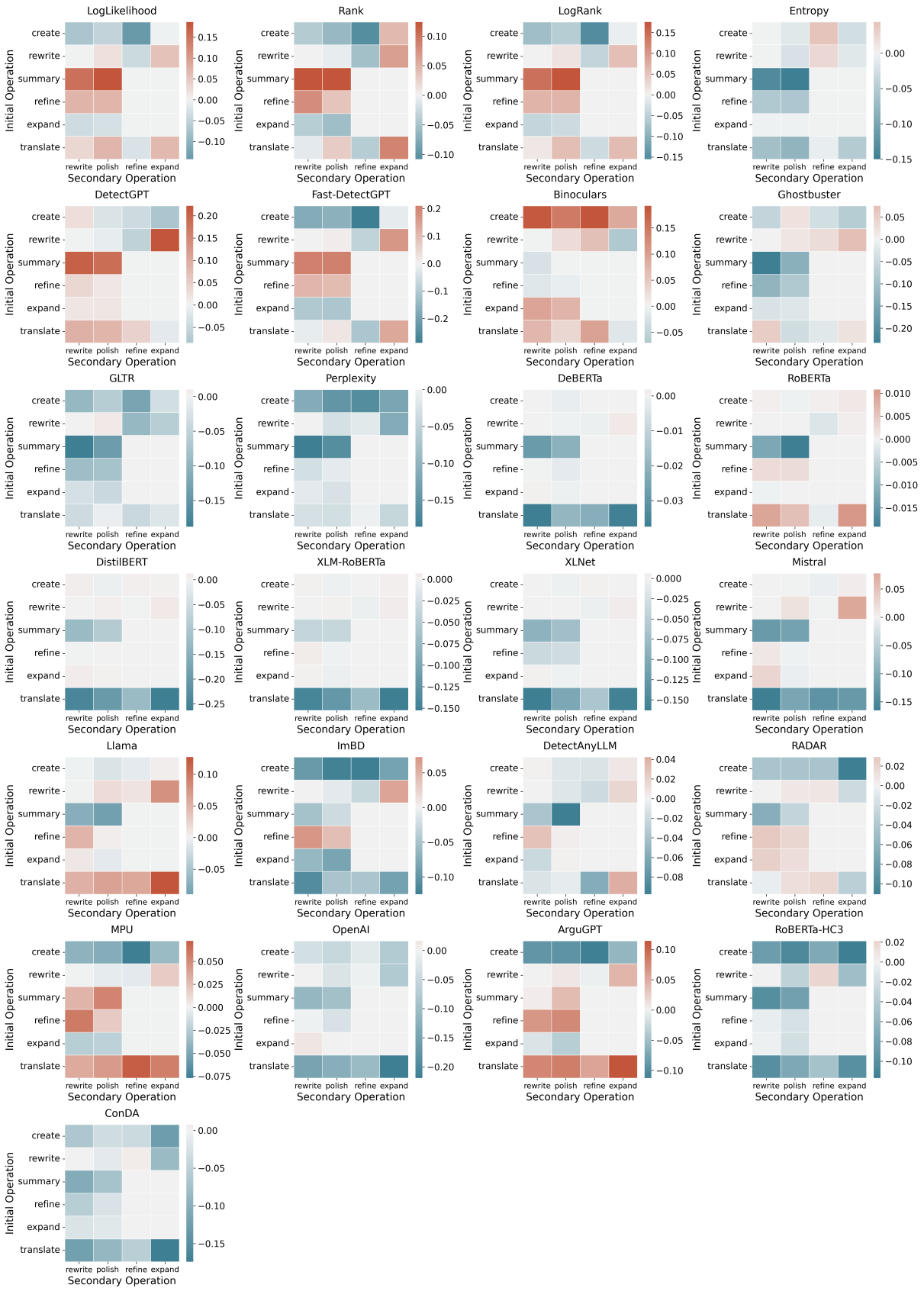


Figure B.2: Heatmap of AUROC changes for all detectors after secondary generation (LLM/operation combinations), relative to their initial accuracy.

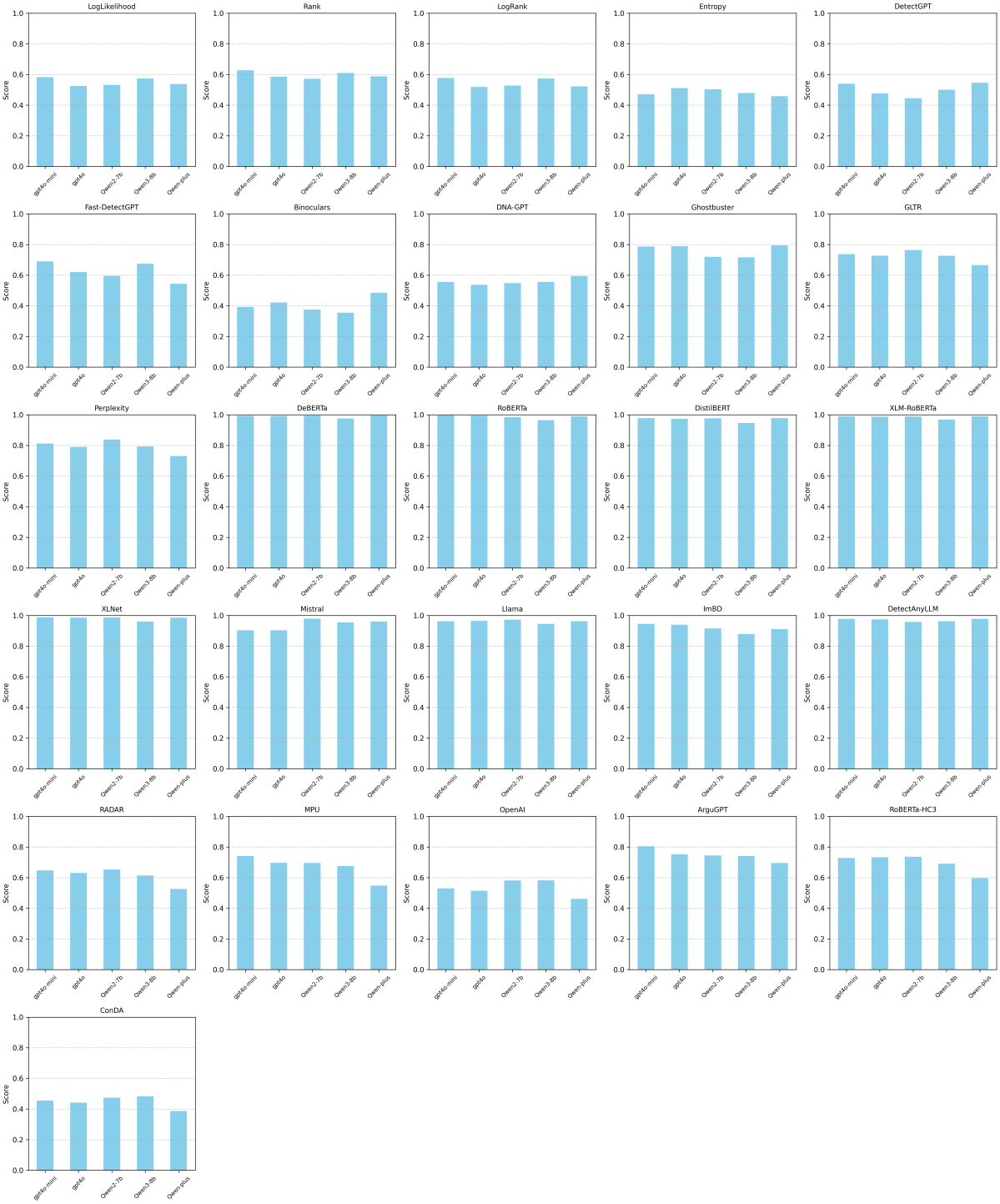


Figure B.3: Detection performance (AUROC) of all detectors on different scales of LLMs.

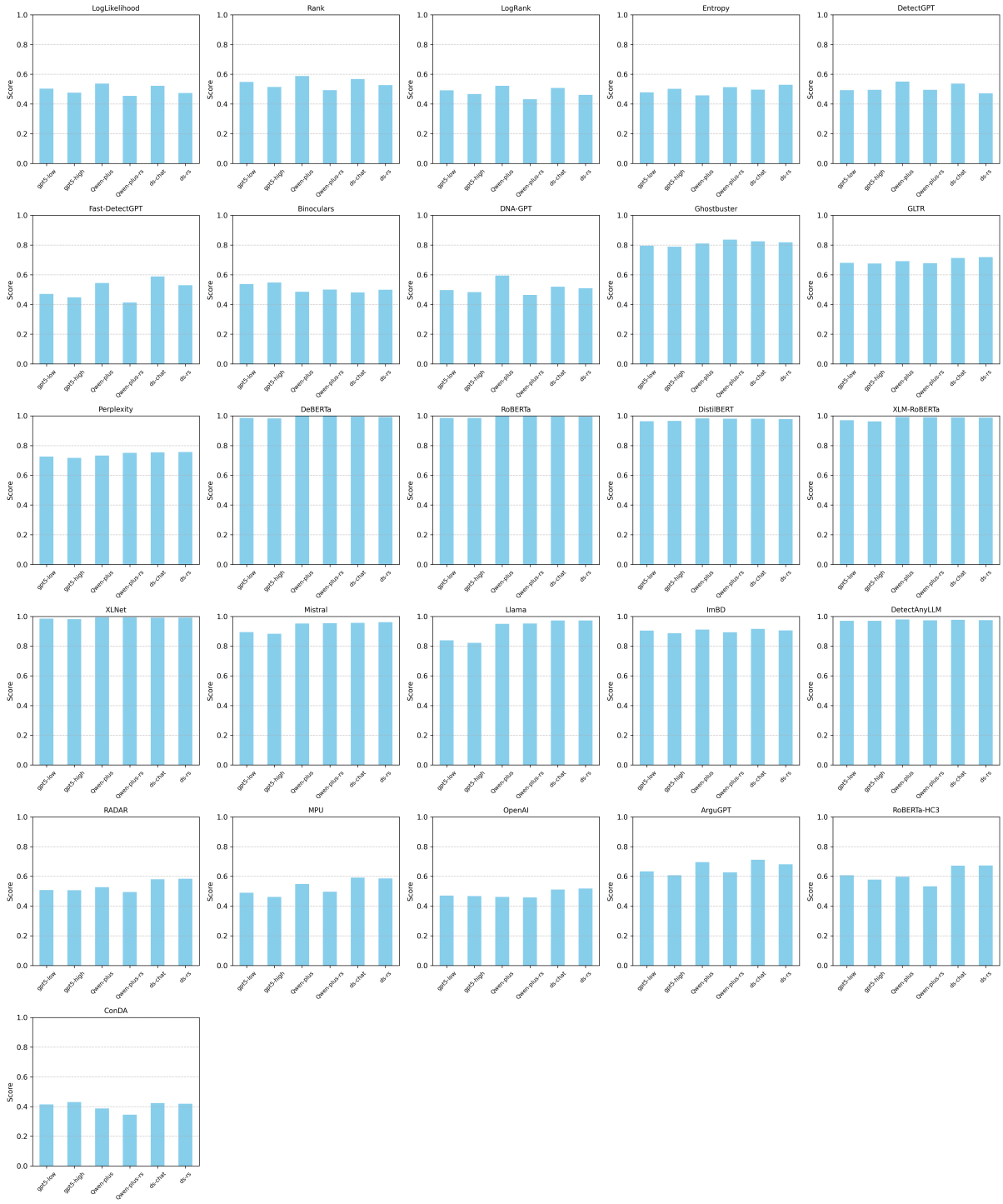


Figure B.4: Detection performance (AUROC) of all detectors across reasoning and non-reasoning models/modes. ‘Qwen-plus-rs’ presents Qwen-plus-reasoning, ‘ds-chat’ presents DeepSeek-chat, and ‘ds-rs’ presents DeepSeek-reasoning.

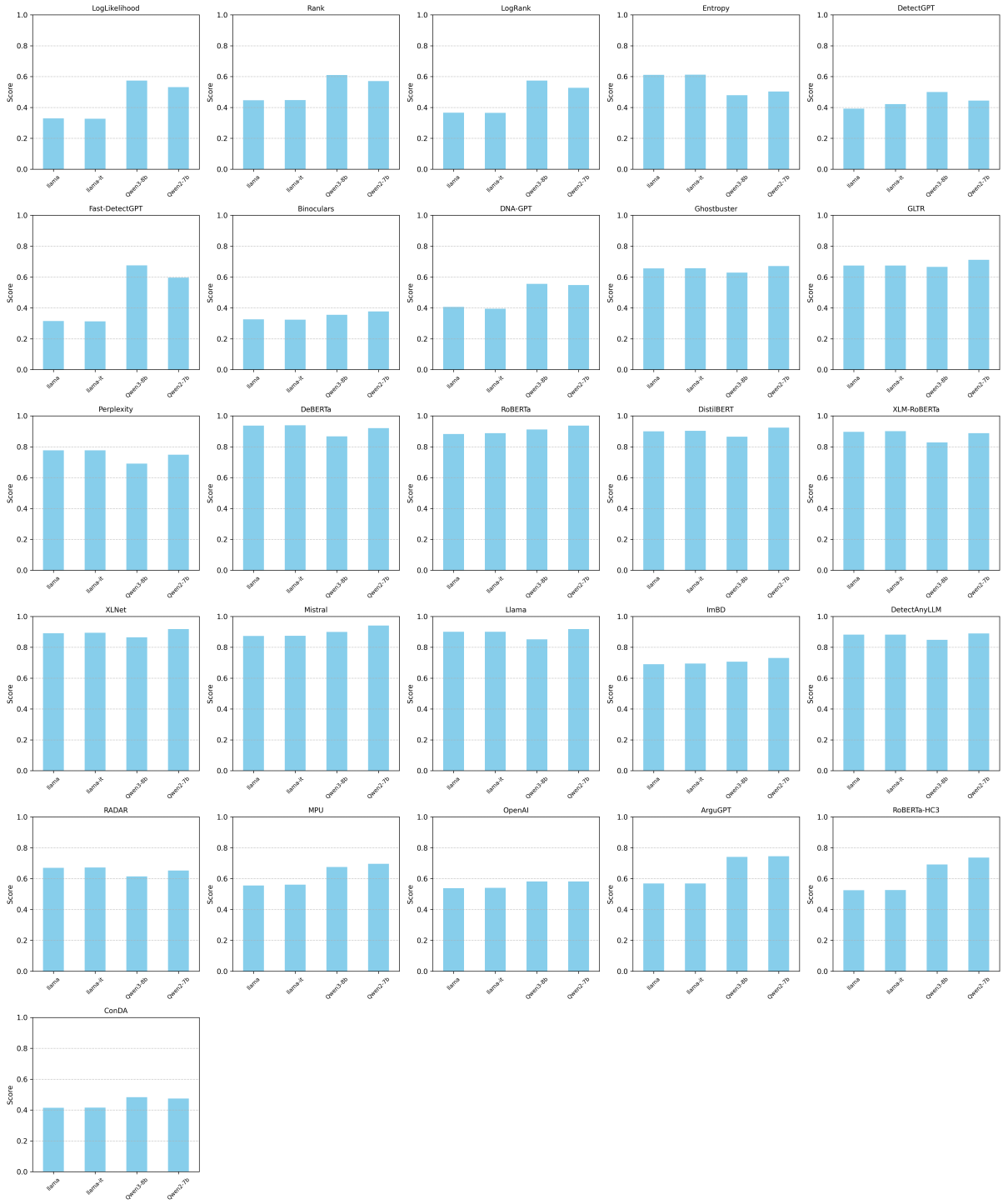


Figure B.5: Detection performance (AUROC) of all detectors across typical and non-typical instruction-tuned models/modes. 'llama' presents Llama3.1-8B, 'llama-it' presents Llama3.1-8B-instruct, and 'Qwen2-7b' presents Qwen2.5-7B-instruct.

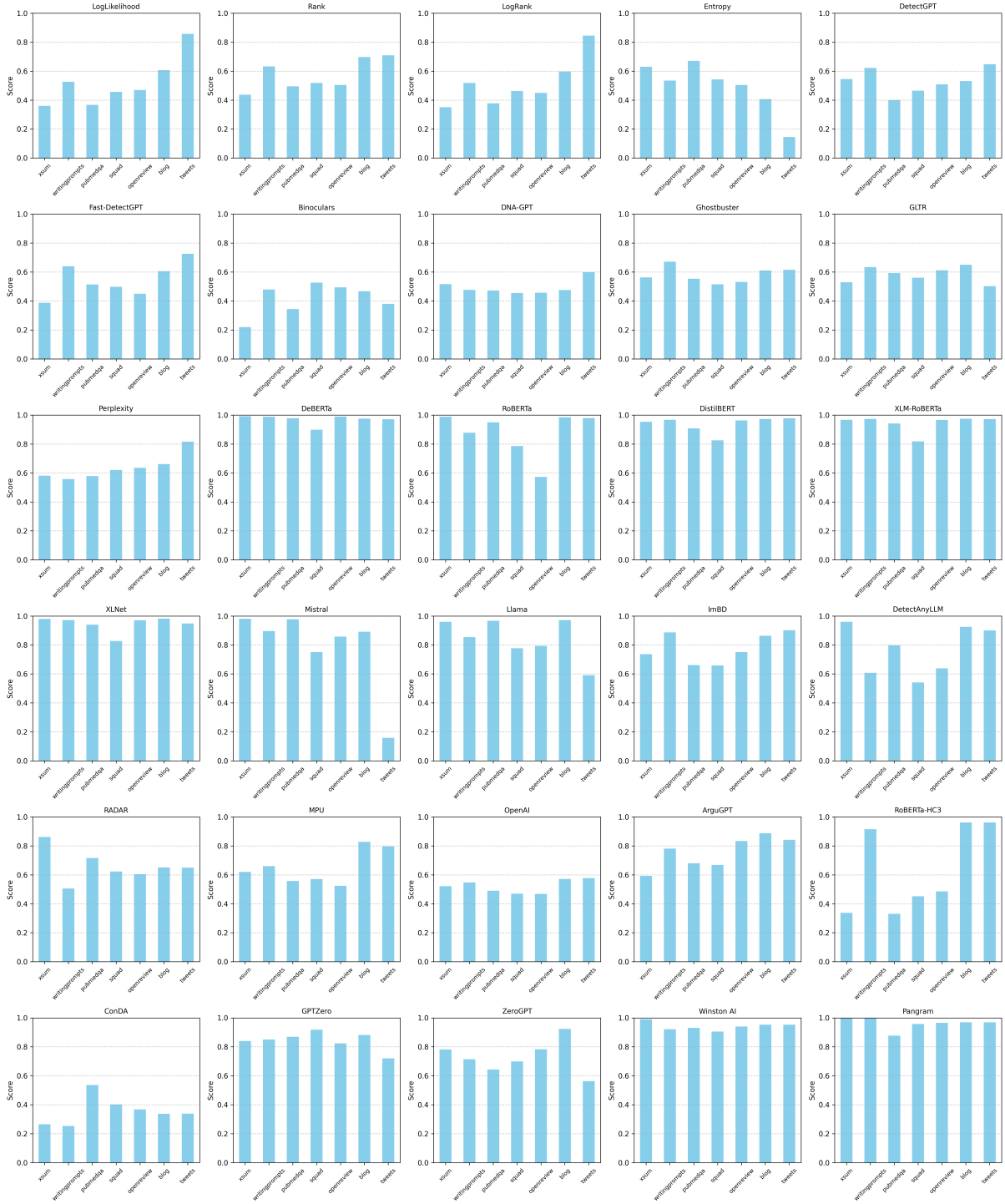


Figure B.6: Detection performance (AUROC) of all detectors on Cross-Domain experiment.

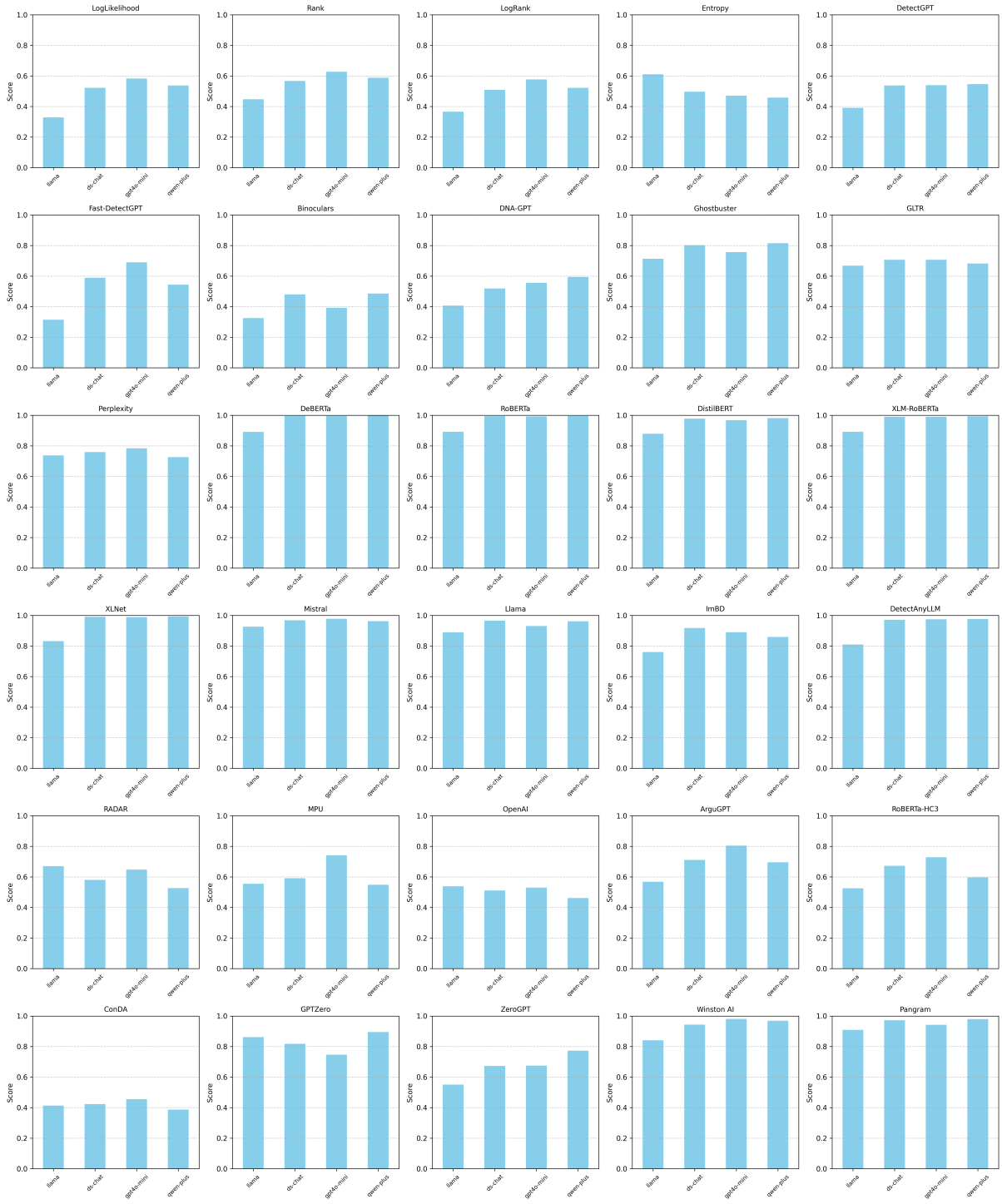


Figure B.7: Detection performance (AUROC) of all detectors on Cross-Domain experiment. ‘llama’ presents Llama-3.1-8B and ‘ds-chat’ presents DeepSeek-chat.

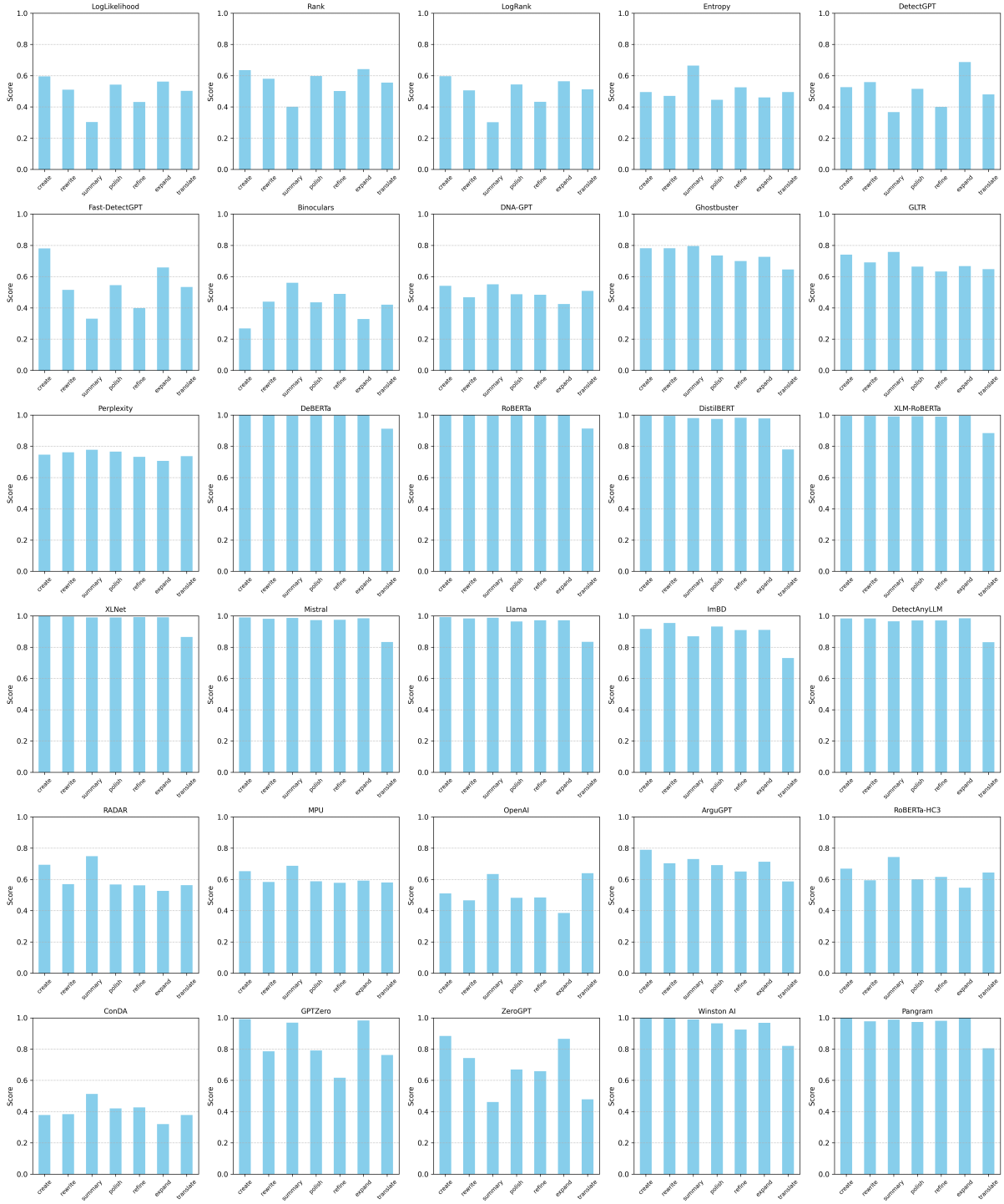


Figure B.8: Detection performance (AUROC) of all detectors on Cross-Domain experiment.