QUESTION OF CONFIDENCE: A ZERO-SHOT AI-TEXT DETECTOR AND BENCHMARK SUITE

Anonymous authors
Paper under double-blind review

ABSTRACT

Ask a large language model whether a person or a machine wrote a text, and it answers in words. Before it writes "human" or "machine", the model already holds a probability distribution over the possible answers. We show that this distribution is a better detection signal than a probability the model is asked to write out, its *stated probability*. We propose two detectors built on it. LogJudge asks the model once and scores the text by the *token probability* it places on "human" versus "machine" before it answers. FusionJudge averages the token probability with the stated probability. Since a detector must rarely accuse a human, we count how many machine-written texts each detector catches while flagging at most 1% of human-written ones. Two state-of-the-art detectors, Fast-DetectGPT and Binoculars, catch 52% and 51%. With a low-cost commercial model as the judge, the stated probability catches 50%, LogJudge 65%, and FusionJudge 76%. The stated probability orders texts well, but it takes only a few rounded values, so many texts tie right where the decision is made. The token probability is fine-grained and separates them. Their average, FusionJudge, keeps the stated probability's order, breaks its ties with the token probability, and follows the token probability where it is confident. Current detectors are near ceiling on benchmarks written by 2023-era models, so we release Manugram, about 30,000 texts that recreate a standard benchmark with four current models and add scientific abstracts, some of them written by automated AI-Scientist systems. Averaged over the four benchmarks, FusionJudge is the strongest detector we tested. Neither detector needs training, so their accuracy holds when the topic changes, while a trained detector's accuracy collapses. They also read only the few probabilities that commercial APIs already return, which means strong API models can be used as detectors without any extra work. We release the code and data.[1]

1 INTRODUCTION

Machine-written text is now everywhere people write: in student essays, news, peer review, and scientific papers, some of which are drafted end to end by automated AI-Scientist systems (Lu et al., 2024; Yamada et al., 2025). Telling it apart from human writing matters, but the two possible errors are not equally bad. Missing a machine-written text is a nuisance, while accusing a human author does real damage. A useful detector must therefore work under a strict false-alarm budget, flagging at most, say, one human text in a hundred, and be judged by how much machine-written text it still catches. Under this budget the best current detectors catch only about half of it, and the benchmarks that certified them were written by 2023 models on which they now score near ceiling. We ask how to catch more machine text at a 1% false-alarm rate, on text from current generators, using the most capable models available.

Existing detectors fall into three families. Watermarking embeds a signal at generation time and works only when the generator cooperates (Kirchenbauer et al., 2023; Dathathri et al., 2024). Trained classifiers learn from labeled human and machine text (Solaiman et al., 2019; Hu et al., 2023; Chen et al., 2023; Verma et al., 2024; Mao et al., 2024; Chen et al., 2025) and lose much of their accuracy when the domain or generator shifts. Zero-shot detectors need no training. Probability-based ones score how predictable each word of the text is under an open reference model, through curvature, cross-perplexity, or related statistics (Mitchell et al., 2023; Bao et al., 2024; Hans et al., 2024; Xu et al., 2025). They need the model's per-token probabilities and ignore what the text says.

[1] Code: `https://anonymous.4open.science/r/logjudge-manugram-D902`. Data: `https://osf.io/gsbtz/overview?view_only=592bbbf2dcca47acab265ed57f591f02`.

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.

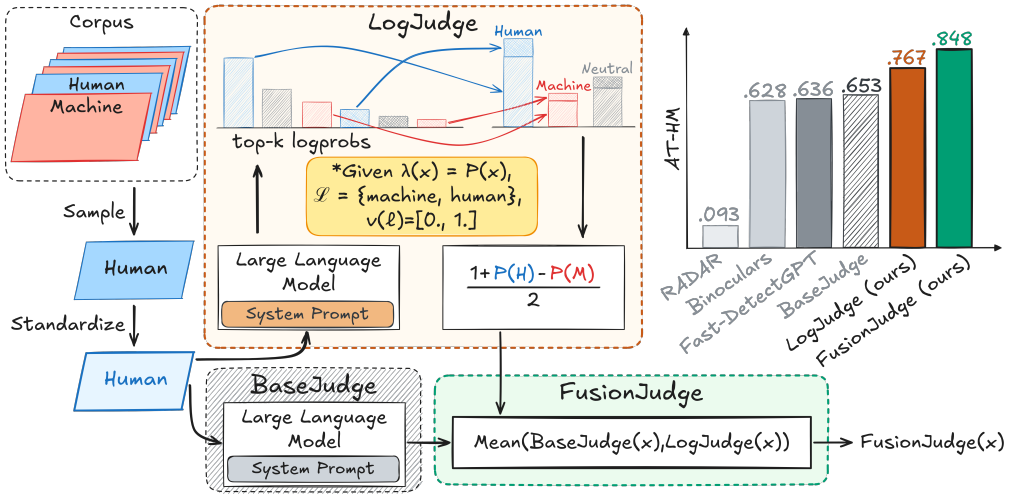


Figure 1: **Overview.** An LLM judge scores a document in two ways. LogJudge (top) reads the top-k log-probabilities at the single answer token and scores the document by the expected human-likeness of the answer distribution, Eq. 2 for the binary scheme $\mathcal{L} = \{\texttt{human}, \texttt{machine}\}$ with $v(\texttt{human}) = 1$, $v(\texttt{machine}) = 0$, and $\lambda(x) = P(x)$. The *Neutral* bar is the remaining mass $1 - P(x)$, which falls outside the top-k window or on tokens that spell neither label, scored as $\frac{1}{2}$ (no evidence either way). BaseJudge (bottom) asks the same judge to state a probability in text. FusionJudge averages the two scores. Right: mean AT-HM (our harmonic-mean summary of ranking quality and recall at 1% false alarms) over our benchmarks for the detectors we compare.

Prompt-based ones use an LLM's judgment of the whole text, either by having it rewrite the text and measuring how much it changes (Zhu et al., 2023) or by asking it directly whether a person or a machine wrote it (Bhattacharjee & Liu, 2024). The direct question takes the model's written answer at face value, yet a written label or a stated probability is coarse (Xiong et al., 2024), and probabilities carry more information than such hard answers (Elkan, 2001; Gneiting & Raftery, 2007).

We show that the LLM judge already holds a finer answer than the one it writes. LogJudge keeps the judge but reads the probability it places on "human" versus "machine" in its single answer token (Figures 1 and 2). FusionJudge averages this with the judge's written probability. Both need no training and only the few top-ranked probabilities that commercial APIs already return, so the strongest models can serve as detectors. With `gpt-5.6-luna` (OpenAI, 2026) as judge, at a 1% false-alarm rate, the stated probability catches 50% of machine-written texts, LogJudge 65%, and FusionJudge 76%, against 52% and 51% for Fast-DetectGPT (Bao et al., 2024) and Binoculars (Hans et al., 2024), the strongest existing detectors we tested. To test this on current generators, we build **Manugram**, which regenerates the Ghostbuster benchmark (Verma et al., 2024) with four current models and adds scientific abstracts, including ones written by AI-Scientist systems (Table 1).

Our contributions are:

- **Why written probability answers fall short.** An LLM judge's written answer takes only a few rounded values, so many texts tie right at the decision threshold. Reading the judge's answer-token probabilities instead leaves its ranking quality unchanged but raises recall at 1% false alarms from 50% to 65% with `gpt-5.6-luna` and from 38% to 51% with `gemma-4-31b` (Gemma Team, 2026).
- **Two training-free detectors.** LogJudge scores a text by the judge's probability of "human" versus "machine". FusionJudge averages it with the written probability: the written answer orders texts coarsely, and the log-probabilities break its ties and correct it where they are confident. FusionJudge is the strongest detector on average over our four benchmarks.
- **A current benchmark, Manugram.** About 30k texts: Ghostbuster's news, essays, and fiction regenerated by four current LLMs, scientific abstracts, and abstracts by AI-Scientist systems, plus two rephrasing corpora with 24k texts, half of them rephrased. We release every text with the scores of all detectors we evaluate.
- **Robustness checks.** Our detectors keep their accuracy under topic shift, where a trained detector collapses. Rephrasing affects the two detector families in opposite ways. A judge gains nothing from recognizing text written by its own model.

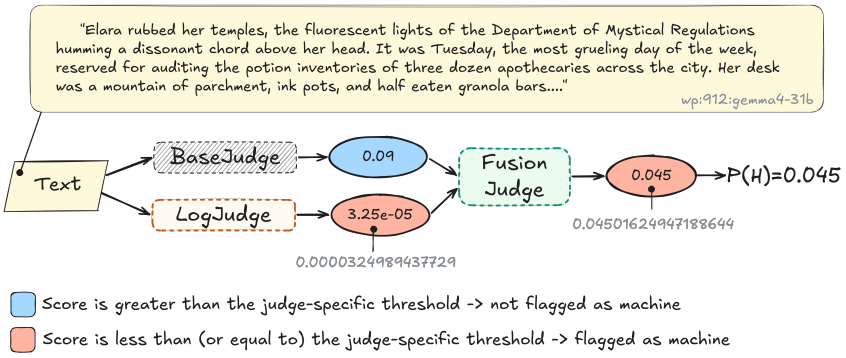


Figure 2: **High-level scoring example.** A machine-written story from Manugram-Ghostbuster (generated by `gemma-4-31b`), scored with `gpt-5.6-luna` backbone. Every score is $P(\text{human})$, and a text is flagged as machine when its score is at or below the detector-specific threshold (red) and passes as human otherwise (blue). Each threshold is set so that at most 1% of the human texts in our four benchmarks, pooled, fall at or below it. Asked to state a probability in text (BaseJudge), the judge answers 0.09, just above its threshold of 0.07, so the text passes as human. Written answers are coarse (Appendix O). Reading the same judge's answer-token log-probabilities (LogJudge) gives 3.25×10^{-5}, below its threshold of 7.45×10^{-5}. FusionJudge averages the two to 0.045, below its threshold of 0.052. Scores are for the full document, the box shows its beginning.

Table 1: **Why a new benchmark.** The Manugram suite next to the legacy Ghostbuster (GB) reference (Verma et al., 2024). The last column gives the AUROC (area under the ROC curve, where 1.00 is perfect separation and 0.50 is chance) of Binoculars (Hans et al., 2024) with its original underlying model. It is at ceiling on the 2023 benchmark and far from it on Manugram (MG). Section 4 defines the split abbreviations. Counts are texts per split: one machine text per human text in GB-orig, one per generator in MG-GB and MG-Sci, 100 matched pairs in MG-AI-Sci, and four classes (human and `gpt-5.6-luna` text, original and rephrased) in the rephrase corpora.

	Split	Domains	Machine text by	Role	Texts	Binoculars AUROC ↑
	GB-orig	essay, news, fiction	2023 generators	legacy reference	5,988	1.00
Proposed	MG-GB	essay, news, fiction	4 current LLMs	core	14,960	0.93
	MG-Sci	CS, mathematics, sociology	4 current LLMs	core	15,000	0.69
	MG-AI-Sci	10 fields	AI-scientist systems	core	200	0.66
	MG-Reph-GB	essay, news, fiction	`gpt-5.6-luna` (orig. and rephrased)	rephrasing study	11,958	–
	MG-Reph-Sci	CS, mathematics, sociology	`gpt-5.6-luna` (orig. and rephrased)	rephrasing study	12,000	–

2 Related Works

Detectors differ in what they need at detection time and in whether their signal reflects the text's meaning or only word-level statistics. Appendix A classifies them along these lines, and Appendix B compares benchmarks in detail.

Watermarking embeds a key-detectable signal at generation (Kirchenbauer et al., 2023; Kuditipudi et al., 2024; Kirchenbauer et al., 2024; Dathathri et al., 2024), which requires control over generation and the key, so it cannot flag text from arbitrary or proprietary models. LogJudge, by contrast, works on any text.

Training-based detection learns from labeled human and machine text, end-to-end (Solaiman et al., 2019; Zellers et al., 2019; Uchendu et al., 2020; Guo et al., 2023; Hu et al., 2023), including LLM judges fine-tuned to answer yes/no (Chen et al., 2025), or with a light head over frozen features (Chen et al., 2023; Wu et al., 2023; Verma et al., 2024; Wang et al., 2023) such as how much an LLM edits a text when rewriting it (Mao et al., 2024). Both need labeled data from the target generator and degrade once the domain drifts. Our detectors need no training.

Zero-shot detection. Probability-based methods assume that machine text sits in high-probability regions of a language model (LM) and score the input's per-token probabilities. They use curvature under perturbation (DetectGPT, Mitchell et al., 2023), computed in a single pass by Fast-DetectGPT (Bao et al., 2024), log-rank (Su et al., 2023), perplexity against cross-perplexity (Binoculars, Hans et al., 2024), time-series statistics (Lastde, Xu et al., 2025), spectral or wavelet statistics (Luo et al., 2026; Özdaş & Osmanoğlu, 2026), shuffling (La Cava & Tagarelli, 2026), a learned

witness function (Zhou et al., 2025), or an instruction-tuned model's log-ratio to its base (Liu et al., 2025; Wu et al., 2026). DNA-GPT compares regenerated endings by n-gram overlap (Yang et al., 2024), and Glimpse estimates token distributions from the top-k probabilities of commercial APIs (Bao et al., 2025). Prompt-based methods look at the whole text. Either an LLM revises it and the two versions are compared (Zhu et al., 2023), or the LLM is asked for a human/AI label (Bhattacharjee & Liu, 2024), and these verdicts turned out to be unreliable. LogJudge sits between the two. It keeps the holistic judgment of the prompt-based approach but scores it with fine-grained probabilities, and it needs only the top-ranked probabilities of one generated token.

Confidence readouts in LLM-as-judge. Reading a judgment from token probabilities has precedents outside detection: G-Eval weights rating tokens by their probabilities (Liu et al., 2023), $P(\text{True})$-style self-assessment reads confidence from a label token (Kadavath et al., 2022), label-token probabilities need calibration (Zhao et al., 2021), and verbalized confidence is coarser than token-level confidence and clusters on round values (Lin et al., 2022; Xiong et al., 2024). Building on this work, we study the readout in AI-text detection, where binary verdicts under a strict false-alarm budget make a written answer's coarseness most costly.

Benchmarks and attacks. Multi-generator, multi-domain benchmarks (Uchendu et al., 2021; He et al., 2024; Li et al., 2024; Wang et al., 2024b; Dugan et al., 2024; Wu et al., 2024) and collections of ChatGPT-written abstracts (Yu et al., 2025; Gao et al., 2023) use 2022–2023 generators. RAID (Dugan et al., 2024) stays hard because of its adversarial attacks, but its unattacked text is just as dated. Manugram regenerates a legacy benchmark's machine half with current generators and adds AI-Scientist abstracts and rephrase corpora. Paraphrasing is the standard evasion attack (Krishna et al., 2023; Sadasivan et al., 2025; Koike et al., 2024). Our rephrased *human* class exposes a failure mode that paraphrase studies of perplexity detectors miss.

3 Reading the Verdict from Log-Probabilities

LogJudge. LogJudge asks the judge the usual question and only changes how we read the answer. As in LLM-as-judge detection (Bhattacharjee & Liu, 2024), a judge LLM receives a fixed system prompt that states the task and the allowed answers, followed by the document. Instead of letting the judge write a verdict and parsing its words, we decode a single token at temperature 0 and record the top-k log-probabilities at that position. Appendix C gives the prompts and API settings.

The prompt fixes an *answer scheme*: a label set $\mathcal{L}$, here $\{\texttt{human}, \texttt{machine}\}$, and a value $v(\ell) \in [0,1]$ per label, 1 for "certainly human" and 0 for "certainly machine". For a document x, let $p_\ell(x)$ be the probability mass the judge places on tokens spelling label ℓ (folding case and stray whitespace or punctuation) and $P(x) = \sum_\ell p_\ell(x)$ the mass captured by the scheme. The LogJudge score is the expected human-likeness under the answer distribution, shrunk toward $\frac{1}{2}$ by the mass the judge did not place on any allowed answer:

$$s(x) \;=\; \lambda(x) \sum_{\ell \in \mathcal{L}} \frac{p_\ell(x)}{P(x)}\, v(\ell) \;+\; \big(1 - \lambda(x)\big) \cdot \tfrac{1}{2}, \qquad \lambda(x) = P(x). \tag{1}$$

Any confidence weight $\lambda(x) \in [0,1]$ gives a valid score in $[0,1]$ that falls back to $\frac{1}{2}$ as $\lambda \to 0$. The choice $\lambda(x) = P(x)$ makes the fallback proportional to the missing mass and adds no hyperparameter. Under the binary scheme, which we use throughout, Eq. 1 collapses to

$$s(x) \;=\; (1 + p_h(x) - p_m(x))/2, \tag{2}$$

with $p_h(x)$ and $p_m(x)$ the mass on `human` and `machine`. The score estimates $P(\text{human})$, so a low score flags machine text. Appendix D gives the general formulation and a five-level ordinal scheme. We leave other confidence weights, value functions, and multi-judge ensembles for future work.

FusionJudge: two readouts of one judgment. In a separate call under a matched system prompt, we can also ask the same judge to state, in text, a calibrated probability that the document is human-written. This stated probability is the written answer of the introduction. The two prompts differ only in the output they request, a label or a probability. We write $s_{\text{base}}(x) \in [0,1]$ for this parsed estimate and call the resulting detector *BaseJudge*, the LLM-as-judge baseline throughout the paper.[2] LogJudge's $s(x)$ and BaseJudge's $s_{\text{base}}(x)$ are two readouts of one judgment: both see the same

[2]Some appendix figures label it "Judge".

evidence, but one comes from the next-token distribution before the model commits to an answer, the other from what the model chooses to write, and the experiments show that they often disagree. FusionJudge is their plain average, $s_{\text{fusion}}(x) = \frac{1}{2}\big(s(x) + s_{\text{base}}(x)\big)$. Both members estimate $P(\text{human})$ on $[0, 1]$, so the average needs no rescaling or fitted weight. Near the decision threshold, however, the two scores differ by orders of magnitude, and the average has two effects. Within each value of the stated probability, the log-probability readout breaks the ties. Where it is confident, it also moves a text across those values, lifting human texts the judge stated as machine-like and pulling in machine texts stated just above the threshold (Section 5).

4 MANUGRAM BENCHMARK

Ghostbuster (Verma et al., 2024) evaluates detectors in and out of domain across news, student essays, and creative writing, but 2023 generators wrote its machine half, and recent detectors are at ceiling on it: with its original Falcon-7B backbone (Almazrouei et al., 2023), Binoculars reaches AUROC 1.00 there, against 0.93, 0.69, and 0.66 on our three Manugram splits below (Table 1, with score distributions in Figure 11 in the appendix). Manugram keeps Ghostbuster's human texts, writing prompts, and domains, regenerates the machine half with four current models under an updated prompt template, and adds scientific abstracts.

Manugram-Ghostbuster (MG-GB) regenerates the machine counterpart of every Ghostbuster human text with four current generators: `gpt-5.6-luna` (OpenAI, 2026), `gpt-oss-120b` (OpenAI, 2025), `gemma-4-31b` (Gemma Team, 2026), and `llama-3.3-70b` (Grattafiori et al., 2024). One prompt template fixes the target length and forbids markdown, so that a detector must key on the writing rather than on formatting artifacts, and both classes pass through the same normalization of dashes and quotation marks (about 15k texts, Appendix F).

Manugram-Science (MG-Sci) extends detection to scientific abstracts in computer science, mathematics, and sociology. Human abstracts published up to 2020 come from OpenAlex (Priem et al., 2022). The four generators write each machine counterpart from the human abstract's title, field, and word count alone (about 15k abstracts, Appendix G). Because the prompt asks for a published abstract's register, with concrete figures, the generated abstracts report invented results by design (see the system prompt in Appendix G).

Manugram-AI-Scientist (MG-AI-Sci) pairs 100 abstracts released by AI-Scientist systems (Villaescusa-Navarro et al., 2025; Lu et al., 2024; Yamada et al., 2025; Weng et al., 2025; Intology, 2025) with human abstracts of matching topic and length, chosen by an optimal assignment over the similarity of their SPECTER2 scientific-paper embeddings (Singh et al., 2023) and their difference in length (Kuhn, 1955; Crouse, 2016). A detector therefore has to rely on authorship rather than on topic or length (Appendix H).

Manugram-Rephrase (MG-Reph-GB/-Sci) has `gpt-5.6-luna` rephrase both the human originals of MG-GB and MG-Sci and its own MG-GB and MG-Sci generations for the same documents, giving a 2×2 design of $\{\text{human, machine}\} \times \{\text{original, rephrased}\}$ (Appendix I). The human class thus holds the original human texts and their rephrasings, and the machine class holds `gpt-5.6-luna`'s own initial generations and their rephrasings.

Original Ghostbuster (GB-orig) is kept, with one machine text per human text, as a legacy reference. Table 1 details all splits.

5 EXPERIMENTS

Metrics. Prior work reports AUROC or F1 (Verma et al., 2024; Mitchell et al., 2023), summaries that hide the task's asymmetry: flagging a human author is far more harmful than missing a machine text, so Hans et al. (2024) report the true positive rate at a low false positive rate. We report both: AUROC, the area under the ROC curve, and TPR@1%FPR, the true positive rate at a 1% false positive rate, i.e., the recall under the introduction's false-alarm budget, with the threshold set post hoc at the 99th percentile of the evaluated split's human-class scores (an oracle threshold, see Appendix J). To rank detectors by one number, we rescale each metric so that chance scores 0 and perfection 1, $\tilde{A} = \max(0,\, 2A - 1)$ and $\tilde{T} = \max\big(0,\, (T - \alpha)/(1 - \alpha)\big)$ with $A = \text{AUROC}$, $T = \text{TPR@}\alpha\text{FPR}$, and $\alpha = 0.01$, and take the AUROC–TPR harmonic mean, $\text{AT-HM} = 2\tilde{A}\tilde{T}/(\tilde{A} + \tilde{T})$ (zero when both are zero). Like F_1, AT-HM is large only when both components are, so it penalizes a detector

Table 2: **Main comparison.** *Left*: AUROC, TPR@1%FPR, and AT-HM averaged over the four splits. AT-HM is computed from the split-averaged AUROC and TPR (Section 5), thus it is not the mean of the per-split AT-HM values on the right. *Right*: AT-HM on each split. The Access column gives what each detector needs from its backbone. T is the text only, K the top-k log-probabilities of one answer token, and F the full per-token log-probabilities of the input. Bold marks the best value per column. Because MG-AI-Sci has only 100 human texts, a single document sets its 1%-FPR threshold. Appendix N shows the same results as a bar chart.

		Mean over four splits			AT-HM per split			
Detector (backbone)	Access	AT-HM	AUROC	TPR@1%FPR	GB-orig	MG-GB	MG-Sci	MG-AI-Sci
FusionJudge (`gpt-5.6-luna`)	K+T	**0.848**	**0.981**	**0.761**	0.951	0.776	**0.948**	0.691
LogJudge (`gpt-5.6-luna`)	K	0.767	0.975	0.648	0.883	0.687	0.794	**0.692**
BaseJudge (`gpt-5.6-luna`)	T	0.653	0.974	0.503	0.692	0.534	0.866	0.462
LogJudge (`gemma-4-31b`)	K	0.633	0.922	0.512	**0.989**	**0.900**	0.284	0.058
BaseJudge (`gemma-4-31b`)	T	0.515	0.905	0.384	0.946	0.722	0.078	0.000
Fast-DetectGPT (`Llama-3.1-8B`)	F	0.636	0.912	0.523	0.535	0.892	0.681	0.373
Binoculars (`Llama-3.1-8B`)	F	0.628	0.912	0.512	0.611	0.897	0.616	0.325
RADAR (released checkpoint, trained)	T	0.093	0.608	0.068	0.206	0.150	0.000	0.017

that ranks well but fails at low FPR, or the reverse. For evaluation, every score is oriented so that larger values indicate machine text. The judge-based scores estimate $P(\text{human})$, so we evaluate $1 - s(x)$ (Appendix D). This orientation is fixed in advance, and we never flip a score post hoc. The *four-split mean* averages AUROC and TPR@1%FPR over GB-orig, MG-GB, MG-Sci, and MG-AI-Sci with equal weight, so that the 15k-text splits do not drown out the 200-text MG-AI-Sci, and computes AT-HM from the averages. We report every per-split value so that readers can reweight. MG-AI-Sci's 1%-FPR threshold rests on one of its 100 human texts.

Baselines and backbones. We compare with the strongest detector of each family in its best configuration. RADAR (Hu et al., 2023) is a trained, adversarially robust classifier, which we apply frozen from its released checkpoint without retraining. Fast-DetectGPT (Bao et al., 2024) and Binoculars (Hans et al., 2024) run on Llama-3.1-8B (Grattafiori et al., 2024), the best of five open-weight 7–8B base/instruct pairs in a sweep over the four splits (Appendix K). BaseJudge (Section 3) is the prompt-based baseline. Of five BaseJudge backbones we tried (`gpt-5.6-luna`, `gemma-4-31b`, `gpt-oss-120b`, `llama-3.3-70b`, `qwen3-14b`), only `gpt-5.6-luna` and `gemma-4-31b` clear chance by a useful margin. For the domain-shift study, we train RoBERTa-Sentinel (Chen et al., 2023), a frozen RoBERTa encoder with a two-layer GELU (Hendrycks & Gimpel, 2016) MLP head. We made every backbone choice, ours included, on the evaluation splits with the same procedure. Backbone sizes differ by construction: the perplexity-based detectors need the input's per-token log-probabilities and run only on open-weight models, whereas LogJudge can use a commercial API model. To separate the effect of the readout from that of the backbone, we always report the same judge's parsed score (BaseJudge) and repeat the comparison with the open-weight judge.

Main comparison. Table 2 and Figure 3 compare all detectors on the four splits (per-domain values in Appendix L, score distributions of every backbone in Appendix M). Changing only how we read the verdict turns the same judge into a better detector: with `gpt-5.6-luna`, the four-split mean AT-HM rises from 0.653 for the parsed readout (BaseJudge) to 0.767 for the log-probability readout (LogJudge), ahead of Fast-DetectGPT (0.636) and Binoculars (0.628). The two readouts are also complementary. BaseJudge is stronger on MG-Sci (0.866 versus 0.794), LogJudge on GB-orig (0.883 versus 0.692), MG-GB (0.687 versus 0.534), and MG-AI-Sci (0.692 versus 0.462), and their unweighted average, FusionJudge, reaches an AT-HM of 0.848 and a TPR@1%FPR of 0.761, 0.081 and 0.195 AT-HM above its two members.

Split by split. With `gpt-5.6-luna`, LogJudge and FusionJudge beat both perplexity-based baselines on GB-orig, MG-Sci, and MG-AI-Sci, and their lead holds without the small MG-AI-Sci and with splits weighted by size (Appendix L). The exception is MG-GB (0.687 and 0.776 versus 0.892 for Fast-DetectGPT and 0.897 for Binoculars), partly through threshold transfer rather than ranking: within each MG-GB domain LogJudge catches 60–100% of machine texts at 1% FPR, but one threshold across the three domains catches only 55%. The backbone decides which splits a judge handles: with `gemma-4-31b`, LogJudge is the best detector on GB-orig and MG-GB (0.989 and 0.900), closing that gap, but falls to 0.284 on MG-Sci and 0.058 on MG-AI-Sci.

Why the readout matters, and why averaging helps. Changing only the readout barely moves AUROC (by at most 0.033 on any split) but raises the four-split mean TPR@1%FPR by 0.145 for `gpt-5.6-luna` and by 0.128 for `gemma-4-31b` (Table 2, per split in Appendix P). The reason

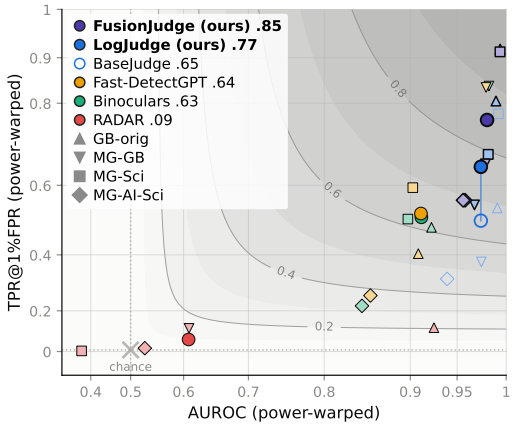


Figure 3: AUROC against TPR@1%FPR for every detector (color) and split (marker). Large circle markers are four-split means, gray curves are AT-HM iso-lines, and both axes are power-warped to spread the high-performance corner. The vertical segment joins BaseJudge and LogJudge for `gpt-5.6-luna`: equal AUROC, TPR@1%FPR 0.503 to 0.648.

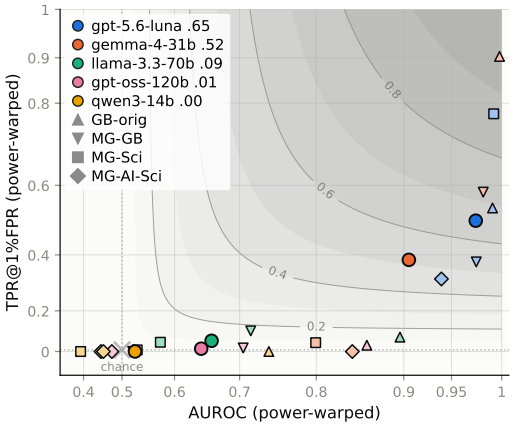


Figure 4: Parsed-judge backbones. AUROC against TPR@1%FPR for five instruction-tuned models used as parsed judges (BaseJudge), same axes as the left panel. Only `gpt-5.6-luna` and `gemma-4-31b` clear chance by a useful margin.

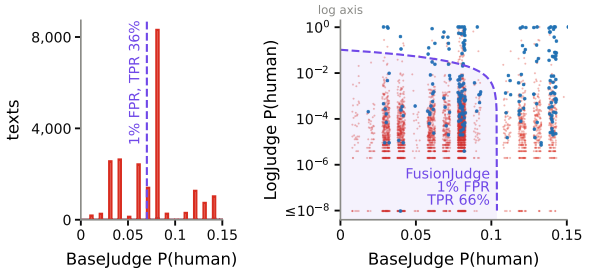


Figure 5: **Why the readout matters, and why averaging helps.** The same `gpt-5.6-luna` judge, all splits pooled. Machine texts are red, human texts blue, and 1%-FPR thresholds dashed. *Left:* the written probability uses only a few values, so thousands of texts tie at 0.08 and the threshold must stop before them. *Right:* each dot is one text, and its log-probability score (y axis, log scale for display only) spreads every tied value apart. FusionJudge flags everything below the curve, which also spares human texts wrongly written as machine-like (upper left) and catches machine texts written just above the threshold (lower right).

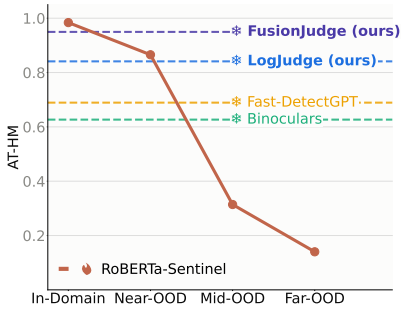


Figure 6: **Zero-shot versus trained under domain shift.** AT-HM of RoBERTa-Sentinel (🔥) in four training scenarios of increasing distance from the MG-Sci evaluation slices, against the zero-shot detectors (❄), which have no training set and are flat by construction. Values are averaged over the three MG-Sci domains.

is resolution (Figure 5, left): the stated probability piles up on a few round values, so many machine texts tie with the most machine-like human texts and the 1%-FPR threshold must stop before them, whereas the log-probability score is continuous. AUROC averages over all thresholds and barely registers these ties, but TPR@1%FPR depends on them. The two readouts therefore complement each other, and this is why their plain average, FusionJudge, works (Figure 5, right). Within each stated value, the log-probability breaks the ties. Across values, a confident log-probability lifts human texts the judge wrongly stated as machine-like and pulls in machine texts stated just above the threshold. Tie-breaking alone gives most of the gain (AT-HM 0.653 → 0.829, against 0.848 for FusionJudge, Appendix O).

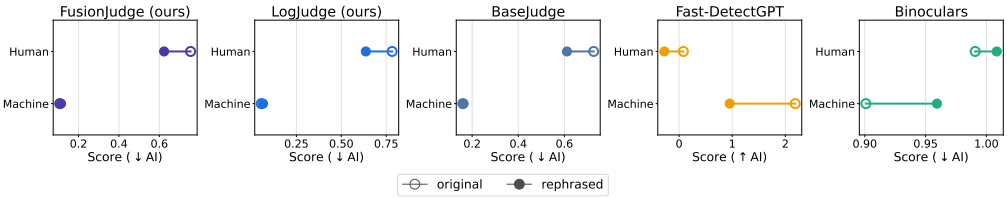


Figure 7: **Center of mass of each class**, pooled over Manugram-Ghostbuster and Manugram-Sci, per scorer. Hollow marker = original mean, filled = rephrased. By design, for Fast-DetectGPT higher scores mean machine-generated, while for others higher score indicates human-written text.

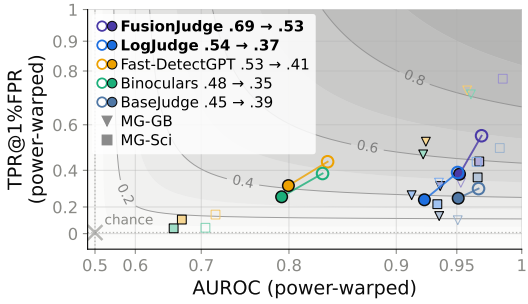


Figure 8: **With and without Rephrasing.** AUROC vs. TPR@1%FPR for human vs. `gpt-5.6-luna` text on MG-GB/MG-Sci, originals only (hollow) vs. with rephrased versions of both classes added (filled). Large markers are two-split means. FusionJudge stays strongest.

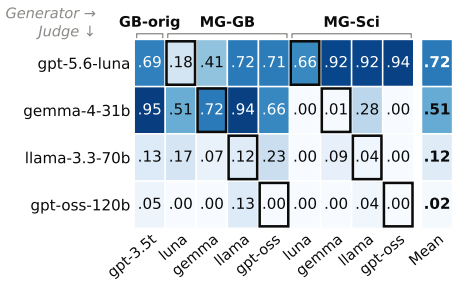


Figure 9: **Judge–generator pairing** (parsed readout): AT-HM of human against one generator's text. Outlined cells mark judge = generator. `gpt-5.6-luna` text is hardest for both judges, indicating no self-recognition inflation.

Zero-shot versus trained under domain shift. A trained detector leads only as long as its training data resemble the test domain, whereas a zero-shot detector has no training data and its score stays the same. We train RoBERTa-Sentinel (RoBERTa is the strongest of 12 embeddings for this task, see Appendix Q) and score it on held-out 30% slices of each MG-Sci domain (computer science, mathematics, and sociology, Table 1). We use four training sets of increasing distance from the test domain (Figure 6). *In-Domain* uses the other two domains plus the held-out domain's own 70%. *Near-OOD* (out-of-distribution) uses the other two domains only. *Mid-OOD* uses all of MG-GB, that is, news, essays, and fiction instead of abstracts. *Far-OOD* uses GB-orig, the same three text types written by a legacy generator. Each zero-shot detector is scored once on the same slices, and all results are averaged over the three domains. RADAR is below chance here and is omitted. Appendix R shows that a logistic head collapses like the MLP head. RoBERTa-Sentinel starts near ceiling (AT-HM 0.984) and falls to 0.865, 0.314, and 0.140, a seven-fold drop. Once its falling line crosses a zero-shot detector's flat one, the zero-shot detector is the better choice. FusionJudge (0.950) is beaten only by In-Domain training. A logistic head on the same RoBERTa features is stronger than the MLP head and also beats FusionJudge at Near-OOD (0.975), but it collapses in the same way further out (0.391 and 0.213). LogJudge (0.841) is just below RoBERTa-Sentinel's Near-OOD value and leads from Mid-OOD onward, and so do Fast-DetectGPT (0.689) and Binoculars (0.627).

Rephrasing: two threat models. Rephrasing hurts the two detector families in opposite ways (Figure 7). When an adversary rephrases machine text (*evasion*), perplexity-based detectors lose it: Fast-DetectGPT's AUROC falls from 0.958 to 0.860 on MG-GB (Appendix S), while the judges' scores barely move. When an author polishes human text with an LLM (*collateral flagging*), the judges instead push it toward machine and flag more human authors, whereas perplexity-based detectors do not. With both kinds of rephrased text added, every detector loses accuracy, and FusionJudge stays the strongest (AT-HM drops from 0.69 to 0.53, Figure 8). This test uses only `gpt-5.6-luna` text, the hardest generator for both judges. A deployment that must never flag LLM-polished human writing should therefore prefer a perplexity-based detector.

Self-recognition. A judge whose model is also one of the generators could score well simply by recognizing its own model's text. To test this, we compute AT-HM for every judge–generator pair

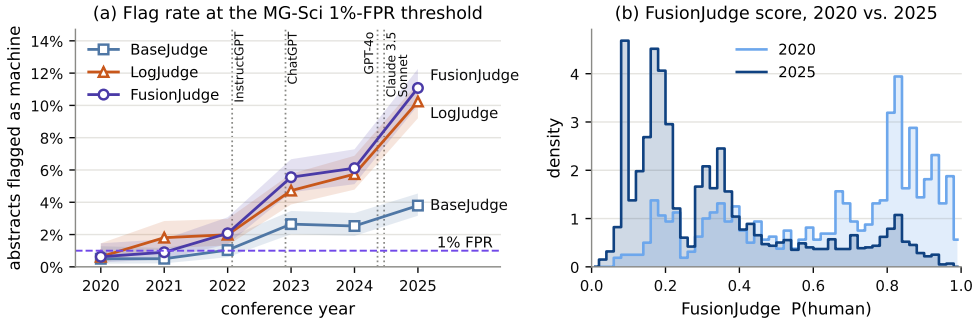


Figure 10: **ICLR abstracts, 2020–2025**, scored by the `gpt-5.6-luna` judge ($n = 9{,}754$ after filtering). (a) Share of abstracts flagged as machine at each detector's 1%-FPR threshold set on the MG-Sci human abstracts. Shaded bands are 95% Wilson intervals, the dashed line is the nominal 1% FPR, and the dotted lines mark, at their calendar dates (each tick is January of that year), InstructGPT becoming the OpenAI API default (27 January 2022) and the releases of ChatGPT (30 November 2022), GPT-4o (13 May 2024), and Claude 3.5 Sonnet (20 June 2024). ICLR abstracts of year Y are submitted in autumn of $Y-1$ and finalized in early Y. (b) FusionJudge $P(\text{human})$ densities for 2020 and 2025. Lower scores are more machine-like.

with the parsed readout (Figure 9, details in Appendix T). The `gpt-5.6-luna` judge separates its own generations from human text *worse* than any other generator's (0.18 versus 0.41–0.72 on MG-GB, and 0.66 versus 0.92–0.94 on MG-Sci), and `gpt-5.6-luna` text is also the hardest generator for the `gemma-4-31b` judge (0.51 versus 0.66–0.94 on MG-GB). Self-recognition therefore does not inflate the headline results. The strongest generator is simply the hardest case for every detector, its own judge included, and this explains part of the MG-GB loss.

In the wild: ICLR abstracts, 2020–2025. We run the `gpt-5.6-luna` judges on 9,754 ICLR abstracts from 2020–2025, all tracks included, with each detector's threshold fixed in advance at its 1%-FPR value on the MG-Sci human abstracts (Figure 10). On the 2020 abstracts every detector flags 0.5–0.6%, within the nominal 1% rate. FusionJudge then flags 2.1% of the 2022, 5.6% of the 2023, and 11.1% of the 2025 abstracts. On the same 2025 abstracts, BaseJudge flags only 3.8%. A flag rate is not the share of LLM-written abstracts. Appendix U gives the data, the full table, and the caveats.

Limitations. Our detectors need a judge API that returns top-k log-probabilities, and FusionJudge makes two calls per document. Their accuracy depends on the judge: two of the five backbones we tried are strong enough, and the open one is weaker on scientific abstracts. The AI-Scientist split is small (100 human texts). Our evaluation covers English text and one attack, rephrasing. A quarter of the machine text comes from the main judge's own model, although we find no self-recognition, and judges, as other LMs, may have seen some public human texts in pretraining, which would affect both readouts alike. Manugram is built to be regenerated as generators change.

6 Conclusion

Before an LLM judge writes "machine" or a rounded probability, its answer-token distribution already holds a finer verdict, and under a strict false-alarm budget this difference matters. With the same low-cost commercial model as judge, LogJudge leaves ranking quality unchanged but raises recall at 1% FPR from 50% to 65%, because the written answer produces numerous ties. The two readouts complement each other. The written answer orders texts coarsely, and the log-probabilities resolve its ties and correct it where they are confident. Their plain average, FusionJudge, catches 76% of machine texts, against 52% for Fast-DetectGPT and 51% for Binoculars. Neither detector needs training, so their accuracy holds when the topic shifts. Both read only the few probabilities that commercial APIs already return, so strong API models can be used as detectors directly. Manugram, released with every detector's scores, allows such comparisons on current generators, scientific abstracts, and AI-Scientist systems, and it can be regenerated when new generators appear. Rephrasing still hurts both detector families, in opposite ways, and combining their signals is an interesting direction for future work.

AI USE STATEMENT

Generative AI is a subject of this paper's methodology, not only a tool used to prepare it, so we disclose the two separately.

As part of the study. Large language models (`gpt-5.6-luna`, `gpt-oss-120b`, `gemma-4-31b`, `llama-3.3-70b`, plus additional open-weight models used only in the backbone-selection sweep, Appendix K) generated every machine-authored document in the Manugram suite and act as the judge underlying LogJudge and FusionJudge. Here the models are what we study, and they did not help write the paper. Sections 3 and 4 document it, and Appendices C and F–I give the exact prompts.

In preparing this work. We used generative AI tools to help implement methods, using a coding assistant (Claude Code) for the data-generation, data-cleaning, training, scoring, and plotting pipelines; to discuss research ideas and give feedback on experiment designs; to help interpret results; for retrieval and discovery, such as finding related work; and to draft and polish portions of this manuscript. We have not used generative AI tools to propose hypotheses or to formulate claims, and translation is not applicable to this work. We have reviewed all AI-assisted work. The authors checked every retrieved reference against its source and made all research decisions. Our own pipelines computed every reported number from the underlying score files, and the authors checked all AI-assisted text against those results and edited it. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

A note on this paper's own text. Parts of this manuscript are human writing polished with an LLM, the kind of text whose collateral flagging we measure in Section 5. This points to a distinction that detection has yet to make, between human work polished with an LLM and content an LLM generated, as in our AI-Scientist abstracts (MG-AI-Sci, Section 4). Current detectors, ours included, do not yet separate the two well enough, and we hope this work helps close that gap.

ETHICS STATEMENT

Risk of false accusation. A detector's error falls on a person: a flagged human author can be accused of misconduct. This is why we evaluate every detector at a fixed 1% false-alarm rate. Even at that rate, one human text in a hundred is flagged, detectors flag non-native English writing more often (Liang et al., 2023), and human text polished with an LLM moves toward the machine side (Section 5). No detector in this paper, ours included, should serve as the sole evidence against an individual.

ICLR abstracts. The analysis in Section 5 scores abstracts of identifiable, published papers. We report only aggregate flag rates per year and do not release the per-abstract texts or scores, so that no paper or author is singled out. A flag is not evidence that a given abstract was machine-written (Appendix U).

Released data. The human texts in Manugram come from the existing Ghostbuster benchmark (Verma et al., 2024), from published abstracts indexed by OpenAlex (Priem et al., 2022), and from abstracts released by AI-Scientist systems. The machine-written scientific abstracts report invented results by design (Section 4); they are labeled as machine-written and are meant for detection research only.

REPRODUCIBILITY STATEMENT

All code[3] and the Manugram texts with every detector's scores[4] are available online. *Reproducing the paper:* every table and figure on the Manugram benchmarks is recomputed from the released data on a CPU, with no API key or GPU, and every reported number is also committed as CSV. The ICLR-abstract analysis (Section 5) is the exception: we do not release its per-abstract data (see the ethics statement). *Rebuilding or extending the benchmark:* the pipeline that generated and scored the corpora is included, with a checker that compares re-computed scores with the released ones.

[3] `https://anonymous.4open.science/r/logjudge-manugram-D902`

[4] `https://osf.io/gsbtz/overview?view_only=592bbbf2dcca47acab265ed57f591f02`

This step needs API keys and a GPU. The exact judge and generation prompts are in Appendices C–I. Section 5 and Appendix J define the metrics and the 1%-FPR thresholds. Appendix V measures how much API log-probabilities change between repeated calls.

REFERENCES

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The Falcon Series of Open Language Models, November 2023. URL http://arxiv.org/abs/2311.16867. arXiv:2311.16867 [cs.CL].

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Bpcgcr8E8Z.

Guangsheng Bao, Yanbin Zhao, Juncai He, and Yue Zhang. Glimpse: Enabling white-box methods to use proprietary models for zero-shot LLM-generated text detection. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=an3fugFA23.

Amrita Bhattacharjee and Huan Liu. Fighting Fire with Fire: Can ChatGPT Detect AI-generated Text? *ACM SIGKDD Explorations Newsletter*, 25(2):14–21, March 2024. ISSN 1931-0145. doi: 10.1145/3655103.3655106. URL https://dl.acm.org/doi/10.1145/3655103.3655106.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. GPT-Sentinel: Distinguishing Human and ChatGPT Generated Content, May 2023. URL https://arxiv.org/abs/2305.07969.

Zheng Chen, Yushi Feng, Jisheng Dang, Changyang He, Yue Deng, Hongxi Pu, Haoxuan Li, and Bo Li. IPAD: Inverse prompt for AI detection - a robust and interpretable LLM-generated text detector. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 167441–167467. Curran Associates, Inc., 2025. doi: 10.52202/085713-5580. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f4d6932b6b9eec9b9d595e2847d095c1-Paper-Conference.pdf.

David F. Crouse. On implementing 2D rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696, August 2016. ISSN 1557-9603. doi: 10.1109/TAES.2016.140952. URL https://ieeexplore.ieee.org/document/7738348.

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, October 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-08025-4. URL https://www.nature.com/articles/s41586-024-08025-4.

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, 2024.

Charles Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'01, pp. 973–978, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608125.

Catherine A. Gao, Frederick M. Howard, Nikolay S. Markov, Emma C. Dyer, Siddhi Ramesh, Yuan Luo, and Alexander T. Pearson. Comparing scientific abstracts generated by ChatGPT to real abstracts with detectors and blinded human reviewers. *npj Digital Medicine*, 6(1):75, April 2023. ISSN 2398-6352. doi: 10.1038/s41746-023-00819-6. URL https://doi.org/10.1038/s41746-023-00819-6.

Gemma Team. Gemma 4 Technical Report, July 2026. URL http://arxiv.org/abs/2607.02770. arXiv:2607.02770 [cs.CL].

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://doi.org/10.1198/016214506000001437.

Aaron Grattafiori et al. The Llama 3 Herd of Models, November 2024. URL http://arxiv.org/abs/2407.21783. arXiv:2407.21783 [cs.AI].

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection, January 2023. URL https://arxiv.org/abs/2301.07597.

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 Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17519–17537. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hans24a.html.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. MGTBench: Benchmarking machine-generated text detection. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, pp. 2251–2265, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706363. doi: 10.1145/3658644.3670344. URL https://doi.org/10.1145/3658644.3670344.

Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs), June 2016. URL https://arxiv.org/abs/1606.08415.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. RADAR: Robust AI-Text Detection via Adversarial Learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 15077–15095. Curran Associates, Inc., 2023. doi: 10.52202/075280-0662. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/30e15e5941ae0cdab7ef58cc8d59a4ca-Abstract-Conference.html.

Intology. Zochi technical report. Technical report, Intology, 2025. https://www.intology.ai/blog/zochi-tech-report.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. URL https://arxiv.org/abs/2207.05221.

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*, pp. 17061–17084. PMLR, July 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DEJIDCmWOz.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. OUTFOX: LLM-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38 of *AAAI'24/IAAI'24/EAAI'24*. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i19.30120. URL https://doi.org/10.1609/aaai.v38i19.30120.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems*, volume 36, Red Hook, NY, USA, 2023. Curran Associates Inc.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=FpaCL1MO2C.

H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, March 1955. ISSN 1931-9193. doi: 10.1002/nav.3800020109. URL https://onlinelibrary.wiley.com/doi/10.1002/nav.3800020109.

Lucio La Cava and Andrea Tagarelli. Luminol-AIDetect: Fast Zero-shot Machine-Generated Text Detection based on Perplexity under Text Shuffling. April 2026. doi: 10.48550/arXiv.2604.25860. URL http://arxiv.org/abs/2604.25860. arXiv:2604.25860 [cs.CL].

Linyang Li, Pengyu Wang, Ke Ren, Tianxiang Sun, and Xipeng Qiu. Origin tracing and detecting of LLMs. *arXiv preprint arXiv:2304.14072*, 2023. URL https://arxiv.org/abs/2304.14072.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36–53, 2024.

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. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2023.100779. URL https://www.sciencedirect.com/science/article/pii/S2666389923001307.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=8s8K2UZGTZ.

Runheng Liu, Heyan Huang, Xingchen Xiao, and Zhijing Wu. Zero-shot detection of LLM-generated text via implicit reward model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2VdsYVXLDl.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2511–2522. Association for Computational Linguistics, 2023.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery, September 2024. URL http://arxiv.org/abs/2408.06292. arXiv:2408.06292 [cs.AI].

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*, AAAI'26/IAAI'26/EAAI'26. AAAI Press, 2026. ISBN 978-1-57735-906-7. doi: 10.1609/aaai.v40i38.40510. URL https://doi.org/10.1609/aaai.v40i38.40510.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. Raidar: generative AI detection via rewriting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=bQWE2UqXmf.

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*, pp. 24950–24962. PMLR, July 2023. URL https://proceedings.mlr.press/v202/mitchell23a.html.

OpenAI. gpt-oss-120b & gpt-oss-20b Model Card, August 2025. URL https://arxiv.org/abs/2508.10925.

OpenAI. GPT-5.6 Luna. https://developers.openai.com/api/docs/models/gpt-5.6-luna, 2026. Large language model, version dated 2026-07-09.

Mehmet Batuhan Özdaş and Murat Osmanoğlu. DWT-Fusion: A Signal-Based Framework for Training-Free LLM-Generated Text Detection. July 2026. doi: 10.48550/arXiv.2607.22026. URL http://arxiv.org/abs/2607.22026. arXiv:2607.22026 [cs.CL].

Jason Priem, Heather Piwowar, and Richard Orr. OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts, June 2022. URL http://arxiv.org/abs/2205.01833. arXiv:2205.01833 [cs.DL].

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can AI-generated text be reliably detected? Stress testing AI text detectors under various attacks. *Transactions on Machine Learning Research*, 2025.

Amanpreet Singh, Mike D'Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. SciRepEval: A Multi-Format Benchmark for Scientific Document Representations. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5548–5566, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.338. URL https://aclanthology.org/2023.emnlp-main.338/.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. Release Strategies and the Social Impacts of Language Models, August 2019. URL https://arxiv.org/abs/1908.09203.

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 12395–12412, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.827. URL https://aclanthology.org/2023.findings-emnlp.827/.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. Authorship Attribution for Neural Text Generation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8384–8395, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.673. URL https://aclanthology.org/2020.emnlp-main.673/.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. Ghostbuster: Detecting Text Ghostwritten by Large Language Models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1702–1717, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.95. URL https://aclanthology.org/2024.naacl-long.95/.

Francisco Villaescusa-Navarro et al. The denario project: Deep knowledge ai agents for scientific discovery. *arXiv preprint arXiv:2510.26887*, 2025. URL https://arxiv.org/abs/2510.26887.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. SeqXGPT: Sentence-Level AI-Generated Text Detection. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1144–1156, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.emnlp-main.73/.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohanned 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, 2024a.

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

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. CycleResearcher: Improving automated research via automated review. In *The Thirteenth International Conference on Learning Representations*, 2025.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. DetectRL: Benchmarking LLM-generated text detection in real-world scenarios. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2024.

Junxi Wu, Kailin Huang, Dongjian Hu, Bin Chen, Hao Wu, Shu-Tao Xia, and Changliang Zou. Alignment Imprint: Zero-Shot AI-Generated Text Detection via Provable Preference Discrepancy. April 2026. doi: 10.48550/arXiv.2604.16923. URL http://arxiv.org/abs/2604.16923. arXiv:2604.16923 [cs.AI].

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. LLMDet: A Third Party Large Language Models Generated Text Detection Tool. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2113–2133, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.139. URL https://aclanthology.org/2023.findings-emnlp.139/.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations*, 2024.

Yihuai Xu, Yongwei Wang, Yifei Bi, Huangsen Cao, Zhouhan Lin, Yu Zhao, and Fei Wu. Training-free LLM-generated text detection by mining token probability sequences. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=vo4AHjowKi.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search, April 2025. URL http://arxiv.org/abs/2504.08066. arXiv:2504.08066 [cs.AI].

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. DNA-GPT: Divergent n-gram analysis for training-free detection of GPT-generated text. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Xlayxj2fWp.

Peipeng Yu, Jiahan Chen, Xuan Feng, and Zhihua Xia. CHEAT: A large-scale dataset for detecting CHatGPT-written AbsTracts. *IEEE Transactions on Big Data*, 11(3):898–906, 2025.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending Against Neural Fake News. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/3e9f0fc9b2f89e043bc6233994dfcf76-Abstract.html.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 12697–12706. PMLR, 2021.

Hongyi Zhou, Jin Zhu, Pingfan Su, Kai Ye, Ying Yang, Shakeel Gavioli-Akilagun, and Chengchun Shi. AdaDetectGPT: Adaptive detection of LLM-generated text with statistical guarantees. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=3Qo2SRcHgU.

Biru Zhu, Lifan Yuan, Ganqu Cui, Yangyi Chen, Chong Fu, Bingxiang He, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. Beat LLMs at Their Own Game: Zero-Shot LLM-Generated Text Detection via Querying ChatGPT. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7470–7483, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.463. URL https://aclanthology.org/2023.emnlp-main.463/.

A Detector Taxonomy

Table 3 classifies the methods of Section 2 by what they need at detection time, whether they are trained, whether their signal reflects the whole text's meaning, and whether it rests on log-probabilities. LogJudge and FusionJudge are the only training-free entries that combine the holistic signal of a prompted judge with a log-probability readout. IPAD reads a similar yes/no probability, but from distinguishers fine-tuned on labeled data. Apart from IPAD and LLMDet (whose n-gram tables are precomputed offline), they need less access than any other log-probability method: the top-k log-probabilities of one generated token rather than per-token log-probabilities of the input.

B Benchmark Comparison

Table 4 compares Manugram with multi-domain machine-generated text benchmarks. Manugram is the only one that labels LLM-rephrased human text as its own class and the only one with abstracts written by AI-Scientist systems.

C Judge Prompts and API Settings

Both LogJudge and BaseJudge use a fixed system prompt followed by the document wrapped in a fixed user template, $\mathrm{doc}(x) =$ `<document>` x `</document>` (Appendix D). LogJudge decodes one token at temperature 0 and records the top-k log-probabilities at that position. BaseJudge lets the judge answer in text and parses the stated probability. `gpt-5.6-luna` is queried through the OpenAI Responses API and `gemma-4-31b` through an institutional inference endpoint. Both judges use $k = 20$, the largest window either endpoint returns. `gpt-5.6-luna` is called with `top_logprobs=20`, `include=["message.output_text.logprobs"]`, and reasoning effort `none`. `gemma-4-31b` uses the OpenAI-compatible chat-completions interface with `logprobs=true` and `top_logprobs=20`. The window is read at the first generated token.

All three system prompts share the same evidence section (below) and differ only in the block appended after it, which states the answer to give. Each system prompt is the evidence section, a blank line, and one of the three answer blocks that follow it.

Table 3: Detection methods by mechanism, access requirement, training, whether the signal reflects the whole text's meaning, and whether it uses log-probabilities or perplexity.

Family	Method	Acc	TF	Hol	LP	Mechanism
Watermark	Green/red list (Kirchenbauer et al., 2023)	○	●	○	○	Biased green-token sampling
	Distortion-free (Kuditipudi et al., 2024)	○	●	○	○	Key-seeded token selection
	SynthID-Text (Dathathri et al., 2024)	○	●	○	○	Tournament sampling, at scale
Training-based	GPT-2 detector (Solaiman et al., 2019)	●	○	●	○	Fine-tune RoBERTa end-to-end
	Grover (Zellers et al., 2019)	●	○	●	○	Generator as its own discriminator
	Authorship (Uchendu et al., 2020)	●	○	●	○	Multi-class source attribution
	HC3 (Guo et al., 2023)	●	○	●	○	Classifier on ChatGPT/human pairs
	RADAR (Hu et al., 2023)	●	○	●	○	Adversarial co-train vs. paraphraser
	IPAD (Chen et al., 2025)	◕	○	●	●	Invert prompt, fine-tuned yes/no distinguishers
	GPT-Sentinel (Chen et al., 2023)[1]	●	◑	●	○	MLP head on frozen RoBERTa
	LLMDet (Wu et al., 2023)[2]	●	◑	○	●	Proxy perplexity from precomputed n-gram tables
	Ghostbuster (Verma et al., 2024)	◑	◑	○	●	Classifier over weak-LM features
	SeqXGPT (Wang et al., 2023)	◑	◑	○	●	Network over token log-prob lists
	Raidar (Mao et al., 2024)	●	◑	●	○	LLM rewrites, classifier on edit distance
Zero-shot	DetectGPT (Mitchell et al., 2023)	◑	●	○	●	Log-prob curvature under perturbation
	DetectLLM (Su et al., 2023)	◑	●	○	●	Log-likelihood / log-rank ratio
	Fast-DetectGPT (Bao et al., 2024)	◑	●	○	●	Conditional curvature, one pass
	Binoculars (Hans et al., 2024)	◑	●	○	●	Perplexity / cross-perplexity ratio
	Lastde/Lastde++ (Xu et al., 2025)	◑	●	○	●	Time-series stats of token-probs
	SpecDetect (Luo et al., 2026)	◑	●	○	●	Spectral energy of token log-probs
	DWT-Fusion (Özdaş & Osmanoğlu, 2026)	◑	●	○	●	Wavelet statistics of token log-probs
	Luminol-AIDetect (La Cava & Tagarelli, 2026)	◑	◑	○	●	Perplexity under sentence shuffling
	IRM (Liu et al., 2025)	◑	●	○	●	Instruct/base log-ratio as implicit reward
	LAPD (Wu et al., 2026)	◑	●	○	●	Information-weighted instruct/base log-ratio
	AdaDetectGPT (Zhou et al., 2025)	◑	◑	○	●	Learned witness over token log-probs
	Glimpse (Bao et al., 2025)[3]	◑	●	○	●	Fast-DetectGPT on API top-k estimates
	LogJudge (ours)	◕	●	●	●	**Verdict read from the judge's answer-token log-probabilities**
	FusionJudge (ours)	◕	●	●	●	**Average of LogJudge and the same judge's parsed probability**
	DNA-GPT (Yang et al., 2024)[4]	●	●	○	○	LLM regenerates ending, n-gram overlap
	Query-and-revise (Zhu et al., 2023)	●	●	●	○	LLM revises, compare similarity
	LLM-as-judge (Bhattacharjee & Liu, 2024)	●	●	●	○	Prompt LLM for a direct verdict

Acc = access, the method needs: ● only the text, ◕ only the top-k log-probabilities of one answer token, ◑ full per-token log-probabilities, ○ the secret key (given controlled generation). **TF** = training-free: ● no training, ◑ base LM frozen and a lightweight head is trained, ○ training required. **Hol** = holistic: ● signal reflects the whole text's meaning, ○ signal is token/word-level only. **LP** = uses log-probabilities or perplexity as its core signal: ● yes, ○ no.

[1]The authors also propose a T5-Sentinel variant that fine-tunes the full model. Only the RoBERTa-Sentinel variant is feature-based. [2]The n-gram tables are built offline by querying each candidate LLM. Detection then needs only the text and the tables. [3]Needs only the top-k log-probabilities an API returns, but at every input token (a completion API that echoes the input), and estimates the full distribution from them. [4]Black-box variant. The white-box variant compares the continuations' log-probabilities instead and needs full per-token log-probabilities.

Shared evidence section (all three system prompts)

```
You are a forensic linguist estimating whether a document was written
by a human or generated by a large language model.
You will be shown one document and nothing else:  no title, no author,
no source, no metadata, no context.  Judge it on its own evidence.
JUDGE RELATIVE TO THE GENRE
Work out what kind of document this is trying to be, then ask whether
it reads like a competent human working in that genre or like a
model imitating one.  A wire report carries no personal voice and a
scientific abstract carries no digressions; in those genres that is
the convention, not evidence of a machine.  The question is always
``unusual for this kind of document?'', never ``unusual in general?''.
```

Table 4: Multi-domain machine-generated text benchmarks: how much of the core domain set each one covers, whether scientific text is among its domains, whether it contains rewritten rather than freshly generated text, and whether it includes text written by autonomous AI-scientist systems.

Benchmark	Dom	Sci	Rew	AIS	Domains covered	Backbones
DetectGPT (Mitchell et al., 2023)	◑	○	○	○	news, QA, creative, biomedical	GPT-2 (1.5B), GPT-Neo-2.7B, GPT-J
HC3 (Guo et al., 2023)	◔	○	○	○	QA, biomedical, legal/finance, encyclopedic	GPT-2
M4 (Wang et al., 2024b)	◑	●	○	○	encyclopedic, QA, scientific, news	GPT4, Cohere, Dolly2, BLOOMz
MGTBench (He et al., 2024)	◑	○	◑	○	essays, creative, news	ChatGLM, Dolly, ChatGPT-turbo, GPT4All, StableLM, Claude
Ghostbuster (Verma et al., 2024)	◑	○	○	○	news, creative, essays	gpt-3.5-turbo, Claude
SeqXGPT-Bench (Wang et al., 2023)	◑	●	○	○	news, reviews, web, scientific, QA	GPT-2, GPT-J, GPT-Neo, LLaMA, GPT-3.5-turbo
SnifferBench (Li et al., 2023)	◑	●	○	○	news, reviews, web, scientific, QA	GPT2, GPT-Neo/J, LLaMA
MAGE (Li et al., 2024)	◕	●	◑	○	opinion, news, QA, creative, scientific, encyclopedic	gpt-turbo-3.5, GLM-130B, GPT-J-6B, GPT-NeoX-20B etc.
RAID (Dugan et al., 2024)	◕	●	◑	○	scientific, news, creative, QA, reviews, encyclopedic, recipes	GPT2-3-4, Mistral 7B, MPT 30B, LLaMA 2
Raidar (Mao et al., 2024)	◕	●	◑	○	news, creative, essays, code, reviews, scientific	GPT-3.5
M4GT-Bench (Wang et al., 2024a)	◕	●	○	○	encyclopedic, QA, scientific, news, essays	ChatGPT, GPT-4, and LLaMA-2, BLOOMz
DetectRL (Wu et al., 2024)	◑	●	◑	○	scientific, news, creative, reviews	GPT-3.5-turbo, PaLM-2-bison, Claude-instant, Llama-2-70b
Manugram suite (ours)	◕	●	●	●	**news, creative, essays, scientific**	gpt-5.6-luna, gpt-oss-120b, gemma-4-31b, llama-3.3-70b

Dom = share of the five core domains (news, scientific writing, question answering, creative writing, student essays) the benchmark covers, shown as a filled fraction of the circle. **Sci** = scientific text among the domains: ● yes, ○ no. **Rew** = rewritten text: ● LLM-rewritten human text is a labelled class, ◑ rewriting appears only as an attack on detectors, ○ text is generated *de novo* throughout. **AIS** = includes labelled text produced by autonomous AI-scientist systems (Villaescusa-Navarro et al., 2025; Lu et al., 2024; Yamada et al., 2025; Weng et al., 2025; Intology, 2025): ● yes, ○ no.

```
EVIDENCE THAT TENDS TO INDICATE HUMAN AUTHORSHIP
- Specifics a model could not reconstruct from general knowledge:
particular names, numbers, dates, places, results, quoted words --
load-bearing rather than decorative, so that removing one would damage
the text.
- Commitment:  a definite claim, an unbalanced opinion, an argument
that goes somewhere instead of hedging back to neutral.
- Asymmetry:  uneven emphasis, a paragraph that runs long because the
writer cared about it, a point raised and dropped, a sentence that
changes direction mid-way.
- Idiom, slang, humour, irony, in-group shorthand; the habits of a
field used the way a practitioner uses them rather than the way they
would be explained.
- Errors of language:  a misspelling, a wrong-word substitution,
an agreement or tense slip, a missing word, a run-on or fragmentary
sentence, a construction that does not resolve.
EVIDENCE THAT TENDS TO INDICATE MACHINE AUTHORSHIP
- Fluent but generic vocabulary; an abstraction where a specific would
have been natural.
- Even, predictable rhythm:  uniform sentence and paragraph lengths,
repetitive clause structure.
- Tidy scaffolding:  symmetrical structure, explicit enumeration, a
```

```
closing sentence that restates what came before and adds nothing.
- Hedged, balanced framing that declines to commit:  every side
acknowledged, none pressed.
- Coverage in place of depth:  each obvious aspect of the topic given
a proportionate share, nothing pursued further than the rest.
- The connectives, intensifiers and stock transitions of assistant
prose, and the reach for a summarising abstraction where the writing
could have stopped.
WHAT IS NEVER EVIDENCE
- Length.  A short document is harder to judge, not more likely to be
machine-written.
- Polish.  Careful, edited human writing is fluent and correct.
- Subject matter.  Any topic can be written about by either.
- Damage that is not the author's doing:  a stranded fragment, broken
encoding, lost markup, a document that begins or ends abruptly or
stops mid-sentence.  Judge the writing, not the state the document
reached you in.
```

LogJudge answer block with binary scheme (`ord2`)

```
ANSWER
Answer with exactly one word, and nothing else:  `human` if the
document was more likely written by a human, `machine` if it was
more likely generated by a large language model.  No probability,
no explanation, no punctuation.
Judge each document on its own.  Do not assume any particular mix of
human and machine documents, and do not try to balance your answers
across documents.
```

LogJudge answer block with ordinal scheme (`ord5`)

```
ANSWER
Answer with exactly one letter, and nothing else, rating whether
the document was written by a human or generated by a large language
model:
A  almost certainly written by a human
B  probably written by a human
C  genuinely uncertain
D  probably generated by a large language model
E  almost certainly generated by a large language model
No explanation, no punctuation.  Judge each document on its own; do
not assume any particular mix of human and machine documents.
```

BaseJudge answer block (text-stated probability)

```
CALIBRATION
- Return a probability that the document is HUMAN-written.  1.0 means
certainly human, 0.0 means certainly machine, 0.5 means the text
carries no usable evidence either way.
- Use the whole range, and give two decimal places.  Do not settle on
round values such as 0.20, 0.50 or 0.80:  if your belief falls between
two of them, say so (0.37, 0.62).  These scores are ranked against
each other, so every tie is information thrown away.
- Reserve values below 0.10 and above 0.90 for documents where the
evidence is genuinely strong.
- Do not assume any particular mix of human and machine documents, and
do not try to balance your answers across documents.  Judge each one
on its own.
Reply with the probability only.
```

D LogJudge: General Formulation

This appendix gives the full derivation summarized in Section 3.

Setup. Let J be a judge LLM with next-token distribution $q_J(\cdot \mid c)$ over its vocabulary $\mathcal{V}$, given a context c. For a document x we build the context $c(x) = \mathrm{sys}(\mathcal{L}) \oplus \mathrm{doc}(x)$: a fixed system prompt $\mathrm{sys}(\mathcal{L})$ that states the detection task and the required answer format for a label set $\mathcal{L}$ (the first component of the answer scheme defined next), concatenated with the document wrapped in a fixed user template, $\mathrm{doc}(x) =$ `<document>` x `</document>`. The judge decodes a single token at temperature 0. Instead of keeping only the sampled token, we record the ranked top-k window $\mathcal{W}(x) \subset \mathcal{V}$ of that one decoding step together with its log-probabilities.

Answer scheme. An *answer scheme* is a triple $(\mathcal{L}, \nu, v)$. The first part is a finite label set $\mathcal{L}$, given in the prompt as the allowed answers. The second is a fold $\nu : \mathcal{V} \to \mathcal{L} \cup \{\perp\}$ that reads a raw token and returns the label it spells, ignoring case and stray whitespace or punctuation (`` `` Human'' ``, `HUMAN.`, and `human` all fold to `human`), or $\perp$ if the token does not spell any allowed answer. The third is a value function $v : \mathcal{L} \to [0, 1]$ with $v(\ell) = 1$ for a label meaning "certainly human" and $v(\ell) = 0$ for "certainly machine". Given a scheme, define the per-label mass and the total answer mass

$$p_\ell(x) \;=\; \sum_{t \in \mathcal{W}(x):\, \nu(t) = \ell} q_J\big(t \mid c(x)\big), \qquad P(x) \;=\; \sum_{\ell \in \mathcal{L}} p_\ell(x) \;\in\; [0, 1]. \tag{3}$$

Empirically $P(x)$ is close to 1: across our benchmarks the 5th percentile of $P(x)$ is ≥ 0.98. The prompted scheme therefore captures nearly all of the judge's mass, and truncating to the top-k window introduces very little error.

The general family. $\lambda(x) \in [0, 1]$ in Eq. 1 is a free confidence weight: any function of x into $[0, 1]$ defines a valid LogJudge score. Two properties hold for any choice of λ. First, $s(x) \in [0, 1]$, since it is a convex combination of values $v(\ell) \in [0, 1]$ and $\frac{1}{2}$. Second, $s(x) \to \frac{1}{2}$ as $\lambda(x) \to 0$, so the score falls back to complete ignorance when we choose not to trust the judge's answer on x at all.

The simple case: $\lambda(x) = P(x)$. Setting the confidence weight to the answer mass itself gives a canonical, hyperparameter-free instance, the *shrinkage* readout used throughout the paper, which treats residual mass as uninformative (value $\frac{1}{2}$):

$$s_{\mathrm{shr}}(x) \;=\; \sum_{\ell \in \mathcal{L}} p_\ell(x)\, v(\ell) \;+\; \big(1 - P(x)\big) \cdot \tfrac{1}{2}. \tag{4}$$

It is a weighted average of values in $[0, 1]$ and so always lies in $[0, 1]$. It is continuous in $P(x)$ and moves smoothly to the uninformative prior $\frac{1}{2}$ as $P(x) \to 0$. It also needs no floor, branching, or clamp, which is why we use it as the default throughout the paper.

Instances. `ord2` and `ord5` are two instances of Eq. 4 that share everything except the answer scheme, summarized in Table 5: a binary scheme asking for `human`/`machine`, and an ordinal five-letter scale asking for one of five graded ratings, A (certainly human) through E (certainly machine).

Table 5: LogJudge score variants as instances of the general formulation.

Variant	Scheme	$\mathcal{L}$	v	Readout
`ord2`	binary	{`human`, `machine`}	1, 0	s_{shr} (Eq. 4)
`ord5`	ordinal (5-level)	$\{A, B, C, D, E\}$	1, .75, .5, .25, 0	s_{shr} (Eq. 4)

All detection results in this paper use `ord2`. For `ord5` we measured only the stability of its log-probability capture (Appendix V). We leave a detection comparison of the two schemes for future work. Since both share the same confidence weight, such a comparison would isolate the effect of the answer scheme's resolution.

Edge cases and orientation. If the judge's call fails outright, $\mathcal{W}(x) = \emptyset$ and the score is undefined (`Null`), rather than $\frac{1}{2}$, so that a failed measurement is not conflated with an answered-but-uninformative one. Throughout, $s(x)$ is oriented as $\Pr(y = 0 \mid x)$ (P(human)). Under the paper's global convention that larger scores indicate the machine-generated class, the LogJudge detector score is $f_\theta^{\mathrm{LogJudge}}(x) = 1 - s(x)$.

E BENCHMARK DIFFICULTY

Figure 11 shows why the regeneration matters: the original Binoculars backbone separates GB-orig perfectly and loses most of its power on the Manugram splits.

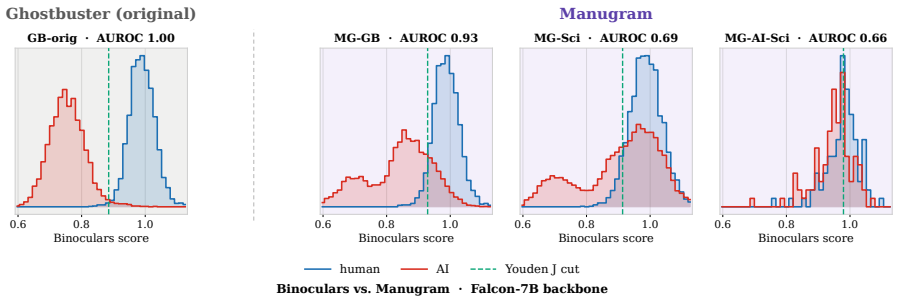


Figure 11: Benchmark difficulty. Human (blue) and AI (red) Binoculars (Hans et al., 2024) score densities with the original Falcon-7B backbone on GB-orig (gray) and the three core Manugram splits (violet). AUROC falls from 1.00 to 0.93, 0.69, and 0.66. The main tables use the Llama-3.1-8B backbone, which is stronger on Manugram (Appendix K).

F MANUGRAM-GHOSTBUSTER PROMPTING

The original Ghostbuster (Verma et al., 2024) benchmark spans three domains: news (`reuter`), creative writing (`wp`), and student essays (`essay`). Each has 1000 human examples and six sets of 1000 machine-generated versions (five `gpt-3.5-turbo` prompts and one contemporary Claude model)[5], and the benchmark is evaluated both in-domain and OOD (training on two domains and holding one out). Recent detectors already reach near-perfect performance on it[6], which motivates Manugram (Section 4). The GB-orig split used in this paper pairs each human text with one machine text ($n = 5{,}988$ in total). The machine text comes from the original `gpt` set: `gpt-3.5-turbo` under the default Ghostbuster writing prompt. The four `gpt-3.5-turbo` prompt-variant sets and the Claude set are not used.

Manugram-Ghostbuster regenerates these three domains with four current generators, each producing a counterpart for every human text, for a benchmark of nearly[7] 15k texts. Both human and machine texts go through a rule-based pass that removes differences in dash and quotation-mark style.

Machine counterparts for Manugram-Ghostbuster are generated with a single user turn (no system prompt), following the domain-specific writing prompt from the original Ghostbuster pipeline (Verma et al., 2024), extended with an explicit anti-artifact instruction. This is the anti-formatting instruction described in Section 4: recent instruction-tuned models default to a markdown-heavy, bulleted style that would otherwise make the machine class trivially separable on formatting alone, so the instruction explicitly forbids it. `{prompt}` / `{headline}` is the item-specific writing prompt or news headline carried over from the paired human document, and `{words}` is that document's own word count, rounded to the nearest 50. The identical template (prompt id `0701A`) is reused verbatim for all four generators (`gpt-5.6-luna`, `gpt-oss-120b`, `gemma-4-31b`, `llama-3.3-70b`), so the four machine classes differ only in the generator and never in the prompt wording.

[5]The original dataset was missing 6 human texts from `essay`, so we dropped these items and their machine counterparts.

[6]Binoculars (Hans et al., 2024) claims 0.949, 0.935, and 0.976 TPR@0.01%FPR on `reuter`, `wp`, and `essay`, respectively.

[7]Some human examples and generations are missing, e.g. where a generator repeatedly refused a topic for safety reasons.

Essay domain, user prompt

```
Write an essay in {words} words to the prompt ``{prompt}.''  Do not
use any markdown formatting (no asterisks, headers, or bullet points)
--- write in plain prose only, as a human would.
```

WP (creative writing) domain, user prompt

```
Write a story in {words} words to the prompt ``{prompt}.''  Do not use
any markdown formatting (no asterisks, headers, or bullet points) ---
write in plain prose only, as a human would.
```

Reuter (news) domain, user prompt

```
Write a news article in {words} words based on the headline
``{headline}.''  Do not use any markdown formatting (no asterisks,
headers, or bullet points) --- write in plain prose only, as a human
would.
```

G MANUGRAM-SCIENCE PROMPTING

Human abstracts are fetched from OpenAlex (Priem et al., 2022) under an automatic filter: publication year ≤ 2020 (to predate widespread LLM use), 80–400 words, a title of ≥ 3 words, English only, boilerplate removed, and de-duplicated by OpenAlex work ID, DOI, and normalized title.

Manugram-Science abstracts are generated in two turns: a fixed system prompt sets the writing register, and a per-item user prompt supplies the paper's title, field, and target length. The released corpus uses the title-only prompt variant (id `0811A`): the generator never sees the human abstract's wording or claims, only its title, field, and word count, so that any human/machine similarity in a pair is due to topic and length matching alone (Section 4), not textual leakage. {`title`} and {`domain`} (one of `computer_science`, `mathematics`, `sociology`) are copied from the paired human abstract's metadata, and {`target_words`} is that abstract's own word count, which keeps the generated abstracts from converging on a single typical length. The same two prompts, with the same substitutions, are used for all four generators.

System prompt

```
You are a researcher writing the abstract of your own paper.
Write in the register and structure of a real published abstract
in the stated field.  Report concrete results the way a published
abstract does:  state the actual figures rather than describing
them qualitatively, and never hedge a quantity as ``significant'',
``substantial'' or ``promising'' in place of giving it.  Where the
prompt supplies numbers, they are your results --- use them as given.
Output the abstract text only:  no title, no ``Abstract:''  heading,
no section labels such as ``Background:''  or ``Methods:'', no
markdown formatting, no bullet points.
```

User prompt template (title-only variant, id 0811A)

```
Write the abstract of a research paper.
Title:  {title}
Field:  {domain}
Write approximately {target_words} words.  Output only the abstract
text, with no title, no headings, and no section labels.
```

H MANUGRAM-AI-SCIENTIST PAIRING

This appendix gives the full pairing cost referenced from Section 4. For an AI abstract a and a candidate human abstract h in the same domain, the pairing cost combines the absolute difference

in abstract length and the topic dissimilarity,

$$C(a,h) \;=\; w_{\text{len}} \cdot \frac{\big|\,|a| - |h|\,\big|}{s} \;+\; w_{\text{top}} \cdot \Big(1 - \cos\big(\mathbf{e}_a, \mathbf{e}_h\big)\Big) \tag{5}$$

where $|a|$ and $|h|$ are abstract word counts, $\mathbf{e}_a, \mathbf{e}_h$ are SPECTER2 embeddings (Singh et al., 2023) of the two abstracts, s is a length-scale constant, and $w_{\text{len}}, w_{\text{top}}$ weight the two terms. We use $w_{\text{len}} = w_{\text{top}} = 1.0$ and $s = 40$. The final assignment $\pi^\star$ minimizes the total pairing cost over all one-to-one matchings Π within a domain,

$$\pi^\star \;=\; \arg\min_{\pi \in \Pi} \sum_i C\big(a_i, h_{\pi(i)}\big), \tag{6}$$

solved with the Hungarian/Jonker–Volgenant algorithm (Kuhn, 1955; Crouse, 2016).

The 100 AI abstracts are drawn from Denario (Villaescusa-Navarro et al., 2025) (80), Sakana AI-Scientist (Lu et al., 2024) (10), Sakana AI-Scientist-v2 (Yamada et al., 2025) (3), CycleResearcher (Weng et al., 2025) (6), and Zochi (Intology, 2025) (1). Candidate human abstracts are fetched from OpenAlex per domain under the same admission rules as Manugram-Science (Appendix G), and every matched human abstract is checked against the Manugram-Science corpus by OpenAlex work ID, DOI, and normalized title to guarantee no overlap.

I Manugram-Rephrase Prompting

Both Manugram-Rephrase corpora are produced with `gpt-5.6-luna` using the same two-part system prompt, which is a fixed rephrasing instruction plus one of six domain-specific hints (Table 6), with the source text as the user turn. Each of Manugram-Rephrase-Ghostbuster and -Science holds two sets of rephrasings. The *human* originals are rephrased under prompt id `0908R`, and `gpt-5.6-luna`'s own MG-GB and MG-Sci generations for the same documents under id `0916R`. Together with the unrephrased texts, this gives a 2×2 design of {human, `gpt-5.6-luna`} $\times$ {original, rephrased} (Section 4). The rephrasing changes the wording and sentence structure substantially but keeps every fact, claim, and structural choice of the source. Each rephrase is checked post hoc against its source: a rephrase whose word-count ratio to the source falls outside $[0.6, 1.5]$ is regenerated, and the surface Jaccard overlap and `difflib` sequence-match ratio to the source are released as quality-control columns but not used for filtering.

System prompt, fixed part (both 0908R and 0916R)

```
You are rephrasing an existing piece of writing.  Rewrite it so that
the wording and sentence structure differ substantially from the
original, while preserving its meaning exactly.
Rules:
     • Preserve every fact, claim, figure, name, date, citation and
       quotation, and keep them in the same logical order.
     • Do not add anything:  no new information, examples, framing, or
       opinions.
     • Do not remove anything:  every point in the original must still
       be present.
     • Do not change the genre, register, or point of view.
     • Rephrase at the sentence and phrase level --- replace
       vocabulary, recast clauses, split or join sentences --- but
       do not reorganise the piece or substitute your own argument.
     • Keep the length within about 15% of the original.
     • Output only the rewritten text.  No preamble, no notes, no
       markdown, no headings, no labels.
```

The domain-specific hint below is appended to the system prompt above, joined by a blank line, chosen by the source document's domain.

J Background: Detection Metrics

A detector is a scoring function $f_\theta : \mathcal{X} \to \mathbb{R}$ over texts, oriented so that larger values indicate the machine-generated class ($y = 1$). Detectors with the opposite native orientation, such as Binoc-

Table 6: Domain-specific hints appended to the rephrasing system prompt (Appendix I), one per source domain.

Domain	Hint
`essay`	The text is a student's argumentative or analytical essay. Keep the thesis, the sequence of arguments, and every supporting example exactly as they are.
`reuter`	The text is a news-wire article. Keep every reported fact, all attributions and quotations, named sources, figures, and the dateline/lede style of a news report.
`wp`	The text is a piece of short creative fiction written to a prompt. Keep the plot, the characters, the sequence of events, the point of view and the tone. Vary the prose, not the story.
`computer_science`	The text is the abstract of a computer-science paper. Keep the problem, the method, the system or algorithm described, every quantitative result and metric, and the stated contributions.
`mathematics`	The text is the abstract of a mathematics paper. Keep every definition, theorem, hypothesis and result, and the precise logical statements. Do not alter notation or the exact form of any claim.
`sociology`	The text is the abstract of a sociology paper. Keep the research question, the data and method, every finding, the theoretical framing and all cited concepts.

ulars (Hans et al., 2024), are negated once, a fixed choice that is never adapted per split. A hard decision thresholds the score at τ. For a threshold τ, the true positive rate and false positive rate are the class-conditional tail probabilities of the score,

$$\mathrm{TPR}(\tau) = \Pr\big(f_\theta(x) \geq \tau \mid y = 1\big), \qquad \mathrm{FPR}(\tau) = \Pr\big(f_\theta(x) \geq \tau \mid y = 0\big).$$

The AUROC is the area under the ROC curve $\{(\mathrm{FPR}(\tau), \mathrm{TPR}(\tau)) : \tau \in \mathbb{R}\}$, or equivalently $\Pr\big(f_\theta(X^+) > f_\theta(X^-)\big)$, the probability that a random machine-generated text is scored above a random human-written one. It summarizes all thresholds in a single number. Because flagging a human author is far more harmful than missing a machine text, we follow Hans et al. (2024) and also report the true positive rate at a fixed false positive rate. The oracle $\mathrm{TPR@1\%FPR}$ fixes the threshold at the $(1-\alpha)$-quantile of the human-class scores of the evaluated split, with $\alpha = 0.01$,

$$\tau_\alpha = \inf\{\tau : \mathrm{FPR}(\tau) \leq \alpha\}, \qquad \mathrm{TPR@}\alpha\mathrm{FPR} = \mathrm{TPR}(\tau_\alpha),$$

so that at most an α-fraction of human texts are flagged. "Oracle" means that τ_α is chosen post hoc on the evaluation set rather than on a training set. AT-HM, the harmonic mean of the normalized AUROC and $\mathrm{TPR@1\%FPR}$, is defined in Section 5.

K Baseline Backbone Selection

Three of the zero-shot baselines in Figure 3 and Table 2 require choosing an observer LLM backbone. Binoculars (Hans et al., 2024) scores a text under one LLM and cross-scores it under a second, closely related one. Fast-DetectGPT (Bao et al., 2024) samples alternative tokens with a sampling LLM and scores them with a scoring LLM, which may be the same model, and as for Binoculars we use a base/instruct pair from one model family. The prompt-based baseline Base-Judge (Section 3) asks an instruction-tuned LLM directly for a verdict stated in text. Rather than keeping each method's original backbone, we swept candidate backbones for all three methods on all four splits (Section 4). We scored oriented AUROC and oracle TPR@1%FPR in the same way as for every other detector in Table 2 and combined both into AT-HM (Section 5). For Binoculars and Fast-DetectGPT this means five open-weight candidates: Falcon-7B (Almazrouei et al., 2023) (the original Binoculars pair), Falcon3-7B, Llama-3.1-8B (Grattafiori et al., 2024), Mistral-v0.3-7B, and Qwen2.5-7B. For the LLM-as-judge baseline it means five instruction-tuned candidates: `gpt-5.6-luna`, `gemma-4-31b`, `gpt-oss-120b`, `llama-3.3-70b`, and `qwen3-14b`. The first four are also the Manugram generators (Section 4), and the first two are the LogJudge backbones (Section 3). Here all five are compared only on the verdict parsed from their written answer, without log-probabilities. Figures 12, 13, and 4 report every (backbone, split) cell plus each backbone's four-split mean, and Table 7 summarizes the four-split means.

Llama-3.1-8B is the strongest backbone for both probability-based methods, by a clear margin on four-split mean AT-HM (Table 7). This is why we use it for Binoculars and Fast-DetectGPT throughout the main paper (Figure 3, Table 2). Falcon-7B is second on AT-HM for both methods (0.557 and 0.572), Mistral-v0.3 trails Llama-3.1-8B by 9–11 points, and Qwen2.5-7B is worst throughout, never clearing AT-HM 0.28.

The only split where Llama-3.1-8B is not the top backbone is GB-orig. There, Falcon-7B, the original Binoculars backbone, is close to ceiling for both methods (Binoculars AUROC 0.998 / TPR@1%FPR 0.986, Fast-DetectGPT 0.996 / 0.973), ahead of Llama-3.1-8B's 0.923 / 0.484 and 0.909 / 0.404 respectively. That ranking inverts on every other split: on MG-Sci, for instance, Falcon-7B falls to AUROC 0.69–0.70 with TPR@1%FPR near 0.31–0.32, while Llama-3.1-8B holds 0.90 AUROC and TPR@1%FPR 0.51–0.59, and the gap is similarly wide on MG-AI-Sci. Figure 11 shows the same pattern qualitatively. Falcon-7B was tuned to an era of machine-generated text that GB-orig's 2023 generations still resemble, so it saturates there but does not transfer to the current generators in the Manugram splits. Llama-3.1-8B gives up a little on GB-orig in exchange for much better and more even results on the harder benchmarks. This is the property we select for, since three of the four splits in the four-split mean are Manugram splits.

For the LLM-as-judge baseline, the gap between backbones is much larger than for either probability-based method (Figure 4). Only `gpt-5.6-luna` (AT-HM 0.653) and `gemma-4-31b` (AT-HM 0.515) clear chance by a useful margin, and these are the two judge backbones used in Figure 3 and Table 2. The other three sit at or barely above chance four-split-mean TPR@1%FPR (at most 0.061): `llama-3.3-70b` (AUROC 0.659, AT-HM 0.089), `gpt-oss-120b` (AUROC 0.643, AT-HM 0.014), and `qwen3-14b` (AUROC 0.527, AT-HM 0.000, i.e. indistinguishable from chance). With Binoculars and Fast-DetectGPT every backbone at least separates the classes, but a weak LLM-as-judge backbone is close to uninformative. This matches Appendix T, where `llama-3.3-70b` and `gpt-oss-120b` also come out as weak judges. It also means that the self-family check in Appendix T has statistical power only for the two strong backbones, because the self-family cells of `llama-3.3-70b` and `gpt-oss-120b` lie within their own near-floor noise. `qwen3-14b` is not a generator and has no self-family cell.

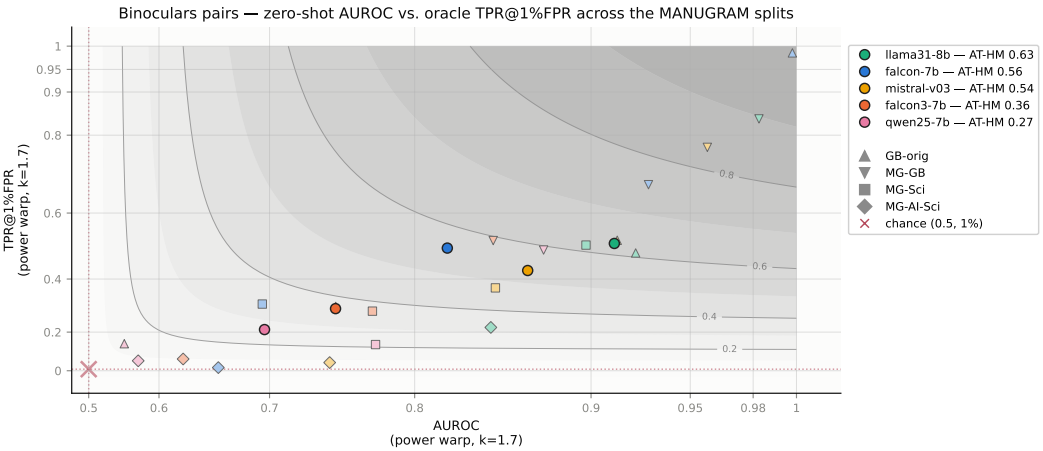


Figure 12: Binoculars (Hans et al., 2024): AUROC vs. oracle TPR@1%FPR for all five candidate backbones across the Manugram splits, same figure design as Figure 3. Llama-3.1-8B (used in the main paper) has the best four-split mean AT-HM, while Falcon-7B is competitive only on the legacy GB-orig split.

L Detailed General Results

Table 8 breaks Table 2 down by domain. We see three patterns. First, a detector can do well within each domain and still lose recall when one threshold must serve the whole split. LogJudge with `gpt-5.6-luna` reaches TPR@1%FPR 0.602–1.000 within the MG-GB domains but 0.547

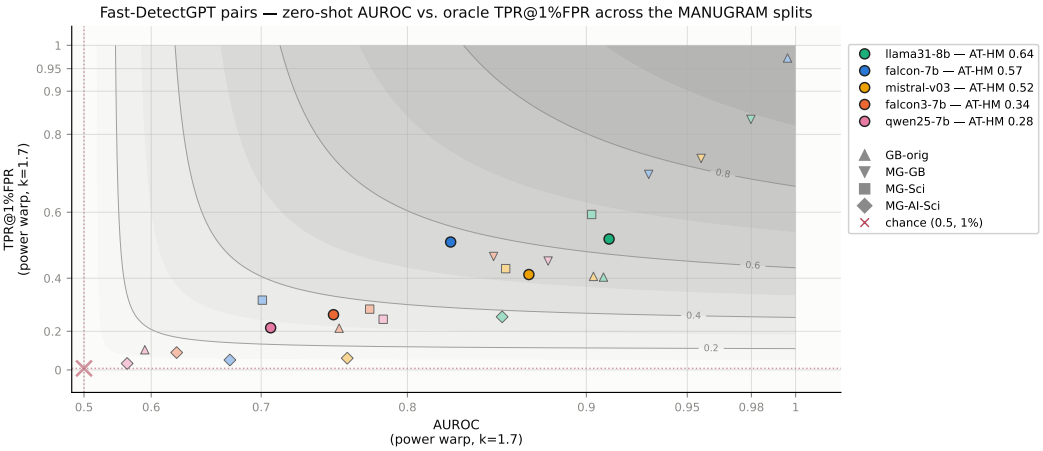


Figure 13: Fast-DetectGPT (Bao et al., 2024): the same backbone sweep and figure design as Figure 12, for the second probability-based zero-shot baseline. Llama-3.1-8B is again the four-split-mean winner.

Table 7: Baseline backbone selection: four-split-mean oriented AUROC, oracle TPR@1%FPR, and AT-HM for every candidate backbone of Binoculars, Fast-DetectGPT, and the parsed-answer LLM-as-judge baseline. **Bold** rows are the backbones used in Figure 3 and Table 2.

Method	Backbone	AUROC ↑	TPR@1%FPR ↑	AT-HM ↑
Binoculars	**Llama-3.1-8B**	**0.912**	**0.512**	**0.628**
	Falcon-7B (orig.)	0.820	0.499	0.557
	Mistral-v0.3	0.866	0.428	0.536
	Falcon3-7B	0.748	0.295	0.364
	Qwen2.5-7B	0.696	0.211	0.268
Fast-DetectGPT	**Llama-3.1-8B**	**0.912**	**0.523**	**0.636**
	Falcon-7B	0.826	0.514	0.572
	Mistral-v0.3	0.870	0.412	0.524
	Falcon3-7B	0.752	0.268	0.344
	Qwen2.5-7B	0.707	0.215	0.276
LLM-as-judge	**gpt-5.6-luna**	**0.974**	**0.503**	**0.653**
	gemma-4-31b	**0.905**	**0.384**	**0.515**
	llama-3.3-70b	0.659	0.061	0.089
	gpt-oss-120b	0.643	0.017	0.014
	qwen3-14b	0.527	0.000	0.000

pooled, and 0.941–1.000 within the GB-orig domains but 0.805 pooled. Fast-DetectGPT loses much less on MG-GB (0.767–0.913 within domains, 0.835 pooled). Second, the domains differ in difficulty. News (Reuter) is close to solved in both Ghostbuster-based splits: every detector reaches AT-HM of at least 0.89 there, except RADAR on MG-GB. On all three MG-Sci domains, FusionJudge is the best detector and BaseJudge the better of the two `gpt-5.6-luna` readouts. Third, FusionJudge inherits BaseJudge's weakness on essays (AT-HM 0.474 on GB-orig and 0.378 on MG-GB, against 0.966 and 0.728 for LogJudge), where averaging with the coarse parsed readout costs recall at 1% FPR. The four-split ranking of Table 2 does not hinge on the small split or on equal weighting: without the 200-text MG-AI-Sci, LogJudge with `gpt-5.6-luna` reaches an AT-HM of 0.792 against 0.712 for Fast-DetectGPT and 0.714 for Binoculars, and with splits weighted by size, 0.767 against 0.749 and 0.736.

Table 8: Per-domain results for GB-orig and MG-GB (Essay, Reuter for news, and WP for WritingPrompts fiction) and MG-Sci (CS for computer science, Math, and Soc for sociology). "All (pooled)" scores the whole split with one threshold. This is the value used in Table 2, and it can differ markedly from the per-domain values. The last two blocks are MG-AI-Sci, which has no per-domain breakdown because its ten fields hold only 8–40 texts each and several have fewer than 10 human examples, and the four-split mean (AT-HM computed from the split-averaged AUROC and TPR, Section 5). Detectors and metrics are the same as in Table 2. Higher is better throughout, and double rules separate splits. **Bold** column headers are the judge family, **bold** values are the best detector per row.

		FusionJudge (`gpt-5.6-luna`)	**LogJudge (`gpt-5.6-luna`)**	**BaseJudge (`gpt-5.6-luna`)**	Fast-DetectGPT (Llama-3.1-8B)	**LogJudge (`gemma-4-31b`)**	Binoculars (Llama-3.1-8B)	**BaseJudge (`gemma-4-31b`)**	RADAR
GB-orig / Essay	AUROC ↑	0.984	0.997	0.973	0.837	0.998	0.864	**0.999**	0.943
	TPR@1%FPR ↑	0.321	0.941	0.195	0.219	**1.000**	0.293	0.998	0.112
	AT-HM ↑	0.474	0.966	0.312	0.322	**0.998**	0.410	**0.998**	0.184
GB-orig / Reuter	AUROC ↑	**1.000**	**1.000**	**1.000**	0.986	0.999	0.989	**1.000**	0.998
	TPR@1%FPR ↑	**1.000**	**1.000**	0.999	0.830	0.998	0.861	0.999	0.998
	AT-HM ↑	**1.000**	**1.000**	0.999	0.894	0.998	0.915	0.999	0.997
GB-orig / WP	AUROC ↑	0.999	**1.000**	0.997	0.878	0.999	0.896	**1.000**	0.872
	TPR@1%FPR ↑	0.997	0.997	0.993	0.246	**0.999**	0.237	**0.999**	0.063
	AT-HM ↑	0.998	0.998	0.994	0.363	**0.999**	0.356	**0.999**	0.100
GB-orig / All (pooled)	AUROC ↑	0.994	0.990	0.991	0.909	**0.998**	0.923	**0.998**	0.926
	TPR@1%FPR ↑	0.918	0.805	0.539	0.404	**0.981**	0.484	0.903	0.126
	AT-HM ↑	0.951	0.883	0.692	0.535	**0.989**	0.611	0.946	0.206
MG-GB / Essay	AUROC ↑	0.957	0.965	0.931	0.961	**0.996**	0.966	0.994	0.443
	TPR@1%FPR ↑	0.246	0.602	0.168	0.767	**0.986**	0.750	0.861	0.136
	AT-HM ↑	0.378	0.728	0.269	0.836	**0.989**	0.830	0.919	0.000
MG-GB / Reuter	AUROC ↑	**1.000**	**1.000**	**1.000**	0.990	0.997	0.991	**1.000**	0.861
	TPR@1%FPR ↑	**1.000**	**1.000**	**1.000**	0.892	0.995	0.878	0.996	0.543
	AT-HM ↑	**1.000**	**1.000**	0.999	0.933	0.994	0.927	0.998	0.617
MG-GB / WP	AUROC ↑	**0.994**	0.985	0.989	0.989	0.987	0.991	0.987	0.485
	TPR@1%FPR ↑	0.918	0.894	0.744	0.913	0.893	0.902	**0.923**	0.051
	AT-HM ↑	**0.952**	0.930	0.844	0.944	0.931	0.940	0.947	0.000
MG-GB / All (pooled)	AUROC ↑	0.979	0.968	0.975	0.980	**0.989**	0.983	0.982	0.609
	TPR@1%FPR ↑	0.655	0.547	0.377	0.835	0.836	**0.838**	0.582	0.123
	AT-HM ↑	0.776	0.687	0.534	0.892	**0.900**	0.897	0.722	0.150
MG-Sci / CS	AUROC ↑	**0.990**	0.975	0.987	0.918	0.761	0.909	0.728	0.317
	TPR@1%FPR ↑	**0.873**	0.570	0.724	0.640	0.122	0.547	0.037	0.001
	AT-HM ↑	**0.923**	0.709	0.829	0.723	0.186	0.652	0.051	0.000
MG-Sci / Math	AUROC ↑	**0.996**	0.989	0.994	0.831	0.878	0.828	0.864	0.442
	TPR@1%FPR ↑	**0.945**	0.772	0.845	0.510	0.469	0.474	0.364	0.011
	AT-HM ↑	**0.968**	0.861	0.910	0.574	0.575	0.547	0.480	0.000
MG-Sci / Soc	AUROC ↑	**0.996**	0.988	**0.996**	0.959	0.867	0.956	0.817	0.355
	TPR@1%FPR ↑	**0.936**	0.789	0.919	0.632	0.275	0.527	0.037	0.002
	AT-HM ↑	**0.963**	0.872	0.953	0.746	0.393	0.665	0.053	0.000
MG-Sci / All (pooled)	AUROC ↑	**0.994**	0.982	0.992	0.903	0.832	0.897	0.799	0.370
	TPR@1%FPR ↑	**0.912**	0.679	0.776	0.594	0.189	0.508	0.051	0.004

continued on next page

Table 8 continued

		FusionJudge (gpt-5.6-luna)	LogJudge (gpt-5.6-luna)	BaseJudge (gpt-5.6-luna)	Fast-DetectGPT (Llama-3.1-8B)	LogJudge (gemma-4-31b)	Binoculars (Llama-3.1-8B)	BaseJudge (gemma-4-31b)	RADAR
	AT-HM ↑	**0.948**	0.794	0.866	0.681	0.284	0.616	0.078	0.000
MG-AI-Sci	AUROC ↑	0.956	**0.958**	0.940	0.855	0.871	0.845	0.843	0.529
	TPR@1%FPR ↑	**0.560**	**0.560**	0.320	0.260	0.040	0.220	0.000	0.020
	AT-HM ↑	0.691	**0.692**	0.462	0.373	0.058	0.325	0.000	0.017
Four-split mean	AUROC ↑	**0.981**	0.975	0.974	0.912	0.922	0.912	0.905	0.608
	TPR@1%FPR ↑	**0.761**	0.648	0.503	0.523	0.512	0.512	0.384	0.068
	AT-HM ↑	**0.848**	0.767	0.653	0.636	0.633	0.628	0.515	0.093

M SCORE DISTRIBUTIONS

For every candidate backbone in Appendix K, Figures 14–17 show the underlying human (blue) vs. AI (red) score distribution on each split and on all four splits pooled (All Benchmarks, $n = 36{,}148$), with two operating points marked: the 1%-FPR threshold (violet) and Youden's J balance point (teal). There is one row per backbone and one panel per split. AUROC, oracle TPR@1%FPR, and Youden's J for that (backbone, split) cell are printed above each panel, together with the score value of the 1%-FPR threshold. Figure 5 in the main text zooms into the decision region of the `gpt-5.6-luna` All Benchmarks panels of Figures 16 and 17.

M.1 BINOCULARS

Figure 14 makes the ranking in Table 7 visible: Falcon-7B (top row) separates the two classes almost perfectly on GB-orig but the distributions collapse into heavy overlap on MG-Sci and MG-AI-Sci, while Llama-3.1-8B keeps a visible gap on every split. Qwen2.5-7B (bottom row) stays the most overlapped throughout.

M.2 FAST-DETECTGPT

Figure 15 uses the same five observer/performer pairs as Figure 14 (Appendix K), so the rows line up one for one. The criterion itself is unbounded rather than centered near 1, but the same backbones separate or overlap on the same splits.

M.3 LLM-AS-JUDGE

Figure 16 shows the parsed-answer judge score for all five instruction-tuned backbones from Appendix K. `gpt-5.6-luna` and `gemma-4-31b` are sharply bimodal, with human and AI mass pulled to opposite ends of the $[0, 1]$ axis on every split. The other three backbones spread their score far more continuously and their human/AI distributions overlap almost entirely, matching their near-chance AT-HM in Table 7.

M.4 LOGJUDGE

Figure 17 gives the LogJudge score (Section 3) for the two judge backbones strong enough to use it. Both collapse to a near-bimodal score with almost all AI-class mass at 0, so the fixed 1%-FPR and Youden thresholds sit right at that edge rather than in open space between the classes. This tighter separation fits LogJudge's higher pooled TPR@1%FPR over its parsed-answer baseline for both backbones in Table 2.

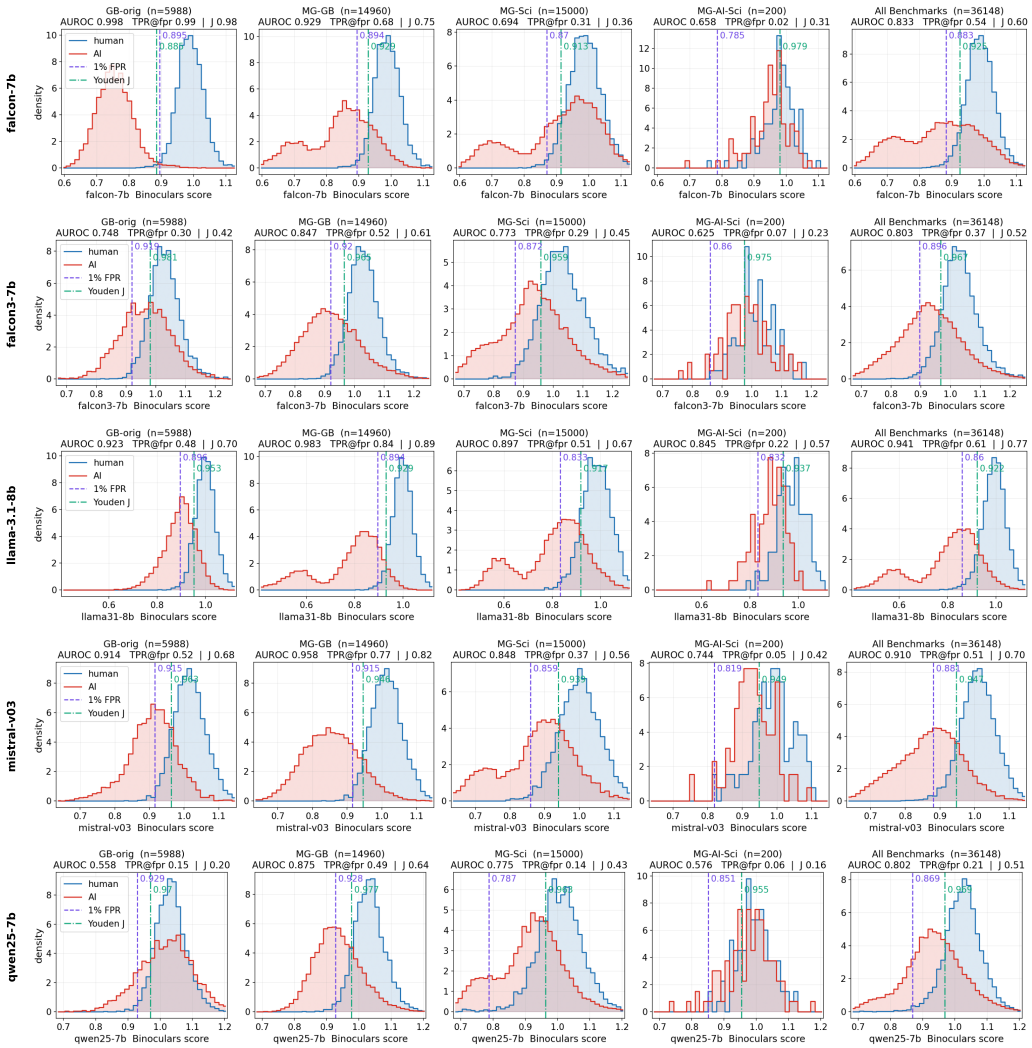


Figure 14: Binoculars score distributions: human vs. AI score distribution for all five candidate backbones (rows) across the four splits and pooled All Benchmarks (columns), with the fixed 1%-FPR (violet) and Youden's J (teal) thresholds marked.

N Four-Split Ranking as a Bar Chart

Figure 18 shows the numbers of Table 2 as a ranking. Each detector has three bars: its four-split mean AT-HM, AUROC, and TPR@1%FPR. Markers show its AT-HM on each split, so the chart shows both the ranking and how much each detector varies across splits. Mean AUROC is close across the judge-based and perplexity-based detectors (0.905–0.981), so the ranking is driven by TPR@1%FPR. The per-split markers show the pattern discussed in Section 5. FusionJudge and LogJudge with `gpt-5.6-luna` are the most even across splits. LogJudge with `gemma-4-31b` ranges from 0.989 on GB-orig to 0.058 on MG-AI-Sci. Fast-DetectGPT and Binoculars peak on MG-GB.

O Same Judge, Two Readouts: The Decision Region

Figure 5 shows each readout only around its own 1%-FPR threshold. Figure 19 shows the whole decision region, with FusionJudge and the achievable operating points, for the same `gpt-5.6-luna` judge on all four splits pooled ($n = 36{,}148$, of which 9,086 are human and 27,062 machine texts).

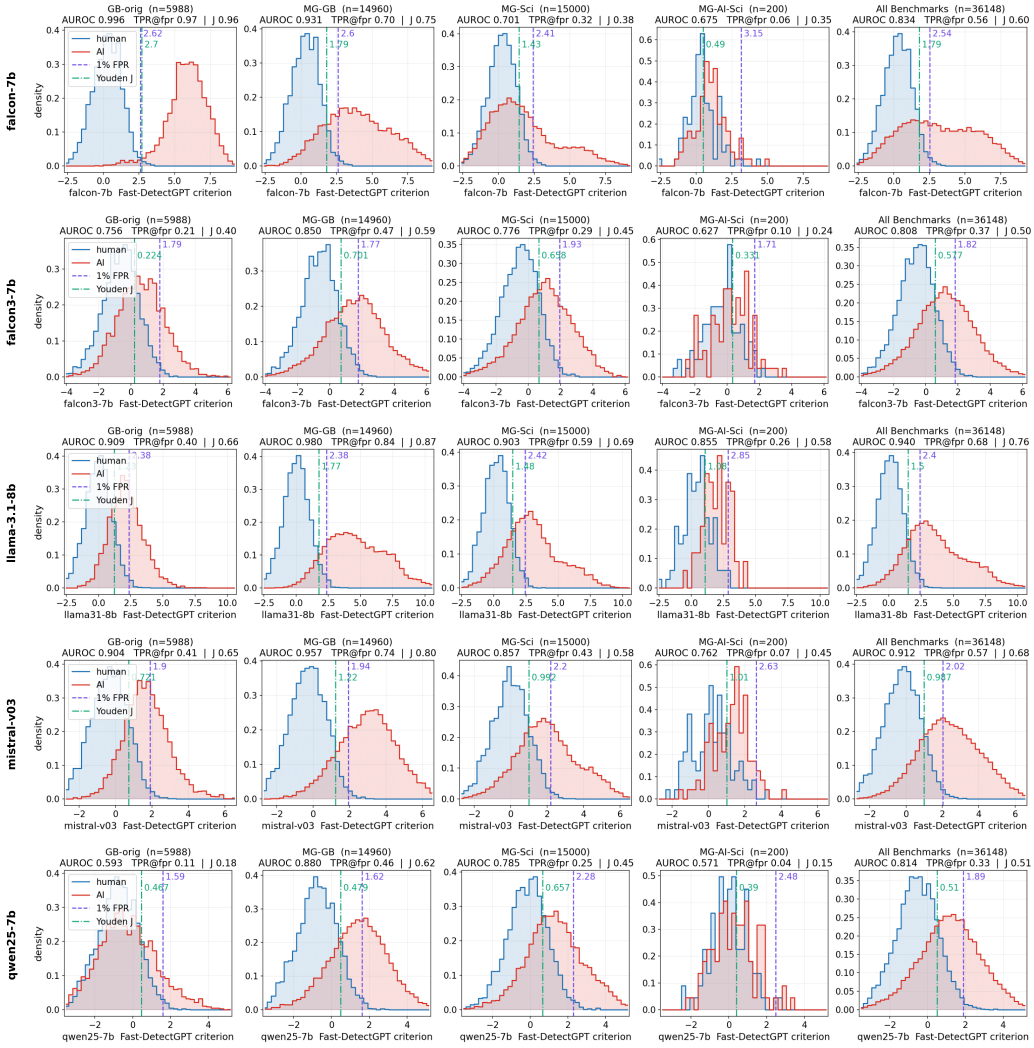


Figure 15: Fast-DetectGPT score distributions: same backbones, splits, and threshold convention as Figure 14.

The two readouts rank texts about equally well, but their resolution is very different. Their AUROC is nearly equal (0.981 parsed, 0.975 log-probability), but the parsed readout takes 91 distinct values over all 36,148 texts, and only 18 of them below the 5th percentile of human scores, while the LogJudge score takes 6,529 and 478. The parsed readout's 1%-FPR threshold falls at $P(\text{human}) = 0.07$ and admits 36% of machine texts. The next attainable value, 0.08, is shared by 66 human and 8,314 machine texts (31% of all machine text), and admitting it moves the operating point to 1.30% FPR and 66% TPR. No threshold can take only part of a tied value, so at 1% FPR the parsed readout must stop before it. The LogJudge threshold falls at 7.45×10^{-5} and admits 62%. Its next value holds 4 human and 98 machine texts, so its operating points are dense through 1% FPR (Figure 19d). The parsed readout's gap is a single straight segment of its ROC curve across the 1% line.

FusionJudge shows why averaging the two readouts helps. Its score, the mean of the two, keeps the clusters of the stated probability (panel c resembles panel a at half the scale), but inside each cluster the LogJudge term breaks the ties: it takes 12,614 distinct values, 5,768 of them below the 5th percentile of human scores. Its 1%-FPR threshold falls at 0.052 with no tie at all (one machine text at the threshold, one human text at the next value), and it admits 66% of machine texts, more than either readout alone. The stated probability supplies the coarse ordering between clusters, on which it is the stronger readout (AUROC 0.981 versus 0.975, and 0.985 for FusionJudge), and

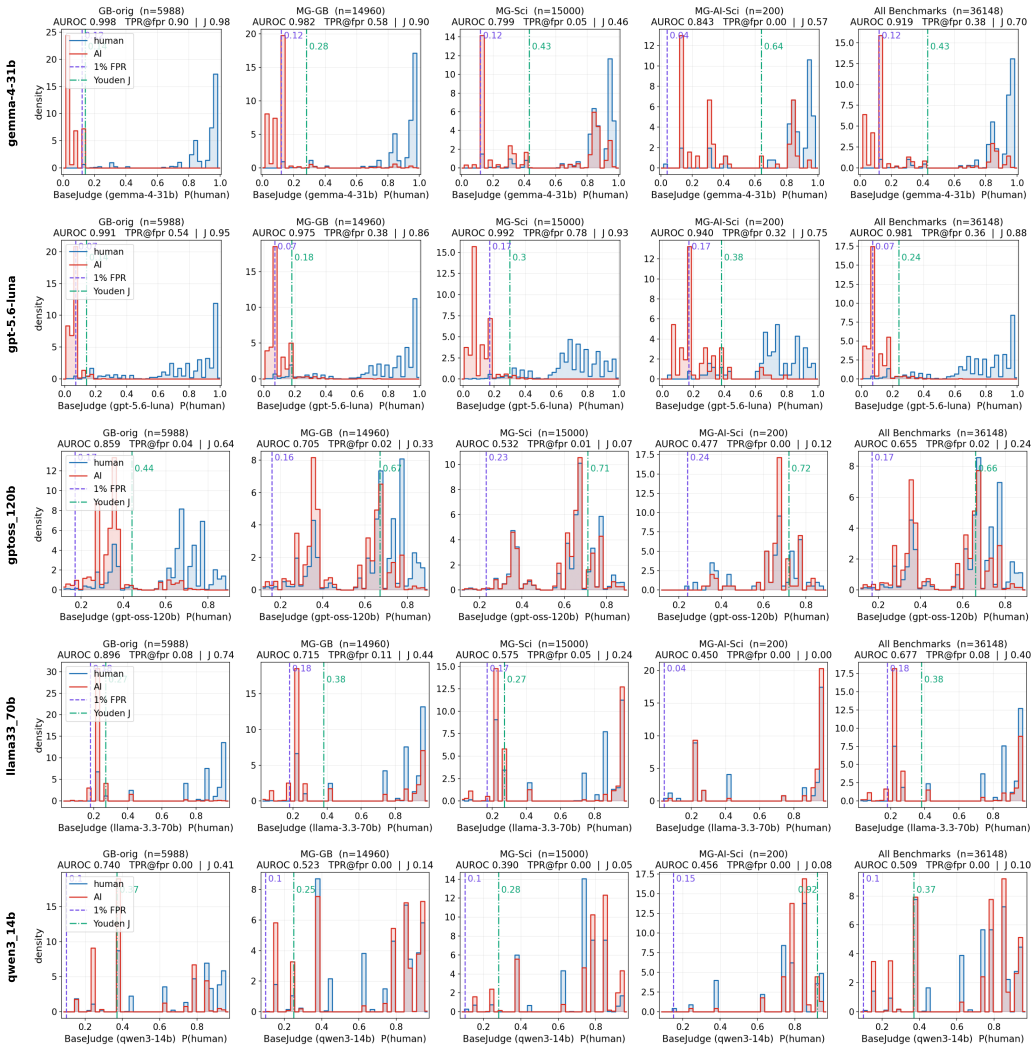


Figure 16: LLM-as-judge (parsed-answer readout) score distributions: same backbones, splits, and threshold convention as Figure 14.

the log-probability supplies the fine ordering within them. On this pooled set its ROC curve lies above both readouts from about 0.4% FPR up to 5% (panel d). Below 0.3% FPR LogJudge alone is stronger (29% versus 11% TPR at 0.1% FPR), since there the fusion's ordering still follows the coarse clusters of the stated probability. Most of the gain is tie-breaking: an explicit rule that sorts by the stated probability and orders each of its values by the LogJudge score agrees with FusionJudge on 99% of text pairs and reaches a four-split mean AT-HM of 0.829, against 0.653 for BaseJudge and 0.848 for FusionJudge. The rest comes from crossing values where LogJudge is confident: 34 of the 118 human texts stated at 0.08 or below end above FusionJudge's threshold, and at the same 90 false alarms FusionJudge flags 388 machine texts that the rule leaves unflagged, against 57 the other way. Pooled recall differs from the four-split means of Table 2, as in Figure 5.

Two features of panel (b) do not affect the threshold: 1,944 machine texts and one human text score at most 10^{-8}, since `human` is essentially absent from the top-k window, and the gaps below 3×10^{-6} and near 10^{-2} reflect the precision of the returned log-probabilities. All of these lie far from the threshold.

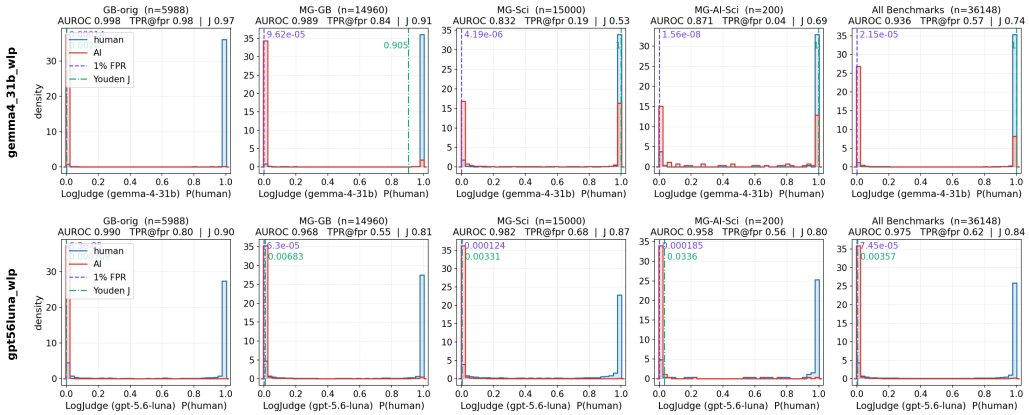


Figure 17: LogJudge score distributions: the two strong judge backbones, same splits and threshold convention as Figure 14.

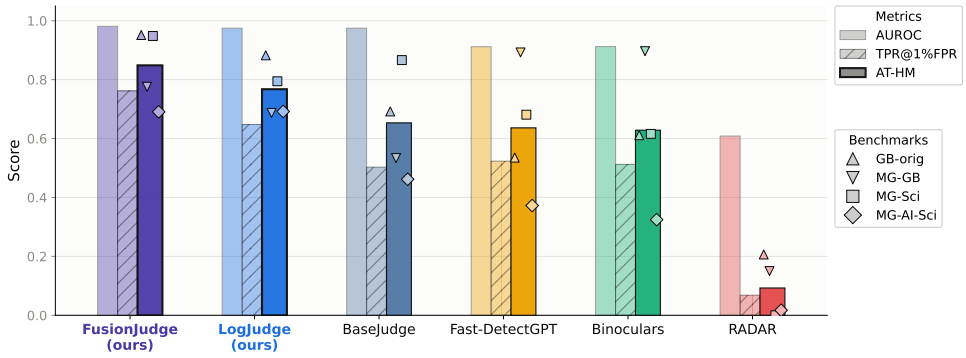


Figure 18: The detectors of Table 2 ranked by four-split mean AT-HM (solid bars), with mean AUROC and TPR@1%FPR alongside. Markers give each detector's AT-HM on the individual splits.

P SAME JUDGE, TWO READOUTS: PER-SPLIT VALUES

Table 9 gives the per-split AUROC and TPR@1%FPR behind Section 5. The two readouts have similar AUROC in every cell. The difference is at 1% FPR. The log-probability readout raises TPR@1%FPR by up to 0.266 (`gpt-5.6-luna` on GB-orig, 0.539 to 0.805) and 0.254 (`gemma-4-31b` on MG-GB, 0.582 to 0.836). The one cell where it loses is `gpt-5.6-luna` on MG-Sci (0.776 to 0.679).

Table 9: Same judge, two readouts. AUROC / TPR@1%FPR of the parsed readout (BaseJudge) and the log-probability readout (LogJudge) for the two judges on each split and averaged over splits. Bold marks the better readout within a judge and cell.

Judge	Readout	GB-orig	MG-GB	MG-Sci	MG-AI-Sci	Mean
`gpt-5.6-luna`	parsed (BaseJudge)	**0.991** / 0.539	**0.975** / 0.377	**0.992** / **0.776**	0.940 / 0.320	0.974 / 0.503
	log-probability (LogJudge)	0.990 / **0.805**	0.968 / **0.547**	0.982 / 0.679	**0.958** / **0.560**	**0.975** / **0.648**
`gemma-4-31b`	parsed (BaseJudge)	0.998 / 0.903	0.982 / 0.582	0.799 / 0.051	0.843 / 0.000	0.905 / 0.384
	log-probability (LogJudge)	0.998 / **0.981**	**0.989** / **0.836**	**0.832** / **0.189**	**0.871** / **0.040**	**0.922** / **0.512**

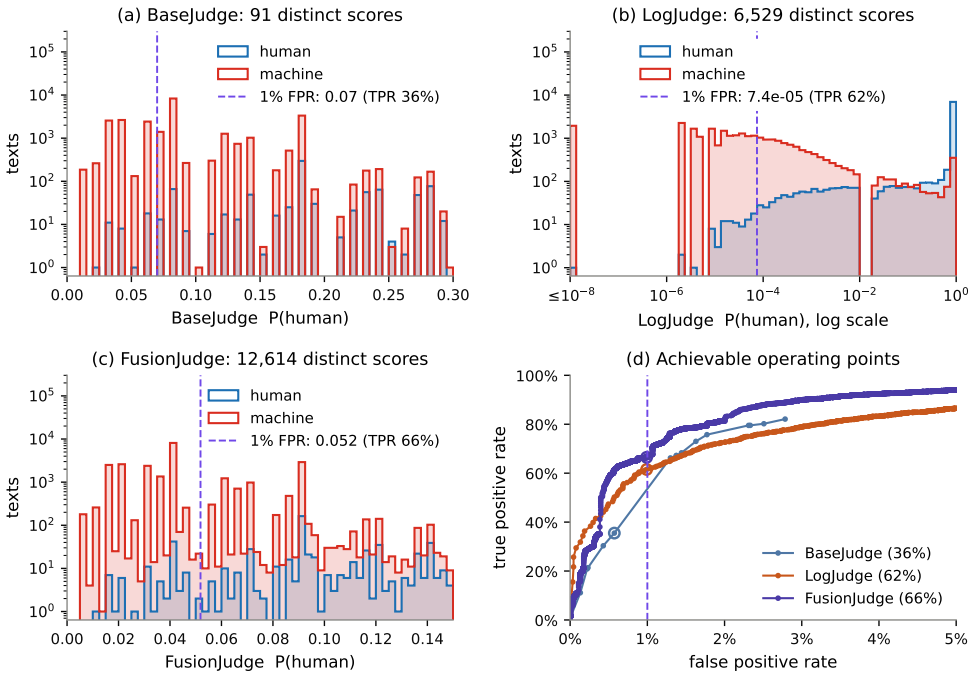


Figure 19: **The decision region, zoomed**, for the same `gpt-5.6-luna` judge on all four splits pooled. (a) The stated probability (BaseJudge), lower tail, fine bins, log count axis. It takes 91 distinct values, and the 1%-FPR threshold (dashed) falls at 0.07, next to a value shared by 66 human and 8,314 machine texts. (b) The LogJudge score on a log scale (values below 10^{-8} clipped onto the left edge). It takes 6,529 distinct values and is continuous through the threshold at 7.45×10^{-5}. (c) FusionJudge, the mean of the two. It keeps the clusters of (a) at half the scale, with ties broken by the LogJudge term, and takes 12,614 distinct values, with the threshold at 0.052. (d) ROC at FPR $\leq 5\%$, one marker per achievable operating point. Circles mark the 1%-FPR points (TPR 36%, 62%, and 66%).

Q Embedding Analysis

This appendix justifies the choice of RoBERTa as the encoder of the trained detector in Section 5. The 12 fixed-length text representations shipped with Manugram span four families: general-purpose sentence encoders (`e5`, `bge`, `mpnet`, `minilm`), a generic semantic encoder (`roberta`), scientific-domain encoders (`specter2`, `scincl`, `scibert`), and authorship or stylistic encoders (`luar`, `luar_crud`, `styledistance`, `style_wegmann`). The classifier is fixed across embeddings so that any gap is attributable to the representation: L2-regularized logistic regression on standardized features (C=1.0), fit on a grouped 70/30 split within each of the four splits (a human text and every machine counterpart of it share a group), repeated over 5 seeds, scored by oriented AUROC and oracle TPR@1%FPR (Section 5). Table 10 reports the four-split mean. Its SD is the spread across splits, so it measures consistency rather than seed noise.

The generic semantic encoder `roberta` is the best embedding on both metrics and the most consistent (cross-split SD $\pm$0.001 AUROC), including on the two scientific splits where a domain-pretrained encoder such as SPECTER2 or SciBERT might be expected to lead. The authorship embedding `luar` is the closest competitor and almost as stable ($\pm$0.002 AUROC), consistent with a writing-style signal that survives the shift from news, essays, and fiction to scientific abstracts. The style-only encoders lose most at the 1%-FPR operating point on MG-AI-Sci, whose human/machine pairs are matched on topic and length (Eq. 5), so these encoders can no longer lean on topic or length cues that correlate with the class. With only 100 human texts in that split, TPR@1%FPR can also take only a few distinct values there.

Table 10: Embedding ablation: four-split-mean AUROC and oracle TPR@1%FPR of an L2 logistic-regression head on each of the 12 embeddings, with the SD across the four splits, ranked by mean AUROC.

Embedding	AUROC ↑	TPR@1%FPR ↑
roberta	**0.999 ± 0.001**	**0.988 ± 0.016**
luar	0.997 ± 0.002	0.958 ± 0.006
luar_crud	0.995 ± 0.004	0.893 ± 0.085
scibert	0.994 ± 0.006	0.929 ± 0.057
specter2	0.992 ± 0.005	0.892 ± 0.044
e5	0.992 ± 0.004	0.882 ± 0.036
styledistance	0.992 ± 0.012	0.904 ± 0.150
scincl	0.987 ± 0.004	0.796 ± 0.080
style_wegmann	0.983 ± 0.026	0.835 ± 0.232
bge	0.979 ± 0.004	0.741 ± 0.084
mpnet	0.965 ± 0.006	0.618 ± 0.118
minilm	0.937 ± 0.010	0.452 ± 0.088

R Domain Shift: Exact Values and Per-Domain Views

Table 11 gives the AT-HM behind each point of Figure 6. Figure 20 shows the computer-science and mathematics domains individually in the AUROC-against-TPR@1%FPR design of Figure 3. These views also include a logistic-regression head trained on the same frozen RoBERTa features, which collapses under domain shift in the same way as the MLP head.

Table 11: AT-HM in the four domain-shift scenarios, averaged over the three MG-Sci domains. 🔥 marks the trained detectors (RoBERTa-Sentinel's MLP head and a logistic-regression head on the same frozen RoBERTa features), ❄ the zero-shot detectors, whose values are identical across columns by construction. Bold marks the best value per column.

Detector	In-Domain	Near-OOD	Mid-OOD	Far-OOD
🔥 RoBERTa-Sentinel (Chen et al., 2023)	0.984	0.865	0.314	0.140
🔥 RoBERTa + logistic head	**0.996**	**0.975**	0.391	0.213
❄ FusionJudge (`gpt-5.6-luna`)	0.950	0.950	**0.950**	**0.950**
❄ LogJudge (`gpt-5.6-luna`)	0.841	0.841	0.841	0.841
❄ Fast-DetectGPT (Llama-3.1-8B)	0.689	0.689	0.689	0.689
❄ Binoculars (Llama-3.1-8B)	0.627	0.627	0.627	0.627

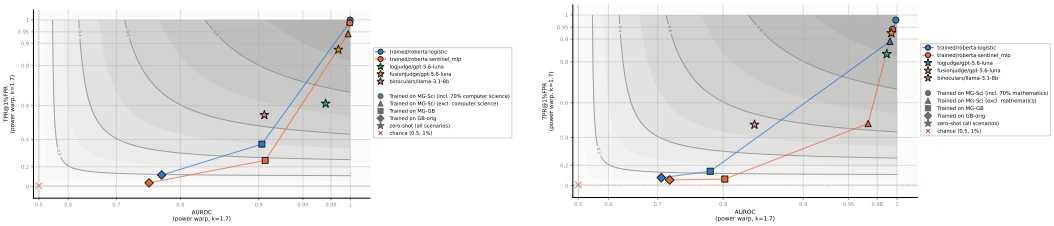


Figure 20: Domain shift on the computer-science (left) and mathematics (right) domains of MG-Sci. Circles, triangles, squares, and diamonds are the trained detectors in the In-Domain, Near-OOD, Mid-OOD, and Far-OOD scenarios, and stars are the zero-shot detectors, whose scores do not depend on the scenario. Both a logistic-regression head and the MLP head on frozen RoBERTa features collapse as the training data move away from the evaluation domain.

S REPHRASING: SCORE DISTRIBUTIONS

Figure 21 shows the Fast-DetectGPT criterion for the four classes of MG-Reph. Rephrasing moves `gpt-5.6-luna`'s machine text toward the human side (human-versus-machine AUROC 0.958 to 0.860 on MG-GB, 0.718 to 0.540 on MG-Sci), while rephrased human text moves, if anything, further toward the human side (human-versus-rephrased-human AUROC 0.405 and 0.390).

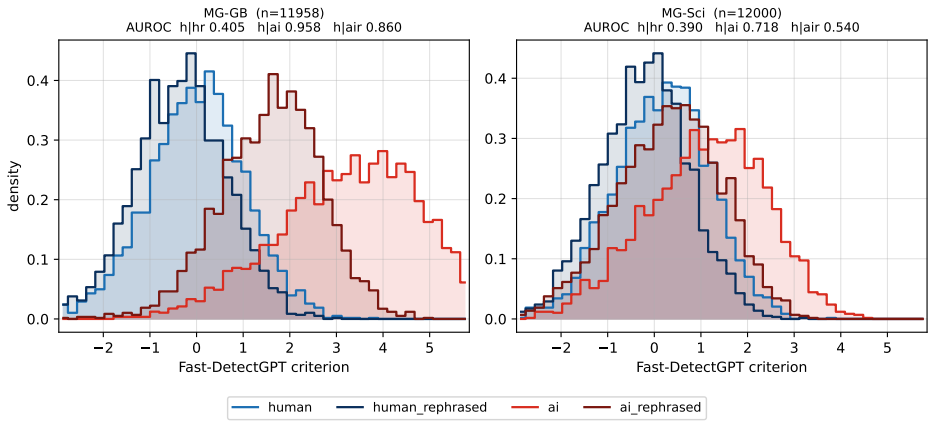


Figure 21: Fast-DetectGPT (Llama-3.1-8B) score densities for human, rephrased human, machine (`gpt-5.6-luna` generation), and rephrased machine text on MG-Reph-GB (left) and MG-Reph-Sci (right). Higher means machine. Panel headers give the AUROC of human against each other class.

T JUDGE–GENERATOR PAIRING

This appendix gives the full diagnostic behind Section 5. The question is whether a judge is soft on text produced by its own underlying model. `gpt-5.6-luna` is both the main judge and one of the four Manugram generators (Section 4), so a same-model judge/generator pairing exists for it, and likewise for `gemma-4-31b`, `llama-3.3-70b`, and `gpt-oss-120b`, which double as generator and judge backbone even though the latter two are not competitive detectors on their own. GB-orig predates this generator roster and contributes no same-model cell.

For each (judge, generator) pair on MG-GB and MG-Sci we take the human rows plus that one generator's rows, compute oriented AUROC and oracle TPR@1%FPR on the restricted set, and fold both into AT-HM (Section 5). The diagnostic uses the parsed readout (BaseJudge), so the single-generator GB-orig cells coincide with BaseJudge's GB-orig values in Table 2. This differs from Appendix L, which pools all generators against human text within a domain. Here each cell isolates one generator, so a judge's self cell can be compared with its cells for every other generator in the same split. Figure 9 reports every cell, with the judge's own model outlined.

A low cell means that the judge separates that generator's text from human text poorly. Two observations bound how much of `gpt-5.6-luna`'s low self cell (0.18 on MG-GB, 0.66 on MG-Sci) is self-leniency. First, `gpt-5.6-luna` text is also the hardest generator for the `gemma-4-31b` judge on MG-GB (0.51 against 0.66–0.94), so part of the effect is that the strongest generator is the hardest case for every detector. Second, `gemma-4-31b`'s own self cell (0.72 on MG-GB) is not the lowest in its row, and on MG-Sci that judge is near floor for every generator except `llama-3.3-70b` (0.28), so there is no consistent self-leniency signature across judges. `llama-3.3-70b` and `gpt-oss-120b` sit at floor in every cell (mean AT-HM 0.12 and 0.02) and carry no information. The check was run on the parsed readout only.

U APPLICATION: ICLR ABSTRACTS, 2020–2025

This appendix gives the details behind Section 5. We apply the `gpt-5.6-luna` judge unchanged (prompt, readouts, and standardization as in the main experiments) to ICLR abstracts indexed by OpenAlex (Priem et al., 2022) for the conference years 2020–2025. These abstracts have no authorship labels, so we report how many a detector flags, not its accuracy. Instead of an oracle threshold, each detector keeps one fixed threshold: its 1%-FPR threshold on the MG-Sci human abstracts, which were all published by 2020. On MG-Sci these thresholds flag 0.7%, 1.0%, and 1.0% of the human abstracts and 77.6%, 67.9%, and 91.2% of the machine abstracts for BaseJudge, LogJudge, and FusionJudge, respectively (Table 12, bottom rows).

Filtering. OpenAlex lists placeholders under ICLR ("International audience", "status: Published online", and records of datasets and slides). We keep an abstract only if it passes the text rules of the MG-Sci filter (Appendix G: 80–400 words, a title of ≥ 3 words, not truncated, no boilerplate, English) and drop repeated abstracts, keeping the earliest year. Of 10,914 records, 9,754 remain. The largest losses are 567 records without an abstract, 280 that are too short, and 237 duplicates.

Results. On the 2020 abstracts, every detector flags 0.5–0.6%, within the nominal 1% FPR, consistent with the MG-Sci thresholds transferring to ICLR's pre-LLM abstracts. From there the flag rate rises (Figure 10a, Table 12). FusionJudge flags 2.1% of the 2022 abstracts, 5.6% of the 2023 abstracts, and 11.1% of the 2025 abstracts (95% Wilson interval 10.0–12.3%), eighteen times its 2020 rate. LogJudge rises almost identically, to 10.2%. BaseJudge rises least, to 3.8%: its threshold (0.17) sits on one of its rounded values, the tie structure of Section 5. The shift is not confined to the tail. FusionJudge's mean $P(\text{human})$ falls from 0.643 to 0.331, and the bulk of the 2025 distribution sits at 0.1–0.2, just above FusionJudge's threshold of 0.092, where the 2020 abstracts sit mostly at 0.8–0.9 (Figure 10b). Papers about language models do not drive the trend: the share of abstracts that mention language models, GPT, transformers, or prompting grows from 8% (2020) to 52% (2025), yet on the remaining abstracts FusionJudge's flag rate follows the same course (0.7%, 2.2%, 5.3%, and 10.2% in 2020, 2022, 2023, and 2025).

Table 12: ICLR abstracts flagged as machine per conference year, at each detector's 1%-FPR threshold set on the MG-Sci human abstracts (all published by 2020). All detectors use the `gpt-5.6-luna` judge. The last column is FusionJudge's mean $P(\text{human})$. The bottom rows apply the same thresholds to MG-Sci itself: the share of human abstracts flagged (FPR) and of machine abstracts flagged (TPR). †BaseJudge's scores take only 88 values on MG-Sci, so its FPR cannot reach exactly 1%: its threshold (0.17) flags 0.7% of the human abstracts, and the next value (0.18) is shared by 38 human abstracts and would flag 2.0%, over budget (Section 5, Appendix O).

		Flagged as machine (%)			
Year	n	BaseJudge	LogJudge	FusionJudge	Mean $P(\text{human})$
2020	799	0.5	0.6	0.6	0.643
2021	996	0.5	1.8	0.9	0.609
2022	1,156	1.0	2.0	2.1	0.557
2023	1,927	2.6	4.7	5.6	0.482
2024	1,898	2.5	5.7	6.1	0.430
2025	2,978	3.8	10.2	11.1	0.331
MG-Sci human (FPR)	3,000	0.7†	1.0	1.0	–
MG-Sci machine (TPR)	12,000	77.6	67.9	91.2	–

V LOGPROB CAPTURE REPRODUCIBILITY

Neither judge call is given an explicit `seed`, and both go through shared, batched production endpoints (the OpenAI Responses API for `gpt-5.6-luna`, an institutional inference endpoint for `gemma-4-31b`) rather than an isolated, single-request process. Floating-point reduction order under batched GPU inference is not guaranteed to be identical across calls even at temperature 0, so we do not assume the captured logprob windows are exactly reproducible. Instead, we measure how much they vary.

We re-captured the answer-token window for the same 150 rows of `ghostbuster_orig`, for both judges and both answer schemes, in a second, independent call issued shortly after the first, and compared the two captures (Table 13). We record whether the top-k window was byte-identical, whether the sampled (arg max) token agreed, whether the resulting LogJudge score (Eq. 4) fell on the same side of $\frac{1}{2}$ (that is, the same human/machine call, regardless of margin), and the distribution of $|s_a(x) - s_b(x)|$.

Table 13: Two independent temperature-0 captures of the same 150 rows, compared directly. "Verdict" is agreement on which side of $\frac{1}{2}$ the LogJudge score falls, that is, the same detection call.

Judge	Scheme	Window	Token	Verdict	Mean $\lvert\Delta s\rvert$	Max $\lvert\Delta s\rvert$	$\lvert\Delta s\rvert>0.05$
`gpt-5.6-luna`	`ord2`	26%	96%	96%	0.039	0.843	15 (10%)
`gpt-5.6-luna`	`ord5`	10%	95%	97%	0.025	0.660	16 (11%)
`gemma-4-31b`	`ord2`	13%	100%	100%	0.001	0.161	1 (1%)
`gemma-4-31b`	`ord5`	8%	99%	99%	0.003	0.222	2 (1%)

The exact top-k window is rarely byte-identical for either judge (13–26% under `ord2`, 8–10% under `ord5`), confirming that the backend is not bit-reproducible even at temperature 0, but the verdict is stable: under `ord2`, the scheme used for all results, the human/machine call changes for none of the `gemma-4-31b` documents and for 4% of the `gpt-5.6-luna` documents, and 10% of the latter (including those that flip) move by more than 0.05. The rare large swings (up to 0.84) are not a consequence of missing answer mass, which is at most 0.02 for 95% of documents (Appendix D). They occur on borderline documents where the judge's top answer itself differs between the two calls (token agreement 95–96% for `gpt-5.6-luna`), so that nearly all of the mass moves from one label to the other. `gemma-4-31b`, whose top answer differed on at most 1% of documents, shows correspondingly small swings. In practice, differences of a few AT-HM points between judge-based detectors are within this jitter. Both captures use the same window as the main results, $k = 20$ for both judges (Appendix C).