USING LLMS TO DETECT LLM-GENERATED TEXTS: A CROSS-GENERATION ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Detecting LLM-generated texts is becoming increasingly important as LLMs have been widely adopted and used in various domains, while their generative outputs can also create risks of misuses and other adverse effects. Various approaches have been developed to design automated detectors, but no single solution remains reliable across models, domains, and evaluation settings. General-purpose LLMs provide an alternative and flexible way to make human/LLM authorship judgements on given texts with explanations to support their decisions, but our understanding of their detection behaviour remains limited, particularly on differences between self-detection (i.e., detection by the same text-generation LLM) and cross-detection (i.e., detection by other LLMs) and how they have evolved over time. This paper presents our study of using 15 LLMs from three generations as both text generators and detectors. We constructed 1,000 human-generated texts (HGTs) and 1,000 LLM-generated texts (LGTs) per model, and instructed each of the 15 detectors to perform binary classification tasks on all 15,000 LGTs and 1,000 HGTs, obtaining 233,428 valid judgments (excluding null/badly-formed ones) with associated natural-language explanations. Our results suggest that detection performance is more strongly related to detector generation than generator generation, while texts produced by the most recent generators remain harder to detect. Rigorous statistical comparisons of self- and cross-detection results provide mixed and model-specific results, suggesting no conclusive evidence of a consistent self-detection advantage or disadvantage across LLMs. Lastly, we analysed error patterns from three perspectives: an analysis based on false-positive and false-negative rates, an LLM-assisted analysis of LLM-generated explanations, and a literature-guided cue-based analysis. The results further show that detectors based on the first (earliest) generation of LLMs have a tendency to predict LGTs as HGTs, detectors based on the second generation of LLMs show stronger tendencies to predict HGTs as LGTs, and detectors based on the latest generation of LLMs have more balanced false-positive and false-negative rates. In addition, potential inconsistent applications of textual cues that would lead to different judgements by different LLMs are also identified and reported.

1 INTRODUCTION

In recent years, large language models (LLMs) have developed from research prototypes into various applications used across a wide range of domains, such as education, science, medicine, programming, journalism, and creative work (Li et al., 2024). While these applications can help improving productivity and work efficiency, the same generative capability enables LLMs to produce convincing content that could reinforce false narratives, raising concerns about the generation and dissemination of misinformation and disinformation at scale (Vykopal et al., 2024). It may also be misused to create personalised phishing messages and deceptive online reviews (Czybik et al., 2026; Kovács, 2024). On the other hand, falsely accusing legitimate authors of using LLMs can also cause reputational harm (Liang et al., 2023; Layton et al., 2026).

LLM-generated text (LGT) detection has therefore emerged as an important research topic, answering questions of textual provenance, plagiarism, and distinction between human- and machine- generated content (Dugan et al., 2024; Lee et al., 2023; Bhattacharjee & Liu, 2024; Wu et al., 2025b). Prior work spans supervised classifiers, statistical methods, rewriting-based approaches, and water-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

marking (Mitchell et al., 2023; Hans et al., 2024; Kirchenbauer et al., 2023). Various efforts have also been made to evaluate large benchmarks and commercial detectors for detecting LGTs (Wang et al., 2024; Dugan et al., 2024; Layton et al., 2026). Despite such substantial efforts from both research and industry communities, such detectors often generalise poorly across unseen models and domains, while they remain vulnerable to prompting and style-shifting attacks (Tufts et al., 2025; Pedrotti et al., 2025). It is also worth noting that when detectors are required to keep low false positive rate, the performance of detecting LGT decline significantly (Tufts et al., 2025). Moreover, detection becomes even more challenging for human-LLM co-authored texts, where detectors need to explicitly identify the source of individual segments (Su et al., 2025).

An increasingly attractive approach is to simply use a general-purpose LLM as the detector, where the selected LLM takes a suspected text and returns a binary decision (Hans et al., 2024; Ji et al., 2025). It can also provide a natural-language explanation on top of the binary decision label (Ji et al., 2025). The study closest to ours was conducted by Ji et al. (2025), where six open- and closed-source LLMs were evaluated as binary and ternary detectors. Their study suggested that an LLM generally performs better at identifying texts generated by itself compared with recognising texts generated by other LLMs. Slightly differently, Leinonen & Denny (2026) examined self-detection by a single LLM, and found that self-detection varies substantially across task, response length, and prompt formulation. Both studies have limited scopes and their results need more independent validation with more recent LLMs, to enhance our understanding on self- and cross-detection of LGTs. More specifically, neither studied generational effects, e.g., how LLMs of different generations behave as self- and cross-detector of LGTs.

To fill the above-mentioned research gaps, we conducted a large-scale study of LGT detection using 15 general-purpose LLMs of three generations (i.e., *G1: 2023-24*, *G2: 2024-25*, and *G3: 2026*) as both text generators and detectors. Following Ji et al. (2025)'s approach, we selected 1,000 human-generated texts (HGTs) from the publicly available M4GT-Bench dataset (Wang et al., 2024). Next, each of the 15 selected LLMs was used to generate 1,000 LGTs. The task is to use each selected LLM as a detector to perform binary classification task on the 15,000 LGTs and 1,000 HGTs and to provide a natural-language explanation for each of its decisions. Our work led to the following main findings and contributions:

1. **A controlled and openly accessible benchmark.** We constructed a corpus containing 1,000 HGTs and 15,000 LGTs produced by 15 general-purpose LLMs. Applying every model as a detector results in a complete matrix of 225 generator–detector combinations and 233,428 valid binary judgements with natural-language explanations.

2. **A cross-generational evaluation of LLM detectors.** We defined and introduced three LLM generations (based on recency) and separately examined how generator and detector generations affect detection performance. Detection performance improves substantially over time, and detector generation is more strongly associated with performance than generator generation, although texts generated G3 generators are generally harder to detect, reflecting the more improved capabilities of more recent LLMs.

3. **A more comprehensive study of self- and cross-detection.** Across all generator and detector generations, there are only slight differences in average performances between self- and cross-detection, while the direction of the difference varies across different detectors. We also conducted a controlled comparison between each detector's self-detection and cross-detection performance from the same generation, using prompt-clustered bootstrap tests, Holm correction, and sensitivity analyses. We have observed both significant self-detection advantages and disadvantages for different detectors within the same generation, however, their effects are generally small and do not support a consistent and conclusive self-detection advantage or disadvantage across LLMs. These results challenge the results shown in (Ji et al., 2025).

4. **A multi-perspective analysis of error patterns and explanations.** We analysed error patterns from three perspectives: false-positive rate (FPR) and false-negative rate (FNR) analysis, an LLM-assisted analysis of explanations, and a literature-guided cue-based analysis. The first analysis revealed distinct generational patterns: G1 detectors show a strong "everything is human-written" tendency, G2 detectors more frequently predict machine, and most G3 detectors maintain a low FPR and a low FNR. The two subsequent analyses of semantic error patterns converge in showing that *structured and generic writing* is commonly associated with LGTs, whereas *speci-*

ficity can lead to a LGT being misclassified as a HGT, and *coherence* is associated with both HGT and LGT judgements.

2 Related Work

LLM-generated text detection: Prior work developed supervised, training-free, rewriting-based, and watermarking approaches to LGT detection as a binary classification task (Wu et al., 2025a; Mitchell et al., 2023; Hans et al., 2024; Huang et al., 2025). More recent methods also use linguistic, semantic, and structural features Wu et al. (2025b); Guo et al. (2026); Xiong et al. (2026); Wu et al. (2026). Although these methods can perform well under matched conditions, their reliability often decreases across unfamiliar generators, domains, decoding settings, and prompt styles (Wang et al., 2024; Dugan et al., 2024; Tufts et al., 2025; Layton et al., 2026).

LLMs as detectors and self-detection: General-purpose LLMs can also be used as zero-shot detectors that detect LGT and HGT with a natural-language explanation. However, their predictions may be strongly biased towards one class (Bhattacharjee & Liu, 2024). Ji et al. (2025) reported a self-detection advantage across six LLMs, but subsequent evidence from other research produced mixed results, with self-recognition varying across models, evaluation settings, task domains, prompts, response lengths, and writing styles (Panickssery et al., 2024; Bai et al., 2025; Caiado & Hahsler, 2025; St. Amand et al., 2026; Leinonen & Denny, 2026).

Error analysis and explanations: Natural-language explanations can reveal the cues used by LLM detectors, but a correct label may still be supported by weak, absent, or incorrectly interpreted evidence (Ji et al., 2025). Human annotators similarly rely on cues such as vocabulary, structure, grammar, and originality, although the same cue may support different judgement (Russell et al., 2025). Research on commercial detector can provide only limited insight because many reported cues overlap between HGTs and LGTs (Ji et al., 2024).

Research gaps Systematic evaluations of general-purpose LLMs acting as both generators and detectors remain limited in scale, leaving self- and cross-detection across broader model generations insufficiently understood (Ji et al., 2025). It also remains unclear how the cues reported by LLM detectors relate to correct decisions, false positives, and false negatives across these settings. A more detailed review of the individual methods and findings is provided in Appendix A.1.

3 Experimental design

Model generations and selections. We grouped models into three generations largely based on time of release: G1 (2023 and early 2024), G2 (2024-25), G3 (2026). G1 covers early-stage LLMs designed primarily to follow input prompts and output direct responses. G2 covers more powerful general-purpose LLMs with improved instruction following, coding, multilingual, long-context, and general reasoning capabilities, but with minor inference-time thinking or reasoning. G3 cover the most recent models (at the time of our experiments) that are designed for inference-time reasoning with extended thinking, tool use, and agentic execution capabilities.

The model selection for each generation follows the same rule. They must support general-purpose text generation and judgement, provide diversity across vendors, and collectively include both proprietary and open-weight models. It is worth noting that we mainly accessed the models via OpenRouter and vendors' native APIs. Since the fast development of LLMs in general, many early candidate models have been deprecated or discontinued. Hence, model availability constrains the final model selection, resulting in fewer G1 models and unbalanced group sizes across generations. The final selection is summarised in Table 1.

Corpus construction. Following (Ji et al., 2025), we directly reused their HGT dataset and the corresponding text generation prompts reported in their study to populate the LGT dataset. Specifically, the HGT dataset contains 1,000 HGTs selected from the publicly available M4GT-Bench dataset (Wang et al., 2024), covering a broad range of topics, styles, and formats. Based on these HGTs, 1,000 matching text-generation prompts were created to align with the themes, structures, and styles appearing in the 1,000 HGTs. See Listing 1 in Appendix A.2.1 for an example prompt and refer to (Ji et al., 2025) for more details. Then, we applied the 1,000 text-generation prompts

Table 1: Selected models across generations. Every model acts as both a text generator and a detector. Models with multimodal interfaces are evaluated using text only.

Generation	Selected models
G1	GPT-3.5 Turbo Instruct (OpenAI, 2023); Mixtral 8×22B Instruct (Mistral AI, 2024).
G2	GPT-4o (OpenAI, 2024); Qwen2.5-72B Instruct (Qwen Team, 2024); DeepSeek-V3 (DeepSeek-AI, 2024); Llama 3.3 70B Instruct (Meta AI, 2024); GLM-4 32B 0414 (Z.AI, 2025).
G3	DeepSeek-V4-Pro (DeepSeek-AI, 2026); Qwen3.6-Max-Preview (Qwen Team, 2026); GLM-5.1 (Z.AI, 2026); MiMo-V2.5-Pro (Xiaomi MiMo Team, 2026); GPT-5.5 (OpenAI, 2026); Claude Opus 4.7 (Anthropic, 2026a); Gemini 3.1 Pro Preview (Google, 2026); Kimi K2.6 (Moonshot AI, 2026).

to each of the 15 selected LLMs, with each model generating one LGT for each prompt. See Appendix A.2.1for a example LGT generated by GPT-5.5 for an example prompt.

For text generation, models were accessed through OpenRouter (2026) using an OpenAI-compatible chat-completions interface. Each request contains the matched user prompt and a fixed system instruction requiring plain, continuous prose without headings, lists, or Markdown formatting. The user prompts (i.e., text-generation prompts) specify the topic, task, expected form, and a target length, which typically contains 150–300 words. In addition, we reviewed the documented default temperature and top-p values for all selected models, as reported in Table 6 in Appendix A.2.1. A temperature of 1.0 and top-p of 1.0 or close to 1.0, such as 0.95, are the most commonly used values. Hence, we set both temperature and top-p to 1.0 as common experimental settings to ensure comparable generation conditions across models. In total, with 15 selected LLM models as text generators, we produced a corpus that contains of 1,000 HGTs and 15,000 (i.e., 15 generators $\times$ $1,000$ prompts) LGTs, comprising 16,000 texts in total. The corpus can be accessed from `https://anonymous.4open.science/r/llm-self-cross-detection-C481`.

Binary classification on LGTs and HGTs. After constructing the corpus, we used the selected LLMs as detectors to perform a binary classification task to judge whether a given text was written by a human or generated by an LLM. For each generator, we combined its 1,000 generated LGTs with the same set of 1,000 HGTs to construct a generator-specific dataset. Hence, we have 15 generator-specific datasets, each containing 2,000 texts. Each detector then performs the binary classification task on all 15 generator-specific datasets. In total, this process generates 240,000 binary classification results (i.e., 15 detectors $\times$ 15 datasets $\times$ 1000 LGTs + 15 detectors $\times$ 1000 HGTs). When the detector and generator are the same model, the evaluation is considered self-detection; otherwise, it is considered cross-detection.

For the binary classification task, the same zero-shot instruction was used for every detector, where the instruction explicitly asks the detector to provide both a binary judgement with a natural-language explanations to support its decision. For each request made by each detector, the text to be judged is appended to this instruction, with no additional system prompt. The required JSON response only contains two fields. The `answer` field must contain either `human` or `machine`, while the `explanation` field records the corresponding detector's explanations in natural language. See Appendix A.2.2 for an example detection result generated by GPT-5.5.

Evaluation metrics. Treating LGT as the positive class, F1-score and accuracy are the primary metrics used to evaluate classification performance. FPR and FNR are also used to further interpret the results and identify class-specific error patterns. Of the 240,000 classification responses, 233,428 (97.3%) produced valid binary judgements after excluding 1,909 missing responses and 4,663 malformed responses (more details in Appendix A.3.2).

4 Results

4.1 Overall performance

Table 2 shows detection performance in a 15×15 generator–detector matrix. Each generator represents its corresponding generator-specific dataset and has one row for F1-score and one row for accuracy. Red and blue values denote F1-score and accuracy for self-detection, respectively. For each detector column, the highest F1-score is shown in bold with a light-red background, while

the highest accuracy is shown in bold with a light-blue background. Tied maximum values are all highlighted. Horizontal rules separate G1, G2, and G3 generators, while vertical rules separate the three detector generations. For simplicity and readability, we use DS to denote DeepSeek, Claude to denote Claude Opus 4.7, and MiMo to denote MiMo V2.5 Pro in the table. Across the matrix, the highest column-wise values are primarily achieved by G3 detectors, particularly on texts generated by G1 and G2 models as illustrated in the table. The mean F1-score is 69.5% and the mean accuracy is 76.5%. However, both metrics vary substantially, with standard deviations of 31.7 and 19.6 percentage points, respectively. To this end, the overall averages of F1-score and accuracy indicate the considerable variation across model pairings rather than overall strong detection performance.

Table 2: Complete classification results for all generator–detector pairings (%).

Generation	Generator	Metric	GPT-3.5	Mixtral	DS-V3	GLM-4	GPT-4o	Llama-3.3	Qwen2	Claude	DS-V4	Gemini-3.1	GLM-5.1	GPT-5.5	Kimi-K2.6	MiMo-V2.5	Qwen3.6
G1	GPT-3.5	F1	3.0	42.6	62.7	59.1	80.6	**77.1**	1.7	97.0	93.1	**99.9**	84.7	**99.0**	99.3	91.9	99.0
		Accuracy	47.4	56.5	60.6	61.2	76.6	**71.1**	48.3	96.9	93.0	**99.9**	86.5	**99.0**	99.3	91.9	99.0
	Mixtral	F1	2.3	38.6	62.0	51.3	75.4	70.0	0.6	96.8	82.9	99.7	80.2	98.9	99.2	86.1	99.1
		Accuracy	47.3	54.7	60.1	56.2	71.6	64.2	47.9	96.7	84.1	99.7	83.2	98.9	99.2	86.8	99.1
G2	DS-V3	F1	2.4	39.5	51.9	53.8	76.5	73.0	1	97.0	88.1	99.8	84.5	99.0	99.2	88.3	**99.2**
		Accuracy	47.4	55.0	52.9	57.8	72.6	67.0	48.0	96.9	88.4	99.8	86.4	98.9	99.2	88.7	**99.2**
	GLM-4	F1	3.4	38.0	56.1	53.9	75.6	71.8	0.6	96.9	85.3	99.7	83.1	98.9	99.0	84.9	98.9
		Accuracy	47.5	54.4	55.7	57.9	71.8	65.8	47.9	96.8	86.1	99.7	85.3	98.9	99.0	85.9	98.9
	GPT-4o	F1	1.7	40.6	60.7	55.3	77.3	73.4	0.4	97.0	89.7	99.8	**89.1**	98.9	99.2	89.9	99.2
		Accuracy	47.0	55.6	59.1	58.7	73.4	67.4	47.9	96.9	89.9	99.8	**90.1**	98.9	99.2	90.1	99.2
	Llama-3.3	F1	3.0	47.9	**71.1**	60.5	**80.8**	76.8	4.7	97.0	**93.4**	99.8	85.2	99.0	99.4	**92.1**	**99.2**
		Accuracy	47.4	59.2	**67.5**	62.2	**76.8**	70.8	49.0	96.9	**93.2**	99.9	86.9	98.9	99.4	**92.2**	**99.2**
	Qwen2	F1	4.8	44.1	64.5	58.9	80.3	76.9	1.7	**97.0**	91.5	**99.9**	87.2	99.0	99.4	92.0	99.1
		Accuracy	47.9	57.3	62.0	61.1	76.3	70.9	48.2	**96.9**	91.5	**99.9**	88.5	98.9	99.4	92.0	99.1
G3	Claude	F1	2.6	31.6	31.4	44.4	66.6	57.7	0.6	96.7	77.6	98.9	78.0	98.8	98.6	79.8	97.7
		Accuracy	47.4	51.6	40.9	52.3	63.9	53.8	47.9	96.6	80.2	98.9	81.7	98.8	98.6	81.9	97.8
	DS-V4	F1	5.0	38.8	57.2	46.7	71.6	60.3	1.1	95.7	76.2	99.3	77.8	98.7	98.8	79.3	97.2
		Accuracy	48.0	54.8	56.5	53.6	68.2	55.9	48.1	95.6	79.1	99.3	81.6	98.7	98.8	81.5	97.3
	Gemini-3.1	F1	9.0	**63.4**	61.1	52.0	73.3	73.3	3.9	95.7	76.2	93.9	84.4	98.7	97.3	66.3	94.2
		Accuracy	49.1	**68.1**	59.4	56.7	69.7	67.3	48.9	95.6	79.2	94.2	86.3	98.8	97.3	72.8	94.5
	GLM-5.1	F1	11.3	59.0	51.5	**69.0**	79.7	76.0	**14.9**	96.2	70.4	98.6	72.7	98.3	98.5	73.8	97.2
		Accuracy	49.9	65.3	52.6	**68.4**	75.7	69.9	**52.0**	96.1	75.2	98.6	78.2	98.4	98.5	77.6	97.3
	GPT-5.5	F1	1.7	31.4	58.9	50.0	72.4	66.6	0.4	96.9	76.9	99.7	82.0	98.9	98.5	82.3	98.2
		Accuracy	47.1	51.5	57.7	55.5	68.8	61.2	47.9	96.8	79.6	99.7	84.6	98.8	98.5	83.8	98.3
	Kimi-K2.6	F1	**15.2**	39.9	53.9	49.9	78.9	58.8	4.9	92.4	68.1	99.4	85.5	98.9	98.9	71.6	96.2
		Accuracy	**50.9**	55.3	54.2	55.3	74.9	54.7	49.1	92.5	73.9	99.4	87.2	98.9	98.9	76.1	96.3
	MiMo-V2.5	F1	3.4	36.1	34.0	47.3	69.3	65.0	0.8	96.8	81.1	99.2	82.6	98.9	98.9	84.6	98.7
		Accuracy	47.5	53.5	42.3	53.9	66.1	59.7	48.0	96.7	82.7	99.2	85.0	98.9	98.9	85.6	98.7
	Qwen3.6	F1	3.9	36.4	64.3	50.5	73.9	67.5	0.2	96.9	80.7	99.6	77.3	98.9	**99.4**	83.0	99.0
		Accuracy	47.7	53.6	61.8	55.8	70.2	62.0	47.9	96.8	82.6	99.6	81.3	98.9	**99.4**	84.3	99.0

To further examine the variation, we also looked at the mean performance of each detector across 15 generator-specific dataset (see Table 7 Panel A in Appendix A.3.1), as well as the mean performance of all detectors on each generator-specific dataset (see Table 7 Panel B in Appendix A.3.1). Gemini 3.1 Pro Preview exhibits the highest performance across all generator-specific datasets, with a mean F1-score of 99.2% and mean accuracy of 99.2%, whereas Qwen2-72B has a mean F1-score of 2.5% and mean accuracy of 48.5%. In addition, across all detectors, texts generated by Llama 3.3 70B are the easiest to classify across all detectors, with a mean F1-score of 74% and mean accuracy of 80%, whereas texts generated by Claude Opus 4.7 are the most difficult to classify, with a mean F1-score of 64.1% and mean accuracy of 72.8%.

4.2 Cross-generation analysis

As shown in Table 3a, detection performance improves substantially from G1 to G3 detectors across all generator-specific datasets. On average, the F1-score improves from 22.8% to 93.1% and the accuracy increases from 52.0% to 93.7% from G1 to G3. In particular, G1 detectors achieved low F1-score despite the accuracy slightly over 50.0%, suggesting that G1 detectors frequently failed to identify LGTs rather than making balanced predictions across two classes.

When comparing generations of generators, we observed a more subtle but still visible pattern. G2 detectors perform similarly on texts generated by G1 and G2 models, but their F1-scores decrease from approximately 54% to 49% and their accuracy values decrease from approximately 62% to 58% on G3-generated texts. G3 detectors show a similar pattern, achieving better performance

Table 3: Detection performance across model generations (%).

(a) Mean classification performance for each pairing of generator and detector generations

Generator Generation	Metric	Detector Generation G1	G2	G3
G1	F1	21.6	54.1	94.2
	Accuracy	51.5	61.8	94.6
G2	F1	22.5	54.3	94.9
	Accuracy	51.9	61.9	95.1
G3	F1	24.3	49.0	90.3
	Accuracy	52.6	57.7	91.5
Macro average	F1	22.8	52.5	93.1
	Accuracy	52.0	60.5	93.7

(b) Self- and cross-detection performance across model generations

Setting	Generator Generation	Metric	Detector and Generator Generation G1	G2	G3
Self	Same	F1	20.8	52.3	90.1
		Accuracy	51.1	60.6	91.3
Cross-Detection by Generator Generation			Detector Generation G1	G2	G3
Cross	G1	F1	22.4	54.1	94.2
		Accuracy	51.9	61.8	94.6
	G2	F1	22.5	54.8	94.9
		Accuracy	51.9	62.3	95.1
	G3	F1	24.3	49.0	90.4
		Accuracy	52.6	57.7	91.5

on texts generated by G1 and G2 models than on those generated by G3 models. In contrast, G1 detectors consistently perform poorly on texts produced by all three generations of generators. These results suggest that detection performance is more strongly associated with detector generation than with generator generation. Nevertheless, texts produced by G3 models appear more difficult to classify, even for the more capable G2 and G3 detectors.

4.3 Self-detection vs. cross-detection

To investigate whether LLMs can detect their own generated texts better than texts generated by other models, we compared self-detection with cross-detection. For a fair comparison, we compared each detector's performance on its own outputs with its performance on outputs generated by other models from the same generation. This controls for differences in detection difficulty across generator generations. Thereby, we averaged self- and cross-detection scores presented in Table 2 by generations and report them in Table 3b.

For readability, we use Gx–Gy cross-detection to represent detectors from Gx evaluating texts generated by Gy models. For example, G1–G2 represents G1 detectors evaluating texts generated by G2 models. For G1 detectors, self-detection achieves an F1-score of 20.8% and an accuracy value of 51.1%, compared with 22.4% and 51.9% for matched G1–G1 cross-detection. A similar pattern appears for G2 detectors comparing with the matched G2–G2 cross-detection, where self-detection reaches 52.3% in F1 and 60.6% in accuracy, below the corresponding cross-detection results of 54.8% and 62.3%. For G3 detectors, the difference becomes much smaller. Taken together, the generation-level averages do not suggest that LLMs generally detect their own outputs more accurately than outputs generated by other models from the same generation. As shown in Table 2, we can observe further differences between individual models. For column-wise comparison, none of the 15 detectors achieves its highest F1-score or accuracy on its own generated texts. Nevertheless, nine detectors achieve higher self-detection F1-score and accuracy than their respective averages on texts generated by other models from the same generation (i.e., GPT-3.5 Turbo Instruct from G1; Llama 3.3 70B and Qwen2-72B from G2; Claude Opus 4.7, DeepSeek-V4-Pro, GPT-5.5, Kimi K2.6, MiMo V2.5 Pro, and Qwen3.6 Max Preview from G3). This is further summarised in Table 4, where Δ represents the self-detection score minus the detector's average score on texts generated by other models from the same generation. Hence, positive values indicate

Table 4: Model-level differences between self- and matched cross-detection (percentage points).

Generation	Detector	ΔF1 (pp)	ΔAccuracy (pp)
G1	GPT-3.5 Turbo Instruct	$+0.76$	$+0.18$
	Mixtral 8x22B	-4.00	-1.85
G2	**DeepSeek-V3**	$\mathbf{-11.15}$	$\mathbf{-8.15}$
	GLM-4-32B	-3.17	-2.06
	GPT-4o	-0.98	-0.99
	Llama 3.3 70B	$\mathbf{+2.99}$	$\mathbf{+2.96}$
	Qwen2-72B	$+0.06$	$+0.01$
G3	**Claude Opus 4.7**	$\mathbf{+0.89}$	$\mathbf{+0.86}$
	DeepSeek-V4-Pro	$+0.30$	$+0.05$
	Gemini 3.1 Pro Preview	$\mathbf{-5.40}$	$\mathbf{-5.06}$
	GLM-5.1	$\mathbf{-8.43}$	$\mathbf{-5.76}$
	GPT-5.5	$+0.10$	$+0.10$
	Kimi K2.6	$+0.36$	$+0.35$
	MiMo V2.5 Pro	$\mathbf{+8.00}$	$\mathbf{+5.90}$
	Qwen3.6 Max Preview	$\mathbf{+1.93}$	$\mathbf{+1.86}$

higher self-detection performance, whereas negative values suggest higher matched cross-detection performance.

To test the statistical significance of differences observed in Table 4, we applied prompt-clustered bootstrap tests and identify statistically significant differences for seven detectors after Holm correction (i.e., see detectors in bold in Table 4). Four detectors (i.e., Llama 3.3 70B, Claude Opus 4.7, MiMo V2.5 Pro, and Qwen3.6 Max Preview) perform significantly better on texts generated by themselves compared with the texts generated by other models within the same generation. In contrast, DeepSeek-V3, Gemini 3.1 Pro Preview, and GLM-5.1 exhibit significant worse self-detection performance than cross-detection within the same generation. To provide a more comprehensive statistical assessment, we also aggregated the detector-level results within each generation, and conducted two sensitivity analyses that compare self-detection with cross-detection across all generations (more details can be found in Appendix A.3.2). Taking all results together, we did not find evidence that support a consistent self-detection advantage or disadvantage shared across LLMs. These results challenge what Ji et al. (2025) reported for smaller-scale experiments, suggesting that their results are likely accidental or contain errors.

4.4 Error pattern analysis

4.4.1 False-positive and false-negative patterns

To further interpret results in Table 2, we examined FPRs and FNRs. A false negative occurs when an LGT is classified as human-written, whereas a false positive refers when an HGT is classified as generated by an LLM. Table 5 presents the mean error rates across model generations. The results show three general patterns: 1) G1 detectors have consistently high FNRs, but a much lower FPR of 13.0%. This suggests that G1 detectors frequently classify LGTs as human-written, exhibiting a conservative "everything is human" tendency. 2) G2 detectors reduce the FNR to 39.6–48.1%, but their FPR increases to 36.5%, suggesting that they make both type of errors relatively frequently. 3) Most G3 detectors have stronger overall performance with more balanced error profile across different generator generations, although their FNR rises to 13.4% on G3-generated texts. This observation further confirms that G3-generated texts are comparatively more difficult to detect. More detector-level examples (e.g., GPT-3.5 Turbo Instruct's human-label tendency and GPT-4o's comparatively higher FPR) showing individual error profiles are reported in Figure 1 and Appendix A.3.3. Overall, detector generation is more associated with the type and magnitude of classification errors, while generator generation mainly influences the rate of misclassification of treating LGTs as human-written.

Table 5: Error rates across model generations (%).

	FNR by Generator Generation			FPR on HGTs
Detector Generation	G1	G2	G3	Shared Corpus
G1	84.1	83.3	81.9	13.0
G2	39.9	39.6	48.1	36.5
G3	7.2	6.1	13.4	3.6

4.4.2 Semantic error pattern analysis

We further analysed the natural-language explanations using an LLM-assisted analysis and a literature-guided cue-based analysis. The LLM-assisted analysis was conducted from two complementary directions. *Direction 1* keeps the exact text constant and compares explanations across different detectors. Its aim is to discover how and why different detectors make wrong judgements given identical texts. *Direction 2* holds the detector and the text generation prompt (i.e., topic and rules) constant and compares explanations for texts produced by different generators. It investigates whether the same detector changes its judgement or reasoning when evaluating texts generated by different LLMs on the same topic under the same rules. Then the literature-guided cue-based analysis was conducted as a complementary and reproducible check of whether recurring patterns identified by the LLM-assisted analysis can also be observed.

We sampled 20 text generation prompts and selected two generators and two detectors from each generation, producing 140 texts and 840 detector judgements. More details such as the sampling procedure and prompts are reported in Appendix A.3.4.

LLM-assisted semantic error pattern analysis We used a modern LLM (GPT-5.6 Sol with medium reasoning capability) to perform the task. The LLM received the prepared dataset, including texts, predicted labels, and explanations side by side. It was instructed to distinguish between describing a textual feature and making a valid inference about the classification task, and to support each qualitative finding with identifiable records. The complete prompts for analysing Direction 1 and Direction 2 are reported in Listings 5 and 6 in Appendix A.3.5, respectively.

In Direction 1, a typical example occurs for a GPT-3.5 generated text about medical malpractice, where DeepSeek-V3 treated the details and coherence appearing in the text as evidence of HGT, whereas DeepSeek-V4-Pro interpreted such characteristics as predictable transitions and uniform tone for evidence as LGT. In Direction 2, GPT-4o labeled empathetic and conversational responses to text generated using an academic-pressure prompt as HGT, but labeled more regularly structured responses to the text generated using the same prompt as MGT. These examples show that detectors can interpret similar properties differently and can change the reasoning across texts on the same topic. Further examples are reported in Tables 12 13 in Appendix A.3.5 and more detailed analyses are also reported in Appendix A.3.5.

Cue-based semantic error pattern analysis To complement the LLM-assisted analysis, we conducted a literature-guided cue-based analysis to examine whether the recurring patterns reported in the LLM-assisted analysis can also be discovered via the exploratory and reproducible cue-based analysis, following prior studies that perform similar explanation centric analyses (Ji et al., 2025; Russell et al., 2025). To define the cue categories, we conducted a focused review of existing studies that investigate textual characteristics and reasoning cues for distinguishing HGTs from LGTs. Based on the small-scale review, we identified eight recurring cue categories, including *structure and formatting* (SF), *fluency and polish* (FP), *generic or abstract language* (GAL), *hedging and uncertainty* (HC), *personal voice* (PV), *coherence and transitions* (CT), *specificity and evidence* (SE), and *factuality and hallucination* (FH). The complete taxonomy and lexical patterns are reported in Tables 14 and 15 in Appendix A.3.6, the detailed experiment procedure is also reported in Appendix A.3.6. It is worth noting that these categories are derived from a focused rather than systematic literature review and should therefore be treated as literature-guided analytical categories, rather than definitive taxonomy of classification cues.

We have observed four cue categories that are particularly consistent with the recurring findings from the LLM-assisted analysis. Since the two analyses converge on these categories, we computed how frequently each cue is mentioned in true-positive and false-negative explanations. The results are reported in Table 16 in Appendix A.3.6, where each value represents the percentage of explanations containing at least one predefined term associated with the corresponding category. Therefore, the percentages measure how frequently detectors refer to each cue in their explanations, rather than how frequently the corresponding textual property occurs in the evaluated texts. SF captures references to regular or formulaic organisation, while GAL captures references to broad and insufficiently distinctive coverage. Detectors commonly used these two to support LGT judgements. For instance, SF appear in 91.4% of GPT-4o's true-positive explanations but only 33.3% of its false-negative ones. In contrast, SE captures the use of detailed examples, technical knowledge, or apparent expertise to support HGT judgements. This cue is more frequent in false-negative than true-positive explanations for Mixtral, GPT-4O, DeepSeek-V3, and DeepSeek-V4-Pro. For instance, DeepSeek-V3 refers to "specific examples" and a "detailed and structured approach" as reasoning evidence to incorrectly classify a LGT generated by GPT-3.5 as human-written. Finally, CT corresponds to the cue-inversion pattern observed in the LLM-assisted analysis, where the same clear and logically organised structure can lead different detectors to assign opposite labels. This appears in 83.3% of Mixtral's true-positive explanations and 85.6% of its false-negative explanations. Overall, the cue-based results provide convergent evidence for several patterns identified by the LLM-assisted analysis, while also showing that the same cue does not consistently lead to the same classification decision.

5 Further Discussions, Limitations and Future Work

Revisiting the self-detection advantage Ji et al. (2025) reported a clear self-detection advantage across the six LLMs evaluated in their study. In comparison, our results covering more (15) models do not show such a consistent advantage. However, our further analysis on comparing them within

the same generation found evidence that self-detection advantages exist for some LLM models. Nevertheless, self-detection advantage appears to be model- and comparison-dependent rather than a general pattern shared by all LLMs. Therefore, our study enriches and corrects results in (Ji et al., 2025) due to our larger coverage models, clear separation of generator and detector generations, introducing matched comparisons, and adding statistical tests. *Future work should systematically examine how different factors, such as model selection, prompts, affect self-detection to identify their individual and interaction effects*.

Watermarking and self-detection Several LLM vendors have announced their own watermarking mechanism to make their generated content more reliably recognisable by utilising specific detection procedures and secret keys Dathathri et al. (2024); Google DeepMind (2024); Anthropic (2026b). We conducted an exploratory review to examine whether such mechanisms are documented for the selected models in this study. As summarised in Table 17 in Appendix A.4, most of the selected LLMs in this study do not have watermarking enabled at the time of our experiments. Even when a provider has announced watermarking, it is not always confirmed that the specific model/endpoint used in this study applied the watermark. Moreover, models showing a self-detection advantage do not consistently associate with those models which support watermarking. This might suggest that watermarking alone cannot explain the mixed self-detection results. *Future work should be conducted to test whether watermarking improves self-detection by comparing confirmed watermarked and un-watermarked outputs from the same model and endpoint using the corresponding official detector or key*.

Implications of cross-generational evaluation Our cross-generational analysis offers a new perspective to evaluate binary classification performance by considering the generation of both generator and detector. Our findings suggest that detector generation is more strongly associated with performance than generator generation, although texts generated by G3 LLMs appears to be more difficult for all detectors. However, it is important to acknowledge that the G1–G3 generation definition we used is rather simplistic, so does not necessarily reflect the most appropriate classification of LLMs. *Future work should conduct a systematic literature review to develop and validate a more comprehensive taxonomy of LLM generations, supported by broader evidence on model capabilities, post-training methods, interaction paradigms, and technological development*.

Interpreting error patterns and explanations We analysed the error patterns from three perspectives. The analysis of FPRs and FNRs revealed distinct generational tendencies, whereas the LLM-assisted and cue-based analyses showed consistent findings on several semantic patterns. Although the cue-based analysis can verify whether predefined cues are mentioned in an explanation, it cannot determine whether the corresponding textual properties or the explanations provide valid evidence of classification. The LLM-assisted analysis compares explanations with the evaluated texts more directly, but its interpretation may be influenced by the model (i.e., GPT 5.6 Sol)'s own understanding of HGTs and LGTs. Moreover, detector explanations may be post-hoc justifications rather than accurate and reliable descriptions of how a decision was made (Ji et al., 2025). *Future work should therefore involve recruitment of human annotators to confirm the the findings presented in this study, report inter-annotator agreement, and compare human coding with the two automated analyses*.

6 Conclusion

In this paper, we present a study of evaluating 15 general-purpose LLMs from three self-defined generations as both text generators and detectors. We instructed LLMs to perform binary classification task on a corpus of 1,000 HGTs and 15,000 LGTs. We compared detection performance and analyse LLM-generated explanations from different perspectives. Our findings suggest that, despite that texts generated by G3 generators are comparatively difficult to detect, detection performance is more strongly associated with detector generation than generator generation. The comparison between self- and cross-detection has mixed and model-specific results, providing no conclusive evidence of a consistent self-detection advantage or disadvantage. The investigation of error pattern combining FPR and FNR analysis, LLM-assisted analysis, and cue-based analysis, identify different generational tendencies and show that detectors can interpret the same textual cues differently. Overall, general-purpose LLMs show great potential as detectors, but their performance and explanations remain model-dependent, and the LLM-generated explanations should be used with caution as they may contain errors.

AI USE STATEMENT

We used generative AI tools to proofread the manuscript and check consistency in the use of terminology. Some data-processing and analysis code was co-generated with GitHub Copilot, and all AI-assisted code was reviewed, tested, and verified for correctness by us. We did not use generative AI to generate research ideas or draft any scientific content of the paper. The LLM-assisted semantic error pattern analysis was conducted as part of the research methodology and it has been explicitly described in Section 4.4.2. The authors take full responsibility for the final content, claims, and artifacts presented in this work.

ETHICS STATEMENT

This study does not involve human participants or any private personal data. All experiments were conducted using publicly available LLMs and datasets, and by ourselves as researchers.

REPRODUCIBILITY STATEMENT

To support reproducibility, we release an anonymous repository containing the complete HGT and LGT corpus, raw detector judgements with explanations, text generation and detection prompts, source code to perform the text generation task and binary classification task, source code to produce all results and conduct all analyses presented in the manuscript. The anonymous repository can be accessed from `https://anonymous.4open.science/r/llm-self-cross-detection-C481`. Section 3 presents the detailed experiment design while Appendix A provides all necessary materials to support further explanations of experiment design.

REFERENCES

Anthropic. Introducing Claude Opus 4.7. Anthropic, 2026a. URL `https://www.anthropic.com/news/claude-opus-4-7`.

Anthropic. How Claude's text watermark works. Anthropic, 2026b. URL `https://www.anthropic.com/news/claude-text-watermark`.

Xiaoyan Bai, Aryan Shrivastava, Ari Holtzman, and Chenhao Tan. Know thyself? on the incapability and implications of AI self-recognition. *arXiv preprint arXiv:2510.03399*, 2025. doi: 10.48550/arXiv.2510.03399.

Amrita Bhattacharjee and Huan Liu. Fighting fire with fire: Can ChatGPT detect AI-generated text? *ACM SIGKDD Explorations Newsletter*, 25(2):14–21, 2024. doi: 10.1145/3655103.3655106.

Antônio Junior Alves Caiado and Michael Hahsler. AI content self-detection for transformer-based large language models. In *Intelligent Systems and Applications*, volume 1554 of *Lecture Notes in Networks and Systems*, pp. 147–168. Springer Nature Switzerland, 2025. doi: 10.1007/978-3-031-99965-9_10.

Elizabeth Clark, Tal August, Sofia Serrano, Nikita Haduong, Suchin Gururangan, and Noah A. Smith. All that's 'human' is not gold: Evaluating human evaluation of generated text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7282–7296. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.565.

Stefan Czybik, Anne Josiane Kouam, Peter Heubl, Jan Magnus Nold, and Konrad Rieck. A large-scale study of personalized phishing using large language models. In *35th USENIX Security Symposium (USENIX Security 26)*. USENIX Association, August 2026. URL `https://www.usenix.org/conference/usenixsecurity26/presentation/czybik`.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, Taylan Cemgil, Zahra

Ahmed, Kitty Stacpoole, Ilia Shumailov, Ciprian Baetu, Sven Gowal, Demis Hassabis, and Pushmeet Kohli. Scalable watermarking for identifying large language model outputs. *Nature*, 634 (8035):818–823, 2024. doi: 10.1038/s41586-024-08025-4.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024. doi: 10.48550/arXiv.2412.19437.

DeepSeek-AI. DeepSeek-V4 preview release. DeepSeek API Documentation, 2026. URL https://api-docs.deepseek.com/news/news260424/.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12463–12492. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.674.

Google. Gemini 3.1 Pro: A smarter model for your most complex tasks. Google, 2026. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/.

Google DeepMind. Watermarking AI-generated text and video with SynthID. Google DeepMind, 2024. URL https://deepmind.google/blog/watermarking-ai-generated-text-and-video-with-synthid/.

Rui Guo, Weibin Zeng, Fuzhang Wu, Yan Kong, Sicheng Shen, Yanjun Wu, and Weiming Dong. HLD: Approximate hierarchical linguistic distribution modeling for LLM-generated text detection. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=l9mqzHROGu.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17519–17537. PMLR, 2024. URL https://proceedings.mlr.press/v235/hans24a.html.

Yifei Huang, Jiuxin Cao, Hanyu Luo, Xin Guan, and Bo Liu. MAGRET: Machine-generated text detection with rewritten texts. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 8336–8346. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-main.557/.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1808–1822. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.164.

Jiazhou Ji, Ruizhe Li, Shujun Li, Jie Guo, Weidong Qiu, Zheng Huang, Chiyu Chen, Xiaoyu Jiang, and Xinru Lu. Detecting machine-generated texts: Not just "AI vs humans" and explainability is complicated. *arXiv preprint arXiv:2406.18259*, 2024. doi: 10.48550/arXiv.2406.18259.

Jiazhou Ji, Jie Guo, Weidong Qiu, Zheng Huang, Yang Xu, Xinru Lu, Xiaoyu Jiang, Ruizhe Li, and Shujun Li. "I Know Myself Better, but Not Really Greatly": How well can LLMs detect and explain LLM-generated texts? *arXiv preprint arXiv:2502.12743*, 2025. doi: 10.48550/arXiv.2502.12743.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):248:1–248:38, 2023. doi: 10.1145/3571730.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Balázs Kovács. The turing test of online reviews: Can we tell the difference between human-written and GPT-4-written online reviews? *Marketing Letters*, 35(4):651–666, 2024. doi: 10.1007/s11002-024-09729-3.

Seth Layton, Bernardo B. P. Medeiros, Kevin Butler, and Patrick Traynor. AI wrote my paper and all i got was this false negative: Measuring the efficacy of commercial AI text detectors. In *2026 IEEE Symposium on Security and Privacy*, pp. 213–232. IEEE, 2026. doi: 10.1109/SP63933.2026.00088.

Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*, WWW '23, pp. 3637–3647. Association for Computing Machinery, 2023. doi: 10.1145/3543507.3583199.

Juho Leinonen and Paul Denny. Distinguishing artificial from authentic: Evaluating LLMs for detecting LLM-generated content. *arXiv preprint arXiv:2607.20446*, 2026. doi: 10.48550/arXiv.2607.20446.

Jiawei Li, Yizhe Yang, Yu Bai, Xiaofeng Zhou, Yinghao Li, Huashan Sun, Yuhang Liu, Xingpeng Si, Yuhao Ye, Yixiao Wu, Yiguan Lin, Bin Xu, Bowen Ren, Chong Feng, Yang Gao, and Heyan Huang. Fundamental capabilities of large language models and their applications in domain scenarios: A survey. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11116–11141. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.599.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. GPT detectors are biased against non-native english writers. *Patterns*, 4(7):100779, 2023. doi: 10.1016/j.patter.2023.100779.

Haitong Luo, Weiyao Zhang, Suhang Wang, Wenji Zou, Chungang Lin, Xuying Meng, and Yujun Zhang. SpecDetect: Simple, fast, and training-free detection of LLM-generated text via spectral analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32356–32364, 2026. doi: 10.1609/aaai.v40i38.40510.

Meta AI. Llama 3.3 70B Instruct model card. Hugging Face Model Card, 2024. URL https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct.

Mistral AI. Cheaper, better, faster, stronger: Mixtral 8x22b. Mistral AI, 2024. URL https://mistral.ai/news/mixtral-8x22b/.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24950–24962. PMLR, 2023. URL https://proceedings.mlr.press/v202/mitchell23a.html.

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. ChatGPT or human? detect and explain: Explaining decisions of machine learning models for detecting short ChatGPT-generated text. *arXiv preprint arXiv:2301.13852*, 2023. doi: 10.48550/arXiv.2301.13852.

Moonshot AI. Kimi K2.6: Leading open-source model in coding and agent tasks. Kimi, 2026. URL https://www.kimi.ai/ai-models/kimi-k2-6.

OpenAI. GPT-3.5 Turbo Instruct model documentation. OpenAI API Documentation, 2023. URL https://developers.openai.com/api/docs/models/gpt-3.5-turbo-instruct.

OpenAI. GPT-4o system card. OpenAI, 2024. URL https://openai.com/index/gpt-4o-system-card/.

OpenAI. Introducing GPT-5.5. OpenAI, 2026. URL https://openai.com/index/introducing-gpt-5-5/.

OpenRouter. Openrouter quickstart guide. OpenRouter Documentation, 2026. URL https://openrouter.ai/docs/quickstart.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2197.

Andrea Pedrotti, Michele Papucci, Cristiano Ciaccio, Alessio Miaschi, Giovanni Puccetti, Felice Dell'Orletta, and Andrea Esuli. Stress-testing machine generated text detection: Shifting language models writing style to fool detectors. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3010–3031. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.156.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115.

Qwen Team. Qwen3.6-Max-Preview. Qwen, 2026. URL https://qwen.ai/blog?id=qwen3.6-max-preview.

Jenna Russell, Marzena Karpinska, and Mohit Iyyer. People who frequently use ChatGPT for writing tasks are accurate and robust detectors of AI-generated text. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5342–5373. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.267.

Jesse St. Amand, Callum Canavan, Sohaib Imran, Joseph Hewson, Aaron Lutz, Shi Feng, Puria Radmard, and Lennie Wells. Self-generated text recognition: Quality heuristics, cross-task transfer, and downstream bias in LLM evaluation. *arXiv preprint arXiv:2608.26159*, 2026. doi: 10.48550/arXiv.2608.26159.

Zhixiong Su, Yichen Wang, Herun Wan, Zhaohan Zhang, and Minnan Luo. HACo-Det: A study towards fine-grained machine-generated text detection under human–AI coauthoring. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22015–22036. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1069.

Yuxia Sun, Ran Zhang, Aoxiang Sun, Xu Li, Zitao Liu, and Jingcai Guo. D&R: Recovery-based AI-generated text detection via a single black-box LLM call. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=FiMZSxo4DO.

Brian Tufts, Xuandong Zhao, and Lei Li. A practical examination of AI-generated text detectors for large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4839–4856. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.271.

Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Maria Bieliková. Disinformation capabilities of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14830–14847. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.793.

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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3964–3992. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.218.

Chenwang Wu, Yiu-ming Cheung, Shuhai Zhang, Bo Han, and Defu Lian. Beyond raw detection scores: Markov-informed calibration for boosting machine-generated text detection. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Lzwkeg2o2z.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338, 2025a. doi: 10.1162/coli_a_00549.

Junxi Wu, Jinpeng Wang, Zheng Liu, Bin Chen, Dongjian Hu, Hao Wu, and Shu-Tao Xia. MoSEs: Uncertainty-aware AI-generated text detection via mixture of stylistics experts with conditional thresholds. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5786–5805. Association for Computational Linguistics, 2025b. doi: 10.18653/v1/2025.emnlp-main.294.

Xiaomi MiMo Team. Xiaomi MiMo-V2.5 series open-sourced. Xiaomi MiMo, 2026. URL https://mimo.mi.com/docs/en-US/news/latest/v2.5-open-sourced.

Ruochong Xiong, Qien Li, Wangwang Lian, Yulong Wan, Hanlin Xue, Zhouxing Tan, Han Yang, Fengyu Lu, and Junfei Liu. Verifiable LLM-generated text detection via projected semantic-structural distributions. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14005–14042. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.638.

Z.AI. GLM-4-32B-0414 model card. Hugging Face Model Card, 2025. URL https://huggingface.co/zai-org/GLM-4-32B-0414.

Z.AI. GLM-5.1: Overview. Z.AI Developer Documentation, 2026. URL https://docs.z.ai/guides/llm/glm-5.1.

Hongyi Zhou, Jin Zhu, Kai Ye, Ying Yang, Erhan Xu, and Chengchun Shi. Learn-to-distance: Distance learning for detecting LLM-generated text. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=2ZUPeEM3FH.

A Appendix

A.1 Supporting materials for Section 2

This section presents a more detailed review on related work.

LLM-generated text detection Prior work conceptualised LGTs detection as a binary classification task to distinguish LGTs from HGTs and has developed various methods to facilitate the detection, including statistics-based detectors, neural-based detectors, and watermarking approaches Wu et al. (2025a). Previous study found that supervised classifiers often perform well when evaluated on texts from the same domains and generators as their training data Wang et al. (2024); Wu et al. (2025a). However, evaluation on M4GT-Bench and RAID show that detectors' performance could vary substantially on unfamiliar data, particularly when the generator, domain, decoding setting changes Wang et al. (2024); Dugan et al. (2024).

Instead, researchers have been exploring various training-free approaches to facilitate LGTs detection. DetectGPT exploits local probability curvature under perturbations to support LGTs detection Mitchell et al. (2023). Binoculars was proposed as a novel LLM detector that compares perplexity estimates using a pair of pre-trained LLMs Hans et al. (2024). More recently, SpecDetect takes the signal processing perspective to detect frequency-domain differences in token log-probability sequences Luo et al. (2026). Rewriting-based approaches use an LLM's response to an input as a provenance signal. MAGRET compares an input with LLM-generated rewrite Huang et al. (2025). D&R analyses the input and measures how well an LLM can recover its semantic and structural properties Sun et al. (2026). Learn-to-Distance learns adaptive metrics between original texts and rewritten texts to help detect LGTs Zhou et al. (2026).

Beyond probability and rewriting based approaches reviewed in above paragraphs, researchers have explored a broader range of textual features such as writing style, token-level dependencies, syntax, and document structure, to facilitate LGTs detection. MoSEs utilises style models and makes detection decision based on an adaptive threshold that is associated with the characteristics and uncertainty of each input Wu et al. (2025b). HLD-Detector compares HGTs and LGTs at multiple linguistic levels in terms of word choice, syntactic structure, and meanings Guo et al. (2026). ProSSD is a statistical framework that learns compact semantic features conditioned on syntactic

structure and can distinguish between HGTs and LGTs using a likelihood-ratio metric that combines Mahalanobis and Wasserstein distances Xiong et al. (2026). Moreover, Wu et al. identified two recurring properties of contextual detection scores: Neighbor Similarity and Initial Instability. Subsequently, Markov-informed score calibration strategy is proposed to model these relationships with a Markov random field and use a lightweight mean-field approximation to calibrate the raw scores, allowing the method to be integrated to existing detectors Wu et al. (2026). Nevertheless, detection reliability remains sensitive to evaluation settings. In a study that conducts broad evaluations, Turfs et al. showed that detector performance varies across unseen dataset, domains, and generators. They also indicated that strong aggregate metrics can obscure weak performance at low false-positive rates Tufts et al. (2025). Similar limitations are also found in a study that investigates commercial systems, Layton et al. reported significant differences in false-positive and false-negative rates across detectors. They show that simple change of styles of prompts, such as using more complex vocabulary, can reduce detection accuracy Layton et al. (2026).

LLMs as detectors and self-detection Complementing the above approaches, another line of research looks at using general-purpose LLMs themselves as detectors, prompting them in a zero-shot setting to classify a given text as either HGT or LGT. Unlike many statistical and likelihood-based methods mentioned above, this approach does not require access to token probabilities, but it can still make a binary HGT–LGT classification decision for a given text as well as providing a natural-language explanation. Bhattacharjee and Liu evaluated ChatGPT as a zero-shot detector and found strongly asymmetric performance between HGTs and LGTs. GPT-3.5 can largely recognise human-written articles but fail to identify most machine-generated articles. In comparison, GPT-4 classifies almost all texts as machine-generated Bhattacharjee & Liu (2024). The study that most closely related to ours is from Ji et al., who systematically evaluated six open- and closed-source LLMs as both generators and detectors under binary and ternary classification settings Ji et al. (2025). Across their experiments, they discovered the self-detection advantage where an LLM generally identifies texts generated by itself more accurately than texts generated by other LLMs (i.e., cross-detection).

There are other studies looking at self-detection and self-recognition of LLMs. Panickssery et al. found that GPT-4 and Llama 2 can identify text generated by themselves among texts generated by other LLMs or humans at above-chance rate. They further suggested that the stronger self-recognition is related to so-called self-preference, meaning that LLM would rate its own generation more favourably than those generated from other models Panickssery et al. (2024). Differently, Bai et al. evaluated ten LLMs and found that self-recognition is rarely above chance. Moreover, the selected models in their study also seems to attribute high-quality text to the GPT and Claude families regardless of its actual source Bai et al. (2025). In a another small scale comparison study, Caido and Hahslef had inconsistent findings, where Bard and ChatGPT exhibit strong self-detection but not Claude Caiado & Hahsler (2025). More recently, using nested sets of 13, 18, or 21 models, depending on their compatibility with each experimental condition, St. Amand et al. showed that the measured self-recognition varies significantly under different contexts, such as evaluation format, conversation structure, and task domain St. Amand et al. (2026). Consistent with this context dependence, Leinonen and Denny found that self-detection performance can be affected by prompt framing, response length, and stylistic instructions in a study that uses GPT-4o as both the text generator and detector Leinonen & Denny (2026).

Error analysis and explanations In addition to instructing LLMs to judge if a given text is a LGT or a HGT, some studies prompt LLM detectors to explain their judgements in natural language, allowing researchers to learn the underlying textual cues and reasoning Ji et al. (2025); Russell et al. (2025). However, based on existing studies, these explanations do not necessarily provide a complete or reliable account of why the detectors made their decisions. Ji et al. examined detector classification results and their explanations collectively and discovered that an LLM can reach the correct decision for the wrong reason. They also identified three recurring issues, including using weak features (e.g., polished grammar) as reliable evidence to distinguish HGTs from LGTs, using features that are not present in the text, and using genuine features but interpreting their relevance incorrectly Ji et al. (2025).

In addition to studies that focus on analysing LLM-generated explanations, Russell et al. collected 1500 free-text explanations from five annotators experienced in using LLMs for writing tasks. They developed a coding scheme for recurring detection cues and applied it with GPT-4o. The analysis

shows that the annotators frequently refer to vocabulary, sentence structure, grammar, and originality as cues to make judgments. By comparing explanations associated with correct and incorrect decisions, they further demonstrated that the same cue can be useful or misleading. For instance, "AI-like" words, such as , *delve* and *crucial*, can contribute to false positives, whereas contractions and colloquial language cause LGTs to be misclassified as human-written Russell et al. (2025). Moreover, researchers have also examined the explanations offered by commercially available detectors. Ji et al., conducted a study to compare the metrics reported by GPTZero with explanations written by human annotators. They reported that cues such as fluency, linguistic errors, repetition, perplexity, and readability can overlap between HGTs and LGTs. In addition, they indicated that the metrics used by GPTZero can only provide limited insights into how GPTZero reached its final decision Ji et al. (2024).

Research gaps Taken together, we have identified a few research gaps that motivated us to carry out this study. Systematic evaluations of general-purpose LLMs acting as both text generators and detectors remain limited in scale. The study most closely related to ours evaluates only six models Ji et al. (2025), leaving it unclear how self- and cross-detection would change across a broader set of models and generations under the same experimental setting. Moreover, prior research has examined LLM explanations, human interpretations, and commercial detector metrics sparsely in separate settings. However, it remains unclear about how the cues reported by general-purpose LLM detectors relate to correct decisions, false positives, and false negatives across self- and cross-detection settings.

A.2 Supporting materials for Section 3

A.2.1 Corpus construction

Listing 1: An example of the matched text generation prompt used to generate LGTs

```
Discuss the properties of gases according to the ideal gas law. How does
temperature, pressure, and volume influence gas behavior? Answer in 150--
300 words.
```

Listing 2: Example of the generated text using GPT-5.5

```
The ideal gas law describes how gases behave when they are assumed to
consist of tiny particles that move freely, have negligible volume, and
do not attract or repel one another. It states that pressure times volume
 is equal to the amount of gas times the gas constant times absolute
temperature. This relationship helps explain how a change in one property
 affects the others when the amount of gas is fixed.
Temperature measures the average kinetic energy of gas particles. When
temperature increases, particles move faster and collide more forcefully
with the walls of their container. If volume is held constant, this
raises the pressure. If pressure is held constant, the gas expands and
its volume increases. Temperature must be measured in kelvins because gas
 behavior depends on absolute temperature.
Pressure is caused by gas particles colliding with surfaces. If a gas is
compressed into a smaller volume while temperature stays the same,
particles collide with the container walls more often, so pressure
increases. This is why pressure and volume are inversely related at
constant temperature.
Volume is the space occupied by the gas. Increasing volume gives
particles more room to move, reducing collision frequency and lowering
pressure if temperature does not change. Although real gases can deviate
from ideal behavior, especially at high pressure or low temperature, the
ideal gas law is a useful model for predicting gas behavior in many
everyday situations.
```

A.2.2 Binary classification

Listing 3: Instruction template used for binary detection.

```
Please determine whether the following text is generated by large
language models or by a human, and provide a detailed explanation for
your decision. You must respond in strict JSON format. The value of "
answer" must be exactly one of: "human" or "machine"---no other values,
capitalisation, or punctuation are allowed. Structure your answer as
follows: {"answer": "human" or "machine", "explanation": "your
explanation here"}
```

Listing 4: Example of self-detection response using GPT-5.5 as both generator and detector


```
"id": 0,
"source": "gpt-5.5",
"ground_truth": "machine",
"response": {"answer":"machine",
"explanation":"The text is highly polished, generic, and evenly
structured in a way that is characteristic of large language model output
. It explains the ideal gas law in broad educational terms without
personal perspective, unusual phrasing, mistakes, or specific contextual
details. The paragraphs follow a predictable pattern: define a concept,
explain the mechanism, then state the consequence. The wording is clear
but somewhat formulaic, with repeated explanatory transitions such as "If
 volume is held constant," "If pressure is held constant," and "This is
why." While a human could certainly write this, the lack of distinctive
style, examples, or variation, combined with the smooth textbook-like
completeness, makes it more likely to be machine-generated."}
```


Table 6: Documented default parameters for selected models

Group	Model	Default temperature	Default top-p	Matches 1.0/1.0
G1	GPT-3.5 Turbo Instruct	1.0	1.0	●
	Mixtral 8×22B Instruct	0.3	1.0	◐
G2	GPT-4o	1.0	1.0	●
	Qwen2.5-72B Instruct	1.0	1.0	●
	DeepSeek-V3	1.0	1.0	●
	Llama 3.3 70B Instruct	1.0	1.0	●
	GLM-4 32B 0414	0.75	0.90	○
G3	DeepSeek-V4-Pro	1.0	1.0	●
	Qwen3.6-Max-Preview	1.0	1.0	●
	GLM-5.1	1.0	0.95	◐
	MiMo-V2.5-Pro	1.0	0.95	◐
	GPT-5.5	Unsupported	Unsupported	**?**
	Claude Opus 4.7	Ignored	Ignored	**?**
	Gemini 3.1 Pro Preview	Not reported	Not reported	**?**
	Kimi K2.6	1.0	1.0	●

Theses defaulted values are verified against OpenRouter or the model provider's documentation. The final column compares the defaults with the common experimental settings of temperature $= 1.0$ and top-$p = 1.0$: ● indicates that both parameters match, ◐ that one matches, ○ that neither matches, and **?** that the comparison is unavailable. The verification was carried out on 3rd September, 2026. For GPT-5.5 and Claude Opus 4.7, these sampling controls are unsupported or ignored; OpenRouter does not publish defaults for Gemini 3.1 Pro Preview.

A.3 SUPPORTING MATERIALS FOR SECTION 4

A.3.1 OVERALL PERFORMANCE

Table 7: Mean performance summaries across generations (%).

Panel A					*Panel B*				
	F1-score		Accuracy			F1-score		Accuracy	
Detector	Mean	SD	Mean	SD	Generator	Mean	SD	Mean	SD
GPT-3.5	4.8	3.9	48.0	1.1	GPT-3.5	72.7	33.3	79.2	19.9
Mixtral	41.9	9.0	56.4	4.6	Mixtral	69.5	33.3	76.6	20.2
DeepSeek-V3	56.1	10.8	56.2	7.1	DeepSeek-V3	70.2	33.7	77.2	20.7
GLM-4	53.5	6.3	57.8	4.1	GLM-4	69.7	33.3	76.8	20.3
GPT-4o	75.5	4.3	71.8	4.0	GPT-4o	71.5	33.8	78.2	20.4
Llama-3.3	69.6	6.7	64.1	6.0	Llama-3.3	**74.0**	32.4	**80.0**	19.2
Qwen2-72B	2.5	3.8	48.5	1.1	Qwen2-72B	73.1	32.9	79.3	19.7
Claude-4.7	96.4	1.2	96.3	1.1	Claude-4.7	64.1	34.6	72.8	22.4
DeepSeek-V4	82.1	8.0	83.9	6.2	DeepSeek-V4	66.9	32.4	74.5	20.4
Gemini-3.1	**99.2**	1.5	**99.2**	1.4	Gemini-3.1	69.5	29.5	75.8	17.8
GLM-5.1	82.3	4.4	84.8	3.1	GLM-5.1	71.1	28.0	76.9	17.7
GPT-5.5	98.9	0.2	98.8	0.2	GPT-5.5	67.7	33.7	75.3	20.6
Kimi-K2.6	98.9	0.5	98.9	0.5	Kimi-K2.6	67.5	30.4	74.5	19.9
MiMo-V2.5	83.1	7.8	84.7	6.0	MiMo-V2.5	66.4	34.4	74.5	22.0
Qwen3.6	98.1	1.4	98.2	1.4	Qwen3.6	68.8	33.0	76.0	20.1

Note: Panel A summarises mean performance of each detector across the 15 generator-specific datasets. Panel B summarises mean performance of all detectors on each generator-specific dataset. SD denotes the sample standard deviation. Bold values indicate the highest mean in each panel.

A.3.2 SELF DETECTION AND CROSS DETECTION STATISTICAL TESTS

Before calculating the evaluation metrics and conducting the statistical tests, we assessed the validity of the classification responses. Of the 240,000 responses, 233,428 (97.3%) contained a valid binary judgement. We considered two types of responses invalid: 1) 1,909 responses (0.8%) were *missing* because the response field was empty; and 2) 4,663 responses (1.9%) were *malformed* because their content could not be parsed into either `human` or `machine`. Both types were excluded rather than treated as classification errors. All reported performance metrics and statistical comparisons were therefore calculated using the available valid responses.

The main statistical test is for testing whether each detector performs differently on its own generated texts and on texts generated by other models from the same generation. We consider two types of responses as invalid. A response is treated as *missing* when its field is empty. A response is treated as *malformed* when its content cannot be parsed into either `human` or `machine`. Both types are excluded rather than treated as classification errors.

For each detector, self-detection accuracy and F1-score are compared with the unweighted average of its scores of cross-detection from the same generation. We computed ΔAccuracy and ΔF1 as difference between the self-detection score and the corresponding cross-detection score. Hence, a positive difference indicates better self-detection performance, while a negative difference indicates better cross-detection performance.

We used a prompt-clustered bootstrap with 10,000 iterations to estimate 95% percentile confidence intervals and two-sided null-centred bootstrap tests. The smallest reportable p-value is therefore $p < 0.0001$. Because a separate hypothesis is tested for each of the 15 detectors, detector-level p-values are adjusted using Holm's procedure to control the family-wise error rate. For model-level tests, a difference (i.e., ΔAccuracy or ΔF1) is considered statistically significant when its Holm-adjusted p-value is under 0.05. For generation-level tests, the difference between self- and

cross-detection are obtained by averaging the detector-level differences within each generation for every bootstrap iteration.

In addition to above, we also aggregated the detector-level results within each generation as reported in Table 8. G2 detectors perform significantly worse in self-detection than in matched same-generation cross-detection, with an accuracy difference of -1.65 percentage points (95% CI $[-2.14, -1.15]$). No generation-level difference has been found for G1 or G3 as their confidence intervals include zero.

Furthermore, two more sensitivity tests are conducted. For the first analysis, we compared each detector's self-detection performance with its performance on texts generated by all 14 other models. We assigned each of the 14 non-self generator models equal weight to compute cross-detection performance. The self-detection accuracy is 0.85 percentage points lower than cross-detection accuracy (95% CI $[-1.12, -0.58]$) on average. For the second analysis, as the three generations contain different numbers of models, we assigned equal total weights to G1, G2, and G3 to calculate cross-detection performance. Under this generation-balanced comparison, self-detection accuracy is 1.42 percentage points lower (95% CI $[-1.69, -1.15]$) on average. Despite both differences are statistically significant ($p < 0.0001$), their magnitudes are small. Overall, as shown in Table 9 and Table 10 in Appendix A.3.2, six detectors show significant differences under both analyses, while GPT-4o is significant only in the first analysis. In summary, the results confirm that the average differences between self- and cross-detection are small, while their direction and magnitude vary across individual detectors.

Table 8: Generation-level within-generation self- and cross-detection results.

Metric	Generation	Self (%)	Matched cross (%)	Δ (pp)	95% CI (pp)	p_{Holm}
Accuracy	G1	51.1	51.9	-0.83	$[-1.79, +0.16]$	0.1876
	G2	60.6	62.3	$\mathbf{-1.65}$	$\mathbf{[-2.14, -1.15]}$	**0.0003**
	G3	91.3	91.5	-0.21	$[-0.59, +0.17]$	0.2696
F1-score	G1	20.8	22.4	-1.62	$[-3.73, +0.58]$	0.2860
	G2	52.3	54.8	$\mathbf{-2.45}$	$\mathbf{[-3.20, -1.71]}$	**0.0003**
	G3	90.1	90.4	-0.28	$[-0.80, +0.25]$	0.3017

Note: Self and matched-cross scores are percentages; differences and confidence intervals are percentage points. The two-sided bootstrap p-values are Holm-adjusted across G1, G2, and G3 separately for each metric; significant results at $\alpha = 0.05$ are shown in bold.

Table 9: Detector-level model-weighted all-generation sensitivity results.

		Accuracy			F1-score		
Generation	Detector	Δ (pp)	95% CI (pp)	p_{Holm}	Δ (pp)	95% CI (pp)	p_{Holm}
G1	GPT-3.5 Turbo Instruct	-0.56	$[-0.96, -0.12]$	0.0632	-1.96	$[-3.36, -0.43]$	0.0686
G1	Mixtral 8x22B	-1.84	$[-3.20, -0.49]$	0.0632	-3.45	$[-6.45, -0.51]$	0.1338
G2	DeepSeek-V3	-3.55	$[-5.06, -1.96]$	**0.0015**	-4.44	$[-6.76, -2.07]$	**0.0018**
G2	GLM-4-32B	$+0.11$	$[-1.33, +1.52]$	1.0000	$+0.47$	$[-1.87, +2.71]$	1.0000
G2	GPT-4o	$+1.73$	$[+0.93, +2.51]$	**0.0015**	$+1.98$	$[+1.14, +2.79]$	**0.0015**
G2	Llama 3.3 70B	$+7.12$	$[+6.41, +7.83]$	**0.0015**	$+7.67$	$[+6.90, +8.46]$	**0.0015**
G2	Qwen2-72B	-0.25	$[-0.50, +0.04]$	0.3470	-0.84	$[-1.78, +0.24]$	0.4092
G3	Claude Opus 4.7	$+0.29$	$[+0.06, +0.49]$	0.0632	$+0.31$	$[+0.07, +0.51]$	0.0608
G3	DeepSeek-V4-Pro	-5.16	$[-6.70, -3.65]$	**0.0015**	-6.34	$[-8.52, -4.26]$	**0.0015**
G3	Gemini 3.1 Pro Preview	-5.34	$[-6.35, -4.37]$	**0.0015**	-5.69	$[-6.83, -4.61]$	**0.0015**
G3	GLM-5.1	-7.14	$[-8.95, -5.27]$	**0.0015**	-10.32	$[-13.21, -7.41]$	**0.0015**
G3	GPT-5.5	$+0.01$	$[-0.08, +0.08]$	1.0000	$+0.01$	$[-0.08, +0.08]$	1.0000
G3	Kimi K2.6	$+0.03$	$[-0.36, +0.38]$	1.0000	$+0.03$	$[-0.36, +0.39]$	1.0000
G3	MiMo V2.5 Pro	$+0.93$	$[-0.25, +2.07]$	0.4840	$+1.63$	$[+0.17, +3.04]$	0.1338
G3	Qwen3.6 Max Preview	$+0.89$	$[+0.65, +1.11]$	**0.0015**	$+0.93$	$[+0.68, +1.15]$	**0.0015**

Note: Differences and confidence intervals are reported in percentage points. The cross-detection baseline includes all 14 non-self generators, each receiving equal weight. Δ is self-detection minus cross-detection. Confidence intervals are obtained from 10,000 prompt-clustered bootstrap iterations. Holm-adjusted p-values control for 15 detector-level tests; significant results at $\alpha = 0.05$ are shown in bold.

Table 10: Detector-level generation-balanced all-generation sensitivity results.

		Accuracy			F1-score		
Generation	Detector	Δ (pp)	95% CI (pp)	p_{Holm}	Δ (pp)	95% CI (pp)	p_{Holm}
G1	GPT-3.5 Turbo Instruct	-0.28	$[-0.67, +0.17]$	1.0000	-0.93	$[-2.34, +0.59]$	1.0000
G1	Mixtral 8x22B	-1.81	$[-3.21, -0.39]$	0.0999	-3.60	$[-6.68, -0.58]$	0.1863
G2	DeepSeek-V3	-5.29	$[-6.80, -3.69]$	**0.0015**	-7.05	$[-9.37, -4.68]$	**0.0015**
G2	GLM-4-32B	-0.49	$[-1.96, +0.96]$	1.0000	-0.57	$[-2.92, +1.72]$	1.0000
G2	GPT-4o	$+0.67$	$[-0.13, +1.44]$	0.7551	$+0.83$	$[-0.02, +1.62]$	0.3856
G2	Llama 3.3 70B	$+5.42$	$[+4.77, +6.10]$	**0.0015**	$+5.78$	$[+5.08, +6.50]$	**0.0015**
G2	Qwen2-72B	-0.11	$[-0.35, +0.17]$	1.0000	-0.33	$[-1.25, +0.71]$	1.0000
G3	Claude Opus 4.7	$+0.11$	$[-0.12, +0.31]$	1.0000	$+0.12$	$[-0.11, +0.32]$	1.0000
G3	DeepSeek-V4-Pro	-6.72	$[-8.27, -5.23]$	**0.0015**	-8.32	$[-10.51, -6.26]$	**0.0015**
G3	Gemini 3.1 Pro Preview	-5.43	$[-6.44, -4.46]$	**0.0015**	-5.78	$[-6.93, -4.70]$	**0.0015**
G3	GLM-5.1	-7.24	$[-9.06, -5.35]$	**0.0015**	-10.46	$[-13.37, -7.55]$	**0.0015**
G3	GPT-5.5	-0.03	$[-0.12, +0.05]$	1.0000	-0.03	$[-0.12, +0.05]$	1.0000
G3	Kimi K2.6	-0.09	$[-0.47, +0.25]$	1.0000	-0.08	$[-0.47, +0.26]$	1.0000
G3	MiMo V2.5 Pro	-0.67	$[-1.84, +0.48]$	1.0000	-0.43	$[-1.86, +0.98]$	1.0000
G3	Qwen3.6 Max Preview	$+0.58$	$[+0.35, +0.77]$	**0.0015**	$+0.60$	$[+0.37, +0.80]$	**0.0015**

Note: Differences and confidence intervals are reported in percentage points. The cross-detection baseline includes all 14 non-self generators. G1, G2, and G3 receive equal total weight, which is divided equally among the models within each generation. Δ is self-detection minus cross-detection. Confidence intervals are obtained from 10,000 prompt-clustered bootstrap iterations. Holm-adjusted p-values control for 15 detector-level tests; significant results at $\alpha = 0.05$ are shown in bold.

A.3.3 False-positive and false-negative patterns

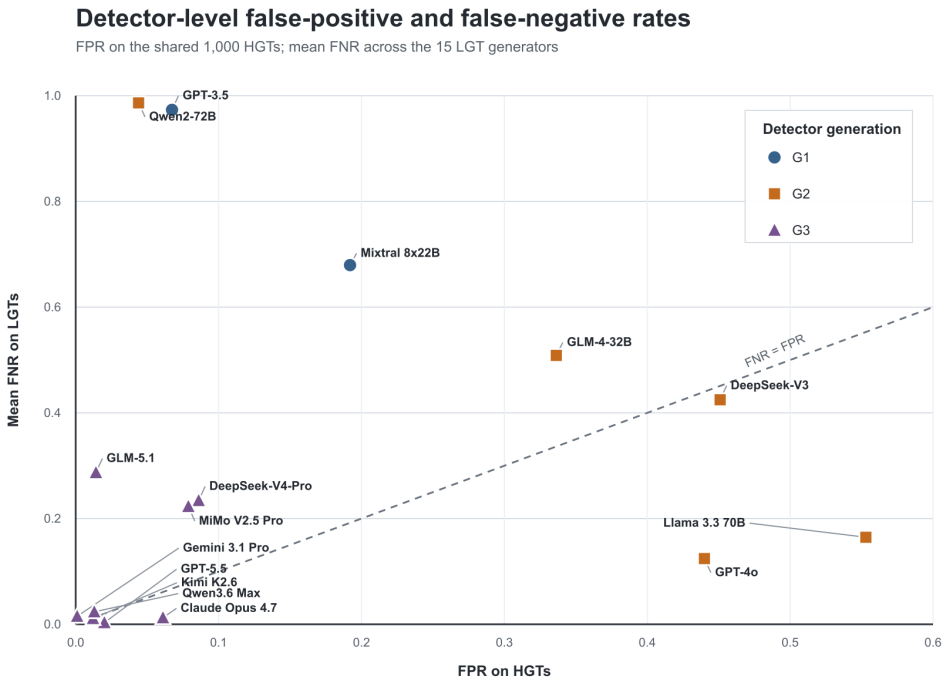


Figure 1: Detector-level error profiles for the 15 selected models.

Figure 1 illustrates the detector-level error profiles for all 15 selected LLM models. Each point in the figures presents a detector's FPR on the shared HGT corpus and its mean FNR across texts

produced by all 15 generators across generations. Detectors in different generations are color coded as indicated in the legend. The dashed line indicates equal FNR and FPR. Points above the dashed line make proportionally more false-negative errors, whereas points below it make proportionally more false-positive errors. As shown in Figure 1, individual detectors within the same generation can exhibit different error patterns. On top-left corner of the figure, GPT-3.5 Turbo Instruct and Qwen2-72B have high mean FNRs but low FPRs, indicating a strong tendency to assign the human label. In contrast, on the right-bottom region of the figure, GPT-4o and Llama 3.3 70B reduce FNRs but have comparatively high FPRs. Most G3 detectors are concentrated in the lower-left region, indicating low rates for both FNRs and FPRs. Nevertheless, GLM-5.1, DeepSeek-V4-Pro, and MiMo V2.5 Pro are more error-prone compared with the rest of G3 detectors.

A.3.4 Semantic error pattern analysis

It is impractical to perform a full manual analysis of all explanations, to facilitate the analyses, we selected two generators and two detectors from each generation. The generators selected are GPT-3.5 Turbo Instruct and Mixtral 8x22B (G1), GPT-4o and DeepSeek-V3 (G2), and GPT-5.5 and MiMo V2.5 Pro (G3). The detectors selected are the same G1 and G2 models, together with GPT-5.5 and DeepSeek-V4-Pro for G3.

Table 11: Overview of the sampled semantic-analysis dataset.

Panel A: Detector agreement by text type				
Text type	Texts	Disagreement	Pairwise comparisons	Label agreement %
HGT	20	8	300	83.7
LGT	120	120	1,800	48.4
Panel B: Classification outcomes by detector				
Detector	TP	TN	FP	FN
GPT-3.5 Turbo Instruct (G1)	0	20	0	120
Mixtral 8x22B (G1)	30	19	1	90
GPT-4o (G2)	93	15	5	27
DeepSeek-V3 (G2)	70	17	3	50
GPT-5.5 (G3)	120	20	0	0
DeepSeek-V4-Pro (G3)	93	18	2	27

We randomly sampled 20 text generation prompts. For every sampled generation prompt, we retained its associated HGT and all six LGTs produced by the selected generators. Direction 1 therefore contains 140 unique texts: 20 HGTs and 120 LGTs (20 prompts $\times$ 6 generators). Each text is judged by six detectors, producing 120 explanations for the HGTs and 720 explanations for the LGTs (i.e, 840 explanations in total). Direction 2 reorganises the 720 LGT explanations into 120 detector–text-generation-prompt comparison groups (6 detectors $\times$ 20 prompts), with each group comparing the six generator outputs.

Table 11 provides an overview of this sampled dataset. Panel A reports whether the six detectors agree on the label assigned to each text, while Panel B reports the resulting classification outcomes for each detector. All 120 LGTs produce disagreement among the six detectors, compared with eight of the 20 HGTs. Across the 840 judgements, TP and FN sum to 720 judgements on LGTs, whereas TN and FP sum to 120 judgements on HGTs.

A.3.5 LLM-assisted semantic error pattern analysis

The LLM-assisted analysis uses separate prompts for Direction 1, Direction 2, and the synthesis of the two outputs.

Listing 5: Direction 1 review prompt

```
Task: Semantic pattern discovery in detector explanations (Direction 1)

You are reviewing explanations from an LLM-generated-text detection study. Each record holds
    one exact text constant and presents judgements from six detectors: two from G1, two from
     G2, and two from G3. The records include HGTs and LGTs from two representative
    generators per generation.
```

```
Analyse the explanations semantically rather than merely counting keywords. Distinguish an
     accurate observation about the text from a valid inference about the classification label
     . A grounded explanation can still be logically unreliable.

Report:
1. reasoning strategies used by each detector and detector generation;
2. whether explanations cite concrete textual evidence or use generic claims;
3. whether detectors that agree on a label also agree on the reason;
4. recurring rationale patterns preceding false positives and false negatives;
5. whether the same textual cue is interpreted in opposite directions;
6. confidence framing and whether confidence appears to track correctness; and
7. similarities and differences across detector generations.

Use only the supplied records. Cite evidence as source + text_id + detector. Do not invent
     frequencies. Give exact counts only when they can be directly tallied from the records;
     otherwise use qualitative terms such as recurring or occasional. Treat generation-wide
     interpretations as exploratory because only two detectors represent each generation.

Output:
- an executive summary;
- a table of named semantic patterns, affected detectors/outcomes, and example record
     references;
- detector-level and generation-level findings;
- contradictory or inverted cue examples;
- limitations;

Data follows.
```

Listing 6: Direction 2 review prompt

```
Task: Semantic pattern discovery in detector explanations (Direction 2)

You are reviewing explanations from an LLM-generated-text detection study. Each group holds
     the detector and text generation prompt constant while varying the LGT source across six
     generators: two from G1, two from G2, and two from G3. All texts in this bundle are
     machine-generated. This view tests whether the same detector changes its label or
     reasoning for different generator outputs written in response to the same text generation
      prompt.

Analyse the explanations semantically rather than merely counting keywords. Distinguish
     genuine sensitivity to textual evidence from a detector that appears consistent only
     because it repeatedly returns the same label.

Report:
1. detector-specific stability and changes across generator outputs;
2. generator or generator-generation properties associated with changed labels;
3. whether a stable label is accompanied by stable reasoning;
4. recurring false-negative rationales, especially on G3-generated texts;
5. cases where the same detector interprets similar cues differently; and
6. implications for comparing self- and cross-detection explanations.

Use only the supplied records. Cite evidence as detector + text_id + source. Do not invent
     frequencies. Give exact counts only when they can be directly tallied from the records;
     otherwise use qualitative terms. Treat generation-wide interpretations as exploratory
     because only two generators represent each generation.

Output:
- an executive summary;
- a table of named semantic patterns, affected detectors/sources, and example record
     references;
- detector-level and generator-level findings;
- limitations;

Data follows.
```

Direction 1 examines whether detectors reach the same judgement and use the same reasoning for the same text. For each text, the LLM first identifies the textual properties all explanations and then compares how each detector link those properties to a "human" or "machine" label. The LLM identifies five recurring forms of disagreement via this process. As reported in Table 12, detectors can notice same characteristics of the writing but interpret them differently to make different decisions. For example, detailed and coherent writing may be treated as evidence of human expertise by one detector, but as evidence of formulaic LLM generation by another. Similar observations of disagreements arise for personal voice, formal style, and textual artefact.

Direction 2 examines whether a detector applies consistent reasoning to outputs generated by different generators with the same text generation prompt. We observed that Mixtral 8x22B , GPT-4o, DeepSeek-V3, and DeepSeek-V4-Pro change their labels on text generated by different generators, and Table 13 shows examples of how GPT-4o, DeepSeek-V3 and DeepSeek-V4-Pro as detectors change their judgements on text generated by the same prompt across three generations. For the 'academic-pressure' prompt, GPT-4o treats empathy and conversational language as human evidence in four outputs, but classifies two more regularly structured dialogues as machine-generated. For the 'unreliable-narration' prompt, DeepSeek-V3 classifies the uniformly structured GPT-3.5 output as machine-generated but treats the Mixtral output as human-written because it integrates literary examples and appears more nuanced. For the 'climate-architecture' prompt, DeepSeek-V4-Pro classifies all G1 and G2 outputs as machine-generated but both G3 outputs as human-written, describing the latter as more natural and less formulaic. These comparisons show that the judgement and explanation can change with the generated text even when the detector and text generation prompt (i.e., same topic under the same rules) remain unchanged.

Table 12: Direction 1 examples: identical texts judged by different detectors.

Generator	Prompt ID	Shared text excerpt	Detector 1: judgement and explanation	Detector 2: judgement and explanation	Observed difference
GPT-3.5	40	"Unreliable narration is a literary technique where the narrator's credibility is called into question."	GPT-3.5: H (FN). Detailed literary analysis is described as beyond an LLM's capability.	GPT-5.5: M (TP). Generic organisation, repeated claims, standard transitions, and the lack of concrete examples support the machine label.	The detectors draw opposite conclusions about the depth of the same text.
Human	43	Factual history of Brentford Football Club.	GPT-3.5: H (TN). Specific records, statistics, and recent updates are treated as human evidence.	GPT-5.5: H (TN). Article cross-references and an incomplete phrase are treated as signs of a copied human-edited source.	The detectors agree on the label but rely on different reasons.
GPT-3.5	113	"Medical malpractice occurs when a healthcare provider fails to follow the accepted standard of care."	DeepSeek-V3: H (FN). Specific examples, coherence, and a detailed approach are treated as human evidence.	DeepSeek-V4-Pro: M (TP). Predictable transitions, generic coverage, and uniform tone are treated as LGT cues.	The same organisation and detail support opposite labels and reasons.
Mixtral	146	"I've always found that the true essence of a city lies beyond its well-trodden tourist paths."	Mixtral: H (FN). Sensory details and personal recollection are interpreted as lived experience.	GPT-5.5: M (TP). Familiar travel phrases and a decorative anecdote are interpreted as generated personalisation.	The same personal voice is treated as genuine or simulated.
Human	425	"Indian Mary Park is part of the Josephine County Parks system."	DeepSeek-V4-Pro: M (FP). Missing distances are interpreted as model generation errors.	GPT-5.5: H (TN). The same omissions are interpreted as extraction or formatting artefacts.	The same textual defect supports opposite provenance inferences.

Note: H and M denote predicted human and machine labels, respectively. Within each row, both detectors judge the same complete text; the third column presents a short excerpt from that shared text. TP, TN, FP, and FN denote the corresponding classification outcomes.

Table 13: Direction 2 examples: one detector judging outputs generated from the same prompt.

Detector	Prompt ID and topic	Generators and output excerpts	Output 1: judgement and explanation	Output 2: judgement and explanation	Observed difference
GPT-4o	43: academic pressure	GPT-3.5: "Hey there! How have you been dealing with all the academic pressure lately?" GPT-4o: "Alice: Hey Ben, you've been looking a bit stressed lately."	GPT-3.5 output: H (FN). Empathy, emotional support, and conversational markers are treated as human evidence.	GPT-4o output: M (TP). Sequential advice and a deliberate narrative arc are treated as machine evidence.	Different dialogues lead the same detector to opposite labels and reasons.
DeepSeek-V3	40: unreliable narration	GPT-3.5: "Unreliable narration is a literary technique where the narrator's credibility is called into question." Mixtral: "Unreliable narration is a narrative technique where the credibility of a narrator is compromised."	GPT-3.5 output: M (TP). Uniform structure, systematic coverage, and repetition support the machine label.	Mixtral output: H (FN). Nuanced analysis and integrated literary examples support the human label.	The judgement changes with the depth and form of the generated text.
DeepSeek-V4-Pro	34: climate-responsive architecture	GPT-3.5: "Climate change is a global issue that affects every aspect of our lives, including the design of our buildings." GPT-5.5: "Future architectural designs are likely to become more climate-responsive, flexible, and resource-efficient."	GPT-3.5 output: M (TP). Formulaic organisation, generic language, and predictable progression support the machine label.	GPT-5.5 output: H (FN). Progressive examples and the absence of awkward phrasing support the human label.	Outputs from different generator generations lead to opposite judgements.

Note: The detector and text generation prompt are fixed within each row, while the generator and resulting text change. Output 1 and Output 2 correspond, in order, to the two generators listed in the third column. H and M denote predicted human and machine labels; because all evaluated texts are LGTs, H is an FN and M is a TP.

A.3.6 CUE-BASED ANALYSIS

Table 14: Explanation cue categories and their links to prior literature.

Cue category	Interpretation in detector explanations	Supporting literature
Structure and formatting	Organisation, paragraph structure, formatting, or ordering of ideas	Previous studies found that human evaluators use formatting, textual structure, and sentence organisation to make a judgment (Clark et al., 2021; Russell et al., 2025).
Fluency and polish	Grammar, spelling, fluency, formality, smoothness, or tonal consistency	Grammar, fluency, formality, and clarity are commonly used as detection cues, although they may support either a human or machine judgement (Clark et al., 2021; Mitrović et al., 2023; Russell et al., 2025).
Generic or abstract language	Broad, generic, shallow, or insufficiently original content	LGTs are often perceived as focusing on general concepts and lacking detail or originality (Mitrović et al., 2023; Russell et al., 2025).
Hedging and uncertainty	Qualified claims and expressions of uncertainty or epistemic caution	Human evaluators sometimes use expressions of uncertainty to make a judgment, although uncertainty is not a consistently reliable detection cue (Clark et al., 2021).
Personal voice	Personal pronouns, feelings, anecdotes, lived experience, or individual perspective	Personal expression and emotional language are often associated with human writing and may cause LGTs to be misclassified as human-written (Clark et al., 2021; Mitrović et al., 2023; Russell et al., 2025).
Coherence and transitions	Logical flow, consistency, repetition, cohesion, or formulaic connections	Previous studies found that evaluators frequently consider coherence, repetition, and sentence structure when making a judgment (Clark et al., 2021; Russell et al., 2025).
Specificity and evidence	Concrete details, examples, citations, data, or other supporting evidence	Evaluators often use detail and specificity to make a judgment, but these cues can also lead to incorrect decisions (Clark et al., 2021; Mitrović et al., 2023; Russell et al., 2025).
Factuality and hallucination	Factual errors, contradictions, implausible claims, or unsupported content	Factual errors and contradictions may be used as signs of machine generation, while hallucination is a recognised problem in generated text (Ippolito et al., 2020; Clark et al., 2021; Ji et al., 2023).

To help identify the eight cue categories reported in Table 14 among LLM-generated explanations, we predefined keywords and phrases as reported in Table 15. We first converted each explanation to lowercase, then search these terms in each explanation. A category is recorded when at least

Table 15: Keywords and phrases used to identify explanation cues.

Cue category	Predefined keywords and phrases
Structure and formatting	`structured`; `format`; `paragraph`; `introduction`; `conclusion`; `essay`; `organization`
Fluency and polish	`polished`; `fluent`; `smooth`; `grammatically`; `well-formed`; `well formed`; `even tone`
Generic or abstract language	`generic`; `abstract`; `high level`; `encyclopedic`; `surface-level`; `surface level`; `broad`
Hedging and uncertainty	`hedge`; `hedged`; `hedging`; `in most cases`; `tends to`; `may`; `might`; `depending on`
Personal voice	`personal voice`; `idiosyncr*`; `anecdote`; `lived experience`; `individual perspective`
Coherence and transitions	`transition`; `coheren*`; `flow`; `parallel`; `balanced`; `systematic*`
Specificity and evidence	`specific`; `citation`; `example`; `concrete`; `detail`; `data`; `factual`
Factuality and hallucination	`hallucination`; `factual error`; `inaccur*`; `wrong`; `confusion`; `contradict*`

Note: Matching is case-insensitive and requires complete words unless an asterisk is shown. An asterisk denotes any continuation of the word stem; for example, `coheren*` matches `coherent` and `coherence`. Hyphenated and unhyphenated forms are both matched where both forms are listed. Each cue category is counted at most once within an explanation.

one of its associated terms are identified in the explanation. One explanation may contain multiple categories, thereby a match indicates that the detector refers to a cue in its explanation but this cannot be considered as direct support of the corresponding binary classification decision.

We then compared cue frequencies between true-positive and false-negative explanations for each detector. As reported in Panel B of Table 11, GPT-3.5 labels all 120 sampled LGTs as human and therefore has no true positives, whereas GPT-5.5 correctly labels all 120 LGTs and has no false negatives. Within-detector comparisons are therefore available only for Mixtral, GPT-4o, DeepSeek-V3, and DeepSeek-V4-Pro, which have observations in both outcome groups.

Furthermore, we observed that four cue categories are largely consistent with the recurring findings of LLM-assisted analysis results. *Structure and formatting* captures references to regular or formulaic organisation, while *generic or abstract language* captures references to broad and insufficiently distinctive coverage. The LLM-assisted analysis found that detectors repeatedly use these two cue categories to support machine judgements. In addition, *specificity and evidence* corresponds to the finding from LLM-assisted analysis that suggests detailed examples, technical knowledge, or apparent expertise can lead detectors to incorrectly assign the human label. *Coherence and transitions* is the cue-category corresponding to cue-inversion pattern observed in the LLM-assisted analysis, where the same clear and logically organised structure can lead different detectors to assign opposite labels. For instance, some detectors treat such patterns as evidence of thoughtful human writing, whereas others treat it as a formulaic pattern typical of LLM-generated text. Since both LLM-assisted analysis and cue-based analysis converge on these four categories, we computed how frequently each cue is mentioned in true-positive and false-negative explanations, as reported in Table 16.

Structure and formatting are considerably more frequent in true-positive than false-negative explanations for every detector. For example, this cue appears in 91.4% of GPT-4o's true positives but only 33.3% of its false negatives. In one true-positive example, DeepSeek-V4-Pro describes a explanation generated by G1 model (i.e., GPT-3.5) as following a "highly structured and formulaic pattern". *Generic or abstract language* follows a similar pattern. These results support the LLM-assisted finding that detectors commonly associate regular organisation and generic coverage with machine generation. *Specificity and evidence* shows the opposite tendency. For Mixtral, GPT-4o, DeepSeek-V3, and DeepSeek-v4-Pro, explanations that have specific details and evidence are more frequently judged as false-negative than true-positive. For instance, DeepSeek-V3 refers to "specific examples" and a "detailed and structured approach" as reasoning evidence to incorrectly classify an explanation generated by G1 model (i.e., GPT-3.5) as human-written. This is consistent with the LLM-assisted analysis examples in Table 12, where technical details, literary examples, and sensory descriptions are interpreted as evidence of human authorship. *Coherence and transitions*, however, appears frequently in both correct and incorrect judgement. For example, it appears in 83.3% of Mixtral's true-positive explanations and 85.6% of its false-negative explanations. This supports the cue-inversion finding, where detectors frequently mention coherence in their reasoning, but they may interpret it as either naturally organised human writing or predictable LLM behaviour. Overall, the

cue-based results provide convergent evidence for several patterns identified by the LLM-assisted analysis, while also showing that the same cue does not consistently lead to the same decision of human/LLM authorship.

Table 16: Cue frequencies in correct and incorrect LGT judgements.

Detector	Label (n)	SF	GAL	CT	SE
GPT-3.5 Turbo Instruct	FN (120)	10.0	2.5	20.0	41.7
Mixtral 8x22B	TP (30)	93.3	73.3	83.3	40.0
	FN (90)	61.1	46.7	85.6	65.6
GPT-4o	TP (93)	91.4	20.4	77.4	34.4
	FN (27)	33.3	11.1	66.7	48.1
DeepSeek-V3	TP (70)	90.0	57.1	78.6	32.9
	FN (50)	28.0	40.0	80.0	42.0
GPT-5.5	TP (120)	83.3	98.3	76.7	91.7
DeepSeek-V4-Pro	TP (93)	93.5	87.1	63.4	62.4
	FN (27)	40.7	63.0	66.7	77.8

Note: SF = Structure and Formatting; GAL = Generic or Abstract Language; CT = Coherence and Transitions; SE = Specificity and Evidence. Values are percentages of explanations containing at least one predefined term from the corresponding cue category. An explanation can contain multiple categories. GPT-3.5 has no true positives and GPT-5.5 has no false negatives in this review sample, so within-detector contrasts are unavailable for these models.

A.4 Supporting materials for Section 5

Table 17 presents our exploratory review on public documentation related to watermarking from different LLM vendors. These public documentations were reviewed on 14 September 2026. "No public model- and endpoint-specific claim identified" does not demonstrate that a watermark is absent, instead, it means that we found no documentation confirming its use for the endpoint through which the experimental text was generated. Google limits its public deployment claim to the Gemini app and web experience Google DeepMind (2024). Anthropic describes a future-model rollout and a transition period for older models Anthropic (2026b). OpenAI documents deployed provenance signals for images and audio while describing text provenance as ongoing work OpenAI (2026). ΔAccuracy is self-detection accuracy minus matched same-generation cross-detection accuracy. Positive values represent self-detection advantage while negative values represent self-detection disadvantage, and bold values indicate statistically significant differences after Holm correction.

Table 17: Public text-watermark evidence and within-generation self-detection results.

Generation	Model	Public evidence for text watermarking on the evaluated endpoint	ΔAcc (pp).
G1	GPT-3.5 Turbo Instruct	Text watermark researched by provider; no deployment confirmed	$+0.18$
	Mixtral 8×22B Instruct	No public model- and endpoint-specific claim identified	-1.85
G2	DeepSeek-V3	No public model- and endpoint-specific claim identified	$\mathbf{-8.15}$
	GLM-4 32B 0414	No public model- and endpoint-specific claim identified	-2.06
	GPT-4o	Text watermark researched by provider; no deployment confirmed	-0.99
	Llama 3.3 70B Instruct	No public model- and endpoint-specific claim identified	$\mathbf{+2.96}$
	Qwen2-72B	No public model- and endpoint-specific claim identified	$+0.01$
G3	Claude Opus 4.7	Provider-level rollout announced; not confirmed for this pre-transition model endpoint	$\mathbf{+0.86}$
	DeepSeek-V4-Pro	No public model- and endpoint-specific claim identified	$+0.05$
	Gemini 3.1 Pro Preview	SynthID-Text documented for Gemini app/web; not confirmed for the evaluated API route	$\mathbf{-5.06}$
	GLM-5.1	No public model- and endpoint-specific claim identified	$\mathbf{-5.76}$
	GPT-5.5	Text watermark researched by provider; no deployment confirmed	$+0.1$
	Kimi K2.6	No public model- and endpoint-specific claim identified	$+0.35$
	MiMo V2.5 Pro	No public model- and endpoint-specific claim identified	$\mathbf{+5.90}$
	Qwen3.6 Max Preview	No public model- and endpoint-specific claim identified	$\mathbf{+1.86}$