WEBMIX-COLLAPSE: A REWRITE BENCHMARK FOR SIZING THE LENGTH CONFOUND IN SHORT-FORM SYNTHETIC-TEXT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

On rewrite-based benchmarks for synthetic-text detection, a detector can score well by counting words. We build such a benchmark, **WebMix-Collapse**, with 10,668 English web documents and model rewrites spanning six contamination rates, ten generator families (1B to 671B parameters) and five rewrite depths, and we measure how much of a statistical detector's accuracy on it is document length. Word count alone reaches AUROC 0.942. Our 19-feature ensemble reaches 0.987, so a single feature recovers 91% of the ensemble's signal above chance. The strongest test uses a split in the release where human documents are the shorter class. There the ensemble does not drift toward chance. It inverts, to 0.289, and matching on length inside the split brings it back to 0.853. Fast-DetectGPT and Binoculars at GPT-2 scale, scored on the same split, do not invert (0.969 and 0.977). The same artifact explains two results that would otherwise pass as findings: accuracy that rises with rewrite depth, where the ensemble beats word count by at most 0.0026, and an apparent DeepSeek-R1 evasion (0.562) that disappears once reasoning traces are stripped (0.994). Out of domain the ensemble keeps most of its ranking (0.855), yet the threshold that gives 1% false positives in domain flags 85% of held-out human documents. A fine-tuned RoBERTa (0.947) and Binoculars at Falcon-7B scale (0.849) hold up out of domain, so the problem is the way rewrite benchmarks are evaluated, not statistical detection itself. We package the controls as the **Confound Attribution Protocol (CAP)**, run it on fifteen detector configurations, and report the two steps that do not work. Our recommendation is simple: report every rewrite-based detection result next to a length-only baseline on the same rows.[1]

1 INTRODUCTION

Language models now produce enough text to end up in the training data of their successors. Training on that output leads to model collapse, which has been analyzed formally (Shumailov et al., 2024) and shown to persist even when synthetic data is a minority of the training set (Dohmatob et al., 2025; 2024). Any mitigation starts with measurement. Without a way to audit a corpus, no one can say whether it is contaminated, or how badly.

Statistical detectors built on features such as perplexity, lexical diversity and burstiness are an attractive auditing tool. They need no access to the generating model, they are cheap to run, and several report high accuracy on standard benchmarks. Before trusting one, it is worth asking how much of that accuracy comes from the benchmark. If the way a benchmark is built separates the classes for reasons unrelated to how the text was produced, every detector evaluated on it looks better than it is.

We build a benchmark that varies contamination ratio, source model scale and recursive generation depth together, and we use it to measure how much of a statistical detector's in-domain accuracy is document length. Most of it is. The damage shows up in two places an in-domain AUROC does not reveal: the decision threshold, which does not survive a change of domain, and the direction of the ranking, which flips when the length relationship does. The measurement also matters for

[1] Benchmark, provenance metadata, Croissant schema and CAP code are in the anonymized supplementary material and will move to a permanent public location on publication.

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.

model-collapse research. At rewrite depths 2 to 5, word count alone reaches 0.996 or higher, so a statistical detector's success on recursively generated text says little until it is compared with a length baseline.

Contributions. (1) WebMix-Collapse: 10,668 documents at six contamination levels, from ten model families (1B to 671B parameters) and five rewrite depths, with provenance metadata, a Croissant schema and a plain account of what it cannot answer (Section 3). (2) The Confound Attribution Protocol (CAP). Its five controls have appeared separately in prior work (Ippolito et al., 2020; Wang et al., 2024; Wu et al., 2024; Bao et al., 2025); we assemble them into one diagnostic, release it as code and run it on fifteen detector configurations. The word-count baseline, length matching and a reversed-length split work. Residualization depends on how a score is represented, and the per-domain correlation orders nothing (Section 5.2). (3) An out-of-domain sweep over four held-out domains, in which the ensemble keeps a usable ranking (0.855) while its in-domain threshold fails completely, with CAP applied to every configuration whose per-document scores we kept, on one shared row set (Sections 5.3 and 5.5). (4) A reversed-length split, found in the release, on which the ensemble inverts to 0.289 while Fast-DetectGPT and Binoculars score 0.969 and 0.977 on the same split; and the tracing of an apparent DeepSeek-R1 evasion to retained reasoning traces (Sections 5.6 and 5.4). (5) A reproducibility finding: the released feature code holds four inconsistent implementations of the same nineteen features, and a single unit convention between two of them moves out-of-domain AUROC from 0.855 to 0.615 (Appendix V). Our generated documents, metadata and code are released under CC BY 4.0; the human documents keep C4's ODC-BY terms. Everything except DetectGPT, the LLM judge and the two larger Binoculars pairs runs on a CPU.

2 Related Work

Model collapse and statistical detection. Shumailov et al. (2024) showed that recursive training on synthetic outputs causes distributional divergence; Dohmatob et al. (2025) proved collapse persists under minority contamination, Dohmatob et al. (2024) framed it as a shift in scaling laws, and Gerstgrasser et al. (2024) showed that keeping the real data and accumulating synthetic data beside it, instead of replacing it, avoids collapse in the settings they test. WebMix-Collapse targets the measurement side of that problem, the tools used to check whether a corpus is already contaminated. Zero-shot detectors read log-probability curvature (DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2024)) or cross-perplexity (Binoculars (Hans et al., 2024)); GLTR (Gehrmann et al., 2019) visualizes token statistics, RADAR (Hu et al., 2023) trains adversarially, and Sadasivan et al. (2023) prove limits on detection as models improve. Ghostbuster (Verma et al., 2024), a structured feature search with a linear classifier, is the closest relative of SynDet-Lite; we do not re-implement its search. These methods are mostly evaluated in domain, where length-correlated approaches look best. Kuznetsov et al. (2024) and Borile and Abrate (2025) improve out-of-distribution robustness by removing harmful embedding subspaces and confounding neurons; the latter targets domain confounders and explicitly sets aside linguistic ones such as sentence length. Both build better detectors; we measure how much of a detector's reported accuracy the length confound explains.

Length is known to matter, and benchmarks handle it in two ways. Ippolito et al. (2020) showed that human and automatic detection both improve with longer excerpts. One line of benchmarks equalizes length when the data is built. M4 (Wang et al., 2024) controls generation length, HART (Bao et al., 2025) gives the source's word and paragraph count in the rewrite prompt and then truncates, and a concurrent benchmark matches each human answer to a model answer within 20% of its word count, calling the length confound well known (Baidya et al., 2026). The other line leaves length free and reports the gap: RAID (Dugan et al., 2024) prints a mean of 378.5 tokens for human documents against 323.4 overall, and DetectRL (Wu et al., 2024) evaluates detectors in 20-word length intervals. Neither line measures the confound. Equalizing length removes the signal along with any way to size it, and reporting the gap does not attribute performance to it. None of these benchmarks reports a length-only baseline or a before-and-after comparison, so the field knows about the problem without knowing its size. BAID (Basu et al., 2025) audits detector bias over demographic and stylistic categories; length is not one of them, and the paper notes without measuring it that ZipPy and other statistical detectors are sensitive to input length. We treat length as a controlled variable instead of a nuisance to suppress, which is what makes a word-count-only baseline meaningful, and one split of our release runs the length relationship the other way, so a detector's ranking can be watched as it turns around.

Dataset bias and watermarking. Torralba and Efros (2011) documented systematic biases in vision benchmarks and Niven and Kao (2019) showed that BERT's near-human score on an argument comprehension task came entirely from spurious statistical cues. The length confound follows the same pattern: high in-domain accuracy hides a failure that only shows under distribution shift. Watermarks (Kirchenbauer et al., 2023; Dathathri et al., 2024) avoid the problem but need the generator's cooperation; we target the auditing case, where only raw text is available.

3 WEBMIX-COLLAPSE BENCHMARK

3.1 DESIGN RATIONALE

Real contamination varies along at least three dimensions that no existing benchmark controls together: the fraction of synthetic documents, the capability of the source model, and how far synthetic text has propagated through repeated rewriting. WebMix-Collapse controls all three. Table 16 in Appendix Q compares it with the two most relevant prior benchmarks, RAID (Dugan et al., 2024) and HART (Bao et al., 2025).

Contamination levels. Six levels (0%, 5%, 10%, 20%, 40%, 60%) run from minority contamination, where Dohmatob et al. (2025) show collapse still occurs, into the majority-synthetic regime.

Model scale range. Ten families from 1B to 671B parameters test whether detectors trained on small local models transfer to large API models, which is the situation an auditor is in.

Recursive generation depth. Model collapse is about iterated generation, so we include five rewrite depths; no existing detection benchmark has this axis.

What the benchmark cannot answer. The rewrite prompt is uniform, so detectors may behave differently under varied prompting or long-form generation. The detector sweep changes domain and generator mix together and cannot separate the two (Section 3.3). No detector is swept on the released GPT-4o-mini and GPT-4o rewrites, although Section 3.3 checks their lengths, and closed-weight models such as Claude and Gemini are absent. Two of the three axes are also thinner than their ranges suggest. One 8B model supplies 87% of Generation 1, so the scale axis says little beyond that model, and AUROC ignores base rate, so the contamination axis is read through threshold metrics (Appendix O). The length confound is the result the benchmark supports best, and the paper is organized around it.

3.2 CONSTRUCTION

Seed documents. The seeds are C4 (Raffel et al., 2020) documents filtered to 200 to 1,000 words. The released pool holds 2,000 of them at a mean of 439 words. The benchmark's human class is larger: 5,298 rows over 4,616 distinct C4 documents, of which 3,838 carry a paired rewrite and the rest serve as human examples in the contamination splits. C4 comes from a 2019 Common Crawl snapshot, so every seed predates every generation model we used. That snapshot is also in the pretraining data of many of them, so rewrites may partly reflect memorized content (limitation 6).

Generation 1. Seeds were rewritten by ten model families using the prompt *"Rewrite the following text in your own words, preserving the meaning and approximate length."*[2] Six families ran locally under Ollama at 1B to 4B parameters, one was hosted on Groq (Llama-3.1-8B) and three were served by SambaNova (Llama-4-Maverick 17B, Llama-3.3-70B and DeepSeek-R1 671B); Appendix F lists them. They produced 4,214 documents in unequal shares. Llama-3.1-8B supplied 3,650 (87%; 3,619 and 31 under its two serving names) and the other nine between 30 and 245 each, because of rate limits, generation failures and quality filtering. The imbalance is recorded in the provenance metadata, and it limits what any per-family reading can support.

Generations 2 to 5. Feeding rewrites back through three of the local models gives 1,156 more documents at depths 2 to 5 (296, 287, 287 and 286). Each document's identifier records its rewrite chain.

Contamination splits and full corpus. Six splits of 500 documents each hold fixed contamination rates (Appendix N) inside a corpus of 10,668 documents, 5,298 human and 5,370 synthetic. Detectors

[2]The length instruction is there to test compliance, and models do not comply; see **Length artifact** below.

are trained on the 9,512 documents at Generations 0 and 1, and the 1,156 at depths 2 to 5 are held out as a separate evaluation axis (Section 5.4).

Length artifact. Despite the instruction to preserve approximate length, Generation-1 synthetic documents average 193 words against 439 for the 2,000-document seed pool, a 2.3× gap (Appendix E). Over the 9,462 rows every in-domain result is computed on, the means are 422 and 194 words, and we quote that ratio, 2.18×, wherever a detector is evaluated. The compression is not one model's habit. For the 1,303 rewrites whose seed is in the released pool, every one of the ten families shortens its input, by 1.86× (Phi3-Mini) to 2.92× (Gemma2-2B) in mean length (Appendix G). It is still a property of this prompt, not a law of rewriting: HART (Bao et al., 2025) equalizes length by giving word counts and truncating. We leave the gap in, because it is what lets us size the confound (Section 5.2).

3.3 OOD Evaluation Set

The OOD set draws on four HuggingFace datasets covering news, product reviews, encyclopedic text and film reviews (AG News, Amazon Reviews, DBpedia and IMDb), with 800 human documents sampled from the original splits. The detector sweep uses 678 synthetic documents from three small open-weight models served locally through Ollama (Qwen2.5-3B, Gemma2-2B, Llama-3.2-3B) under the same rewrite prompt. The release also carries 799 rewrites from Llama-3.1-8B and 800 each from a second Qwen2.5-3B run, GPT-4o-mini and GPT-4o, which no detector is swept on here. The three swept models also appear in the in-domain training set, though thinly, with 103 of its 4,214 synthetic documents. This is therefore not a held-out generator subset: the sweep changes domain and generator mix at once, and we draw no conclusion about either shift alone. On these rows human documents average 125 words against 99 synthetic, an H/S ratio of 1.26×, and 1.30× over the 1,074 documents where all five feature families extract, far below the in-domain 2.18×. The length cue is weaker in absolute terms too: word count alone falls from 0.942 to 0.577. Every generator we released, the commercial ones included, returns these short sources only slightly shortened. On the Llama-3.1-8B, second Qwen2.5-3B, GPT-4o-mini and GPT-4o files the H/S ratio is 1.10 to 1.19 and word count alone scores 0.500 to 0.558, while the same Llama-3.1-8B shortens the long in-domain seeds 2.40× (Appendix G).

CAP Step 4, which correlates each domain's H/S ratio with the AUROC reached there, gives -0.21 for the ensemble across these four domains and $+0.96$ across the four categories of the reversed-length split (Section 5.6); both are descriptive, on $n = 4$ (Appendix AC).

4 SynDet-Lite

SynDet-Lite is a reference statistical detector, built to expose benchmark artifacts and not to compete with published detectors. Its behavior does not generalize to detectors as a class (Section 5.3).

It describes a document with nineteen features from five signal families. Four read the perplexity profile of a GPT-2 117M scorer (S1), small enough to run on a laptop CPU. Four measure lexical diversity (S2): type-token ratio, hapax ratio, Yule's K and bigram entropy (Yule, 1944). Three measure burstiness (S3), after Altmann et al. (2009), four the depth of a spaCy dependency parse (S4), and the last four the document's self-similarity (S5) through its compression ratio, trigram repetition and mean cosine similarity between paragraphs. A 5-fold GroupKFold XGBoost (Chen and Guestrin, 2016) classifier (200 estimators, depth 6, learning rate 0.1) groups each seed with its rewrites, so a document and its own rewrites never share folds. Identical texts published under two identifiers still can, and Section 5.1 measures what that is worth. The top feature is bigram entropy (33.6% importance, $r = 0.93$ with document length), which is what prompted the analysis in Section 5.2.

5 Experiments

5.1 In-Domain Detection

Full in-domain results are in Appendix B. The re-implemented baselines there use a GPT-2 117M scorer, well below published configurations, and are included only to order the feature families; no claim depends on them.

SynDet-Lite reaches AUROC = 0.987 over 9,462 documents.[3] The classifier matters little. Grouped by ID, XGBoost reaches 0.989, polynomial logistic regression 0.979 and logistic regression 0.957, so the ceiling is set by the features.

Byte-identical duplicates in the in-domain set. The released in-domain set contains 679 pairs of byte-identical human documents, each published under two IDs. Because the grouping rule reads the ID, 523 of those pairs straddle a fold boundary. Merging identical text into one group moves the estimate from 0.9883 to 0.9871 on the same rows, and dropping one copy of each pair gives 0.9860 on the remaining 8,783 documents, so the duplication is worth about 0.001 AUROC (Appendix K). The code release groups by text as well as by ID; the data files still carry the duplicates, and users should know that.

5.2 The Confound Attribution Protocol and the Length Confound

We introduce the **Confound Attribution Protocol (CAP)**, a reusable five-step procedure for deciding whether benchmark performance reflects detection or a dataset artifact. Steps 1 to 4 are correlational and need only a per-document score and a word count, so they apply to a detector distributed as weights or behind an API. Step 5 needs a split where the length relationship runs the other way. The steps make an attribution argument, not a ranking. Each holds length fixed in a different way, and what matters is never a detector's number after a control but the gap between that number and the same control applied to a detector that sees only length. The steps do not always agree, and Section 5.3 reports where they disagree and which ones we would keep.

CAP: Confound Attribution Protocol (for rewrite-based detection benchmarks)

Core tests (Steps 1, 3, 5).

Step 1. Run a word-count-only classifier. If AUROC > 0.90, a length confound is present, and this baseline becomes the control every later step is read against.

Step 3. Evaluate on a caliper-matched subset, and report the same subset for the word-count baseline. Compare the two drops.

Step 5 (new). Find or construct a split where the length relationship is reversed. A detector reading length ranks the classes backwards there, and matching on length inside the split restores it. This is the only step whose prediction can fail. A found split is not randomized, so show that the reversal does not follow from whatever else changed with it.

Supporting readings, with known failure modes (Section 5.3).

Step 2. Regress length out of the features, or out of the score, and read the result on ranks. It gives an upper bound on non-length signal, not a measurement.

Step 4. Correlate the H/S length ratio with AUROC across held-out domains. With four domains it orders nothing; we keep it only as a description.

Recommended additions: (6) OOD evaluation with bootstrap CIs; (7) a threshold carried from the in-domain operating point, since a detector can keep its ranking and lose its calibration; (8) a fine-tuned transformer upper bound.

Feature-to-length correlations. The three most length-correlated features, two of them also the most important overall, together carry more than half of total importance: bigram entropy ($r = 0.93$, 33.6%), type-token ratio ($r = -0.67$, 14.9%) and hapax ratio ($r = -0.59$, 3.0%). Perplexity and syntactic features have near-zero length correlation (r between 0.02 and 0.15; Appendix M).

Step 1. Word count alone reaches 0.942 against SynDet-Lite's 0.987, which is 91% of the signal above chance (0.442 of 0.487). The ensemble's entire advantage over that one feature is $+0.045$ [0.041, 0.050] on a component bootstrap. Adding word count back as an explicit 20th feature moves it only from 0.989 to 0.991 (grouped by ID), so the nineteen features already carry nearly all of it.

Steps 2 and 3. Residualizing all 19 features on document length gives 0.964, or 0.898 when length enters as its logarithm and folds are grouped as in the code release (Appendix C). This step bounds the non-length signal from above; it does not measure it. Step 3 is the sharper control, because the same operation applies to the ensemble and to the word-count baseline. On the 2,908-document

[3] Grouping folds by document ID gives 0.9885 ± 0.0014, and 0.9883 recomputed from the caches; grouping by connected component, as the release's shipped out-of-fold predictions do, gives 0.9869. Every CAP cell and bootstrap interval here uses those predictions, so we quote the lower 0.987 throughout; the duplicate pairs below account for the gap (Appendix K). The release's historical ablation file reports 0.981, which neither fold rule reproduces (Appendix L).

Table 1: Confound Attribution Protocol (CAP) results, each reading paired with the word-count-only control on the same rows: a step that costs the ensemble little and the control a great deal has isolated the confound, while a step that costs both the same has changed the difficulty of the data. "% signal" is $(x - 0.5)/(0.9869 - 0.5) \times 100$ for each AUROC x; negative means the ranking is inverted. Step 2 residualizes each of the nineteen features on log word count before training, so word count alone has no counterpart there. Step 3 uses the caliper $\max(5 \text{ words}, 5\%)$; Appendix X sweeps it. Every cell uses the extractors that wrote the training caches on both sides of every comparison (Appendix V).

		AUROC		% signal	
Setting	n	**19 feat.**	**word count**	**19 feat.**	**w.c.**
In domain, out of fold	9,462	0.987	0.942	100%	90.8%
Step 2: features residualized	9,462	0.898	–	81.7%	–
Step 3: length-matched	2,908	0.910	0.599	84.1%	20.3%
Out of domain, four domains	1,074	0.855	0.577	72.9%	15.8%
Step 3 within out of domain	802	0.828	0.525	67.4%	5.1%
Falsification: length inverted	680	0.289	0.091	−43.3%	−84.1%
Step 3 within the inverted split	180	0.853	–	72.5%	–

caliper-matched subset the ensemble reads 0.910 while word count alone falls to 0.599, so matching costs the ensemble 0.077 and the pure length detector 0.343. A control that moved both equally would be measuring how hard the matched region is, not the confound.

Both of the following hold. Length accounts for most of what is *detectable* in domain: one feature recovers 91% of the signal above chance, and the other eighteen add no more than 0.045. Yet the ensemble is not only a length detector, because inside a matched region where word count is close to useless it still reads 0.910. The cost of the shortcut appears where a detector is used. Sections 5.5 and 5.6 locate it in the calibration, which does not transfer, and in the direction of the ranking, which reverses when the length relationship does. Table 1 collects the steps, and Figure 1 in Appendix C plots them.

5.3 CAP Applied to Detectors We Did Not Build

A protocol shown only on the authors' own detector proves little, so we ran CAP on all fifteen configurations whose per-document scores we kept, nine of them detectors we did not design. Table 19 in Appendix U has every cell and the runs behind it, Appendix W repeats every row on one shared row set so that method and rows are not varied together, and Appendix D gives Step 4 per domain.

Running the protocol on fifteen configurations exposed two limits. **Step 2 depends on how the score is written down.** Subtracting a least-squares fit does not preserve ranks, and AUROC depends on nothing else. Six rows move by more than 0.009, five of them saturated probabilities, the largest being the recomputed ensemble out of domain at 0.844 on ranks against 0.551 on the score. Even on ranks the word-count control keeps 0.729, since a linear fit cannot remove a nonlinear function of length, so Step 2 bounds the length contribution without measuring it. **Step 4 fails, and we report the failure.** Across the four out-of-domain domains the ratio-to-AUROC correlation is -0.21 for the recomputed ensemble and $+0.67$ for the fine-tuned transformer with the least length dependence we measured. Over the eleven out-of-domain rows that are measurements it spans -0.58 to $+0.69$ and orders the detectors in a way none of the other steps supports. Four domains that differ in topic, register and length ratio at once cannot separate the length effect from the domain effect, and a four-point correlation has no power to try.

Step 3 works. It is invariant to rescaling and applies unchanged to the detector and to the word-count control, which is what tells a reader how large a drop is. In domain, matching moves all three rows, and it moves the control more than four times as far as the ensemble (-0.343 against -0.077). Out of domain it moves ten of the twelve rows by less than sampling noise, the ensemble's -0.027 among them (Appendix D.1). The largest moves are upward, for the two DetectGPT rungs, whose scores track word count more strongly than any other out-of-domain row ($\rho \approx +0.5$): 0.648 to 0.701 and

0.643 to 0.693. Here the synthetic documents are the shorter class, so a score that rises with length works against them. The length confound in this benchmark is an in-domain phenomenon.

5.4 Model Scale and Generation Depth

Training on the 1B to 4B models and testing on the larger held-out ones gives 0.981 (Table 14). A leakage-safe GroupKFold analysis stratified by source model spans 0.886 to 0.997 across eleven models on 27 to 3,579 documents each, with only Phi3-Mini below 0.94 (Appendix P). DeepSeek-R1 (671B) looks like an exception at 0.562, but 217 of its 245 documents keep their `<think>` reasoning traces, which lengthen them to 368 words against 422 for their length-matched human comparison, where word count alone scores 0.513. Strip the traces (135 words) and word count scores 0.996 and the full detector 0.994. The release ships both variants, with the original rows untouched so the published checksums still verify (Appendix J). With no non-reasoning model at that scale, we read this as a second demonstration of the length confound rather than a statement about frontier scale (limitation 2). Length also falls steadily with rewrite depth, from 193 words at depth 1 to 120 at depth 5, and detection rises as it falls. At depths 2 to 5, which no released evaluation covers, word count alone reaches 0.996 to 0.9996 and the feature ensemble beats it by 0.0003 to 0.0026 (Appendix AA). Accuracy climbs with depth for reasons that have nothing to do with detecting synthetic content. All depth documents descend from 100 seeds and each depth has a single rewriter, so this is one instance of iterated rewriting, not an estimate of a general depth effect.

5.5 Out-of-Distribution Evaluation

Table 19 (Appendix U) reports every detector on the out-of-domain set of Section 3.3. Row sets differ (1,478 documents, a shared 500-document subsample, and 1,074 for the feature ensemble, whose extractors decline a document whenever any of the five families does), so Appendix W repeats the comparison on those 1,074 rows, and the paired figures below come from there.

The ranking transfers; the calibration does not. Recomputed with the extractors that wrote its training caches, the ensemble reaches 0.855 [0.830, 0.879] out of domain against 0.577 [0.544, 0.610] for word count alone on the same rows, and 0.828 after length matching. On the shared rows it sits 0.115 [0.089, 0.141] below RoBERTa, $+0.003$ [-0.029, $+0.035$] from Binoculars at Falcon-7B scale and 0.278 [0.243, 0.313] above word count. On the fifteen features our port reproduces it reaches only 0.776 [0.747, 0.803], below Binoculars rather than level with it. Its probabilities are another matter. 86.0% of them exceed 0.99, and the threshold that gives a 1% false positive rate in domain flags 85.0% of the out-of-domain human documents. Even a threshold chosen on the out-of-domain rows themselves, which no deployment can do, gives only 31.2% recall at a 1% false positive rate, against 34.7% for RoBERTa and 82.0% for the same ensemble in domain (Appendix Y). As a ranker the detector keeps most of what it had. As the decision rule a corpus audit needs, it is unusable, and AUROC alone would never show it.

A correctly implemented statistical detector does not collapse. Binoculars, implemented as published (Hans et al., 2024), reaches 0.722, 0.787 and 0.849 on the same 1,478 documents as its observer and performer scale from GPT-2 to Falcon-7B. Scorer scale is worth 0.127, and only the top rung is level with the feature ensemble. DetectGPT, which reads probability curvature rather than length-adjacent features, reaches 0.648 at the 100 perturbations prescribed by Mitchell et al. (2023), and a quarter of that budget is indistinguishable from it (0.643; Appendix D.1). Fast-DetectGPT (Bao et al., 2024) gives 0.768 and 0.843, and a Qwen2.5-32B judge 0.817 with a score-to-length correlation of -0.02, against -0.37 for the ensemble. Statistical detection as a class does not fail out of domain. Threshold transfer does, and the zero-shot detectors are no better there: Binoculars at Falcon scale recovers 29.8% of the synthetic documents at a 1% false positive rate chosen on these rows.

Per domain, the length explanation does not hold. AG News is the one domain where synthetic text is *longer* than human, 0.89× on the rows the ensemble covers, so word count alone scores 0.376 there, below chance. The ensemble scores 0.931 [0.871, 0.980] on those same 97 documents, its best domain of the four. IMDb, where the gap is widest in the training direction (1.71×, word count 0.766), gives 0.875. The ensemble's spread across the four domains, 0.110, matches RoBERTa's 0.105; the zero-shot detectors spread from 0.126 to 0.368. The release's shipped predictions, from a second implementation of the same features, put AG News at 0.345 (Appendix V). Under the extractors the model was trained on, the domain where the length cue actively misleads is the one

where this detector does best. This is the failure of CAP Step 4 seen from the domain side, and it argues against reading the out-of-domain ranking as length.

Fitting one signal family at a time, the out-of-domain loss follows how length-correlated the family's features are (rank correlation $+0.90$ on five points; lexical richness falls 0.316, perplexity 0.125; Appendix AB).

5.6 A Reversed-Length Falsification Split

CAP Steps 1 to 4 are correlational. They say how much of a detector's ranking coincides with length, not what the detector would do if length pointed the other way. We do not manipulate length to find out. CAP Step 5 instead looks for a split in which the length relationship is already reversed and checks the prediction that follows. The release holds one: 787 documents that the models wrote with no seed document, shipped beside the benchmark's 10,668 and disjoint from them (400 human, 387 synthetic). There the human class is the shorter one, 190 words on average against 523, an H/S ratio of $0.36\times$ against the benchmark's $2.18\times$. A detector that reads properties of synthetic text should not care about the reversal; one that reads length should rank the classes backwards. Because the split was found rather than built, domain and length change together, and the checks below separate them.

The ensemble ranks them backwards. On the 680 documents its extractors accept, it scores 0.289 [0.246, 0.333] against 0.091 [0.068, 0.116] for word count alone on the same rows, well below chance and most of the way to the inversion the length cue produces by itself. Matching on length inside the split recovers 0.853 on the 180 documents that match, a point estimate on a sample too small for a useful interval. That is close to the 0.828 the same detector reaches on the length-matched out-of-domain subset, so the split is not simply harder. The synthetic side has a fifth category, how-to guides, with no human counterpart; restricted to the four categories both classes share, the three figures are 0.293 [0.250, 0.338], 0.096 and 0.847, so composition does not produce the reversal. Two further checks tie it to length. Across the nine generators, the more words a generator writes, the more confidently the detector calls its output human (-0.95 by rank). And CAP Step 4, which orders nothing out of domain, orders this split at $+0.96$; the category whose classes come closest in length is the only one near chance (blog posts, $0.62\times$, 0.494).

The same test can be put to detectors we did not build. We scored the split with Fast-DetectGPT and Binoculars at GPT-2 scale, on the CPU with the package's runners (Appendix U.1). Neither inverts. They reach 0.969 [0.955, 0.981] and 0.977 [0.968, 0.985] on all 787 documents, where word count alone scores 0.068, and 0.963 and 0.988 on the 182 documents that match on length. Length matching moves the ensemble from 0.289 to 0.853 and leaves the two zero-shot detectors about where they were. The split can be ranked the right way round, and what turns the ensemble around is length.

The fifteen verified features give 0.231 [0.196, 0.270] here and 0.811 after matching, so the reversal does not depend on the four features we cannot reproduce. RoBERTa stays above 0.99 and DeBERTa reaches 0.990. The release's own figure for this split, 0.204, comes from a fourth implementation of the features read through a `StandardScaler` that the published in-domain path does not use (Appendix V); we list it and use the recomputation.

6 Discussion

What the feature ensemble learned. In domain, rewriting shortens documents (422 words human against 194 synthetic), and the two features the released model leans on hardest are the two that track word count most closely (Section 5.2). Out of domain the gap narrows from $2.18\times$ to $1.30\times$, and word count falls to 0.577 while the ensemble keeps 0.855. The ensemble is therefore more than word count with extra steps. Something in it survives the loss of the cue, and Appendix W puts that part level with the strongest zero-shot detector we ran. What does not survive is the calibration, nor the direction of the ranking once the length relationship reverses (Section 5.6). RoBERTa reads only its first 128 tokens, roughly 100 words. In domain that hides total length for every human document (all have at least 200 words) and for 99% of rewrites, but out of domain 58% of documents have 100 words or fewer and fit whole. Truncation alone therefore does not explain why RoBERTa transfers, and we have not tested what does.

Can de-biasing recover OOD performance? None of three strategies, dividing features by word count, keeping the least length-correlated families, or residualizing length, reaches transformer-level transfer in historical runs that are not comparable with the 0.855 above (Appendix Z), so length is not separable additive noise. In domain, residualizing costs 0.089 and the residualized model still reaches 0.854 on the length-matched subset against 0.599 for word count.

Implications for corpus auditing. An audit needs a threshold, not a ranking, and the threshold fails first. The operating point that costs 1% false positives in domain costs 85% out of domain, while AUROC moves by 0.13. The contamination axis shows the same thing from the other side. At one fixed operating point AUROC is nearly flat across the five contaminated levels (range 0.030), while precision moves from 0.53 at 5% contamination to 0.98 at 60% (Appendix O); the 5% figure rests on 10 synthetic documents out of 228. A contamination estimate should therefore come with the precision and recall of its threshold on the domain being audited. The R1 result adds a separate caution: an apparent evasion by a frontier model turned out to be an artifact of how reasoning-model output was handled, so evaluations like this one should control for formatting first.

Scope: paragraph-length user content. Every document is paragraph length: seeds of 200 to 1,000 words, rewrites at a median of 174, the out-of-domain set at 90. That is where automated screening is deployed, and it is the harder regime (Ippolito et al., 2020). Long-form generation is future work. Among the eight limitations in Appendix A, closed commercial detectors are untested, and four of the nineteen features cannot be reproduced from the released caches, so every recomputation is reported for all nineteen and for the fifteen that verify.

7 Conclusion

WebMix-Collapse shows that a statistical feature ensemble gets most, though not all, of its in-domain accuracy on a rewrite benchmark from document length. Out of domain its ranking holds (0.855) but its threshold does not (85% false positives), and where the length relationship reverses it ranks the classes backwards (0.289) until length is matched (0.853), while Fast-DetectGPT and Binoculars rank the same split correctly. RoBERTa, DeBERTa and Binoculars also hold up out of domain, so the fault lies in the evaluation, not in statistical detection. For rewrite-based benchmarks we recommend CAP's three core tests: a word-count baseline on the same rows, a length-matched comparison read against that baseline, and a reversed-length split. For auditing a corpus today, a fine-tuned transformer was the strongest tool we measured, and no detector's in-domain threshold should be carried to a new domain without checking it there.

AI use statement

This paper studies machine-generated text, so we distinguish two things that ICLR's policy treats differently: the generative models that produce the object of study, and the assistance used in preparing the paper.

The benchmark's synthetic half is model output by construction. Every synthetic document is a rewrite or original generation from the ten model families listed in Appendix F, and the generation prompts, decoding settings and provenance records are released with the dataset. This is the phenomenon the paper measures, not assistance with authoring it.

Assistance in preparing the work. The authors framed the research question, designed the protocol and its controls, and ran the experiments, with the one exception described below. Every stated number was checked against a result file in the release. LLM coding assistants helped implement parts of the feature-extraction, evaluation and Confound Attribution Protocol code, and helped with data cleaning, namely the mojibake repair described in Appendix K and the assembly of the out-of-distribution benchmark files. An assistant was used for literature retrieval, to find prior benchmarks and detectors that treat document length, and to draft passages of the related work and of the appendix sections describing the detector reruns. It also gave feedback on experiment design and on interpreting results, including the suggestion that the apparent DeepSeek-R1 evasion was a reasoning-trace length artifact, which we then confirmed by rerunning with the traces stripped. The exception: an assistant executed the package's own scripts for the length-by-generator counts (Appendix G) and for the two CPU zero-shot runs on the reversed-length split (Section 5.6). Before those runs, it repeated the recorded Fast-DetectGPT GPT-2 out-of-domain run on the machine it then used and reproduced the

recorded AUROC exactly. None of these runs samples anything, and the commands and outputs ship with the package. Prose was edited for readability, figure captions and table layouts were drafted and revised, the two figures were regenerated from stored result files, and candidate titles were proposed, all with assistance.

Verification and what it changed. No suggestion entered the paper before it was checked against a run of the code, and every cited work was read at the source, so the characterizations in the related work section are the cited papers' own words. Before submission, an assistant rechecked each reference entry, and each description of a cited work, against the published source. That check changed the paper: one work offered as a precedent for the length confound turned out to audit demographic and stylistic categories, not length, and was replaced by works whose treatment of length we could quote. No citation, number or table entry entered this paper on a model's word alone. The paper makes no mathematical claims, so no proofs were generated or checked by an assistant. Numerical claims were verified by rerunning the released code under the pinned environment, and the in-domain evaluation path was rerun four times, including from a clean extraction of the release archive at a different path, and the outputs were compared with diff, not by eye. The authors take full responsibility for the content of this work, including all text, claims and artifacts.

Ethics statement

The benchmark contains no human subjects data and no new human-written text: the human side is drawn from C4, a public web crawl, under its existing terms, and the synthetic side is model output. We inherit whatever personal information C4 contains and add none; documents are released as extracted, with provenance metadata; the documents we generated, the metadata and the code are under CC BY 4.0, and the C4-sourced human documents remain under C4's own terms.

We see two risks. First, a benchmark that documents how a class of detectors can be fooled is usable by someone who wants to evade detection. We judge the disclosure to be net positive, because the failure mode belongs to the evaluation protocol and is not a new attack, and because auditors currently have no way to tell an inflated in-domain number from a real one. Second, the length bias we measure has a likely fairness consequence, set out in Appendix A, item (8): a detector that keys on document length should flag short authentic writing more often. Liang et al. (2023) found that GPT detectors misclassify non-native English writing as machine-generated far more often than native writing; we have not tested whether the length cue contributes to that, and we report the risk as a reason for caution in deployment, not as a measured result.

Reproducibility statement

The release package contains the benchmark, the evaluation code, the provenance metadata and a Croissant schema, and is submitted as anonymized supplementary material. Section 3 states the construction protocol, Appendix G the out-of-distribution benchmark, Appendices H and I the transformer training settings, and Appendix R the dataset card. The environment is pinned in the package, down to the interpreter and library versions, because the in-domain figures move in the fourth decimal place under a numpy and scipy upgrade. The in-domain evaluation path was run four times: twice into separate directories with byte-identical output, once with the thread count pinned to one, and once from a clean extraction of the archive at a different path, and all outputs were diffed. Three numbers of our own cannot be recomputed from the package, and each is flagged where it appears with the reason: a first-run Binoculars GPT-2 value of 0.834, a first-run DetectGPT value of 0.653, and a first judge run of 0.797 whose prompt was not retained. Each is superseded by a released rerun, and nothing in the paper rests on them. Four figures for the feature ensemble in the release's historical files also cannot be recomputed, because the scripts that wrote them are not in the package; Appendix V sets out what they are, and the paper builds on none of them. Every CAP cell in Table 19 comes from a run whose per-document scores ship with the package, and a checker in the package refuses to pass if any cell disagrees with them.

References

E. G. Altmann, J. B. Pierrehumbert, and A. E. Motter. Beyond word frequency: Bursts, lulls, and scaling in the temporal distributions of words. *PLOS ONE*, 4(11):e7678, 2009.

G. Bao, Y. Zhao, Z. Teng, L. Yang, and Y. Zhang. Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *ICLR*, 2024.

G. Bao, L. Rong, Y. Zhao, Q. Zhou, and Y. Zhang. Decoupling content and expression: Two-dimensional detection of AI-generated text. *arXiv preprint arXiv:2503.00258*, 2025. Introduces HART, a hierarchical framework of AI-risk levels and its released dataset.

M. S. Baidya, S. S. Baidya, and C. Chawla. Detecting the machine: A comprehensive benchmark of AI-generated text detectors across architectures, domains, and adversarial conditions. *arXiv preprint arXiv:2603.17522*, 2026.

P. Basu, Y. Zhang, and V. Raheja. BAID: A benchmark for bias assessment of AI detectors. *arXiv preprint arXiv:2512.11505*, 2025.

C. Borile and C. Abrate. How to generalize the detection of AI-generated text: Confounding neurons. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 25461–25476, 2025.

T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *KDD*, 2016.

S. Dathathri, A. See, S. Ghaisas, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634:818–823, 2024.

E. Dohmatob, Y. Feng, P. Yang, F. Charton, and J. Kempe. A tale of tails: Model collapse as a change of scaling laws. In *ICML*, 2024.

E. Dohmatob, Y. Feng, A. Subramonian, and J. Kempe. Strong model collapse. In *ICLR*, 2025.

L. Dugan, A. Hwang, F. Trhlík, A. Zhu, J. M. Ludan, H. Xu, D. Ippolito, and C. Callison-Burch. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *ACL*, 2024.

T. Gebru, J. Morgenstern, B. Vecchione, J. W. Vaughan, H. Wallach, H. Daumé III, and K. Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, 2021.

S. Gehrmann, H. Strobelt, and A. Rush. GLTR: Statistical detection and visualization of generated text. In *ACL System Demonstrations*, pages 111–116, 2019.

M. Gerstgrasser, R. Schaeffer, A. Dey, R. Rafailov, et al. Is model collapse inevitable? Breaking the curse of recursion by accumulating real and synthetic data. In *COLM*, 2024.

A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, M. Goldblum, J. Geiping, and T. Goldstein. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text. In *ICML*, 2024.

P. He, X. Liu, J. Gao, and W. Chen. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *ICLR*, 2021.

X. Hu, P.-Y. Chen, and T.-Y. Ho. RADAR: Robust AI-text detection via adversarial learning. In *NeurIPS*, 2023.

D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck. Automatic detection of generated text is easiest when humans are fooled. In *ACL*, pages 1808–1822, 2020.

J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein. A watermark for large language models. In *ICML*, 2023.

K. Kuznetsov, E. Tulchinskii, L. Kushnareva, G. Magai, S. Barannikov, S. Nikolenko, and I. Piontkovskaya. Robust AI-generated text detection by restricted embeddings. In *Findings of the Association for Computational Linguistics: EMNLP*, 2024.

W. Liang, M. Yuksekgonul, Y. Mao, E. Wu, and J. Zou. GPT detectors are biased against non-native English writers. *Patterns*, 4(7):100779, 2023.

E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *ICML*, 2023.

T. Niven and H.-Y. Kao. Probing neural network comprehension of natural language arguments. In *ACL*, 2019.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140), 2020.

V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi. Can AI-generated text be reliably detected? *arXiv:2303.11156*, 2023.

I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, and Y. Gal. AI models collapse when trained on recursively generated data. *Nature*, 631:755–759, 2024.

A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

V. Verma, E. Fleisig, N. Tomlin, and D. Klein. Ghostbuster: Detecting text ghostwritten by large language models. In *NAACL*, 2024.

Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, C. Whitehouse, O. M. Afzal, T. Mahmoud, T. Sasaki, T. Arnold, A. F. Aji, N. Habash, I. Gurevych, and P. Nakov. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *EACL*, pages 1369–1407, 2024.

J. Wu, R. Zhan, D. F. Wong, S. Yang, X. Yang, Y. Yuan, and L. S. Chao. DetectRL: Benchmarking LLM-generated text detection in real-world scenarios. In *NeurIPS Datasets and Benchmarks Track*, 2024.

G. U. Yule. *The Statistical Study of Literary Vocabulary*. Cambridge University Press, 1944.

A Scope and Limitations

The limitations summarized in Section 6, in full.

(1) The detector sweep changes domain and generator mixture together across four domains and separates neither shift; the released commercial-generator rewrites (GPT-4o-mini, GPT-4o) are not yet swept, and Claude and Gemini are excluded. (2) The Generation-1 corpus is dominated by a single 8B model, so per-family conclusions beyond that model are correspondingly limited, and the only 671B condition is a reasoning model, which leaves scale and extended-RL chain-of-thought entangled until a DeepSeek-V3-Base condition is added. (3) The reversed-length split changes what the documents are about as well as which class is longer, so the confound cannot be fully eliminated. Four things bound it: the split's own length matching restores 0.853 from 0.289, restricting to the four categories both classes hold leaves 0.293, the per-generator AUROC tracks generator verbosity at a rank correlation of -0.95, and two zero-shot detectors rank the same split the right way round (0.969 and 0.977, Appendix U.1). What remains unexcluded is a content difference that happens to align with all four. (4) Claims apply to statistical feature ensembles on rewrite-based English benchmarks. Every document sits at paragraph length. The seed documents run 200 to 1,000 words by construction, but rewriting compresses, so the rewrites have a median of 174 words against the 365 of the human documents they were built from, and 74.5% of them fall below 200, and the out-of-domain set is shorter again, with a median of 90 words and 93.2% below 200. Per domain the out-of-domain medians are 139 words on IMDb, 107 on Amazon Reviews, 82 on DBpedia and 56 on AG News, so two of the four sets are platform-hosted user reviews and the regime this audit covers is the short user-generated text that review and comment screening operates on, which is also the harder regime, since detection degrades as excerpts shorten (Ippolito et al., 2020). Nothing here speaks to long-form generation, where the human and model length distributions, and therefore the confound itself, may look nothing like they do at paragraph length. That is future work, and extending the construction protocol that way is the benchmark's largest single gap; the title states the rewrite scope for that reason. (5) The detectors evaluated are the ones we could run and re-run locally: a feature ensemble, logistic regression on the same features, two fine-tuned transformers, Binoculars, DetectGPT, Fast-DetectGPT and an open-weight instruction-tuned judge. Commercial services such as GPTZero are excluded because a closed detector that changes without notice cannot be pinned to a version, which leaves open how proprietary detectors behave under the same length inversion. (6) Because C4 seeds predate and likely appear in the pretraining data of the generation models,

rewrites may partly reflect memorization; this does not affect the length-confound conclusion, which concerns length rather than novelty. (7) Four of the nineteen features cannot be reproduced from the released caches, because they come from a spaCy parse whose model version the release does not record, and the trained booster itself is not in the release, so every SynDet-Lite number recomputed here comes from refitting the published estimator on the published caches, since the original object is not available. Every recomputed figure is therefore reported for all nineteen features and again for the fifteen that verify (Appendix V); the two agree in direction everywhere and differ in level by up to 0.08. (8) The length bias probably has equity implications, though we have not measured them. A detector that reads word count and lexical diversity will tend to score short human text as synthetic, and threshold tuning cannot remove that without misclassifying short human text in general. Liang et al. (2023) report that GPT detectors flag non-native English writing as machine-generated far more often than native writing; whether length and lexical diversity drive part of that effect is a hypothesis our data cannot test, since the benchmark carries no author metadata.

B In-Domain Detection Results

Table 2: Detection methods on WebMix-Collapse (in-domain). † Recomputed here from the released feature caches, with the perplexity family scored by the release's GPT-2 117M; these rows order the feature families and are not comparable to published detector results. ‡ Logistic regression on the same 19 features as SynDet-Lite, once on the features themselves and once with quadratic terms added. The script that produced these two numbers is not in the release, so only the numbers can be reported; they are not a re-implementation of Ghostbuster, whose method is a search over features from weaker language models. The two single-family rows are the grouped recomputation described in Appendix L; the release's historical ablation file reports 0.961 and 0.752 for them, which that recomputation does not reproduce.

Method	**AUROC**	**F1**	**Prec.**	**Rec.**
Re-implemented baselines (GPT-2 117M; not comparable to published results)				
Word-count only†	0.942±0.003	–	–	–
S2 Lexical only†	0.954	0.855	–	–
Perplexity-only†	0.832	0.718	–	–
Statistical detectors				
SynDet-Lite (XGBoost, 19 feat.)	0.989±0.001	0.937	0.950	0.926
Logistic regression, 19 feat.‡	0.957±0.003	–	–	–
Logistic regression, quadratic terms‡	0.979±0.002	–	–	–
Learned detectors				
RoBERTa-base (fine-tuned)	0.997±0.001	0.977	0.995	0.959
DeBERTa-base (fine-tuned) (He et al., 2021)	0.997±0.001	0.984	–	–

C Confound Attribution Protocol Summary

Table 1 (in Section 5.2) summarizes all five CAP steps against the word-count control.

Scope of claims. The length-confound attribution is demonstrated for SynDet-Lite (a 19-feature XGBoost ensemble) and for logistic regression on the same 19 features, on a rewrite-based benchmark. Steps 1 to 4 also run on every other detector whose per-document scores we retained, including three Binoculars rungs up to the Falcon-7B pair (Table 19). Step 5 needs a split where the length relationship reverses, and the residualized-feature variant of Step 2 needs retraining, so both are available only for detectors we can retrain; a detector reached through an API gets Steps 1 to 4 and the score-to-length correlation. CAP is general and can be applied to any rewrite-based benchmark; the implementation is released with the benchmark code.

Cross-detector evidence and what is still missing. The lexical signal family, which supplies the largest share of SynDet-Lite's importance, has the highest mean feature-to-length correlation of the five at 0.60 and loses the most under domain shift, 0.316, whereas the Qwen2.5-32B judge scores correlate with word count at -0.02 and the judge reaches 0.817 OOD (Appendix T). Read together with the per-signal table in Appendix AB, where the OOD drop tracks each family's length dependence, this supports the mechanism claim across detector types, not only within one ensemble.

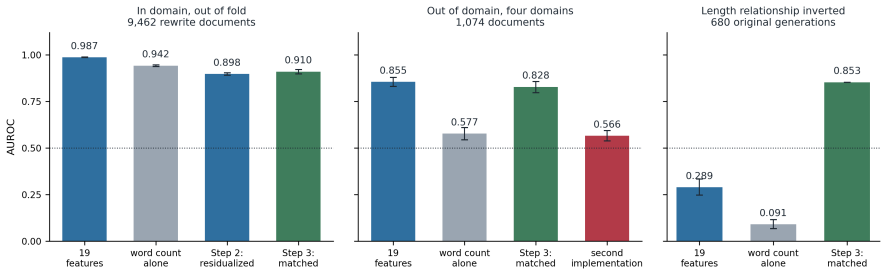

Figure 1: The nineteen-feature ensemble in three settings, with word count alone beside it in each. Left: in domain and out of fold on 9,462 documents, with CAP Step 2 (features residualized on word count before training) and Step 3 (caliper-matched subset, $n = 2{,}908$). Middle: out of domain on the 1,074 documents the training-time extractors score, with Step 3 on 802 and, in red, the release's own shipped predictions on 1,428 documents for comparison (Appendix V). Right: the reversed-length split of Section 5.6 on 680 documents, where both the ensemble and word count fall below chance and length matching restores the ensemble to 0.853. Error bars are the 95% intervals of Table 5: components resampled in domain, rows resampled within class out of domain. The matched bar in the right panel is a point estimate on 180 documents and carries none. Regenerated from the result files in the package by `experiments/scripts/make_paper_figures.py`, which prints every value it draws.

Two gaps remain. Every row of Table 19 is open weights or our own code, so how a closed commercial detector behaves under the same controls is untested. And Step 5, the one step that can fail, ran only on the detectors we can retrain: the reversed-length split has no scores from the zero-shot detectors or the transformers beyond their reported AUROCs, so what CAP establishes causally it establishes for the feature ensemble alone.

Step 2 depends on the assumed functional form of length. Residualization removes length by regressing each feature on document length and keeping the residual, so the number it produces depends on how length is assumed to enter. One linear regression per feature on raw word count gives 0.964. Fitting the same regression on log word count instead gives 0.910, and additionally grouping documents by connected component instead of seed ID, which is what the code release does, gives 0.898, the value Table 1 prints. The three values are recomputed on the same 9,462 documents and the same cached features by `audit/scripts/residualization_protocol.py`; the file it writes also records that the original script fitted the length regression on all rows including the held-out ones, which is a leak but is worth only 0.0002 here, and that repeated runs under shuffled row orders move the value between 0.9636 and 0.9643.

Step 2 depends on how the score is represented, and Steps 1 and 3 do not. We found this defect of the protocol while extending CAP across detectors. AUROC is a function of the ranks of a score alone, so replacing a score s by any strictly increasing $g(s)$ leaves the reported AUROC, Step 1 and the Step 3 matched AUROC unchanged: Step 1 never reads the score, and Step 3 changes which documents are scored rather than how they are ordered. Subtracting a least-squares fit is not rank preserving, so Step 2 carries no such guarantee. The judge shows the size of the effect. Its AUROC is 0.8165 on its probability, on the logit of that probability and on its rank, all three identical, and its Step 2 on those same three readings is 0.7445, 0.8135 and 0.8142. The positive control moves further in the other direction: the length-only logistic detector reads 0.7368 on its probability and 0.4715 on the logit, because its logit is close to linear in log word count and Step 2 removes exactly that. Step 3, which does not depend on the representation at all, puts the same detector at 0.599.

Two conditions produce it. A score packed against the ends of the probability scale cannot absorb a subtracted line without reordering: 61.4% of the judge's probabilities fall below 0.01, and ties are not the explanation, since the 500 documents still take 431 distinct values. And there has to be an ordering worth disturbing, which is why the release's shipped out-of-domain predictions are equally saturated at 56.9% of their mass outside $[0.01, 0.99]$ and yet move by 0.008: an AUROC of 0.566 leaves little to reorder. Table 19 therefore prints Step 2 twice, on the score and on its ranks, and the released CAP output adds the logit reading, the saturation fractions and a check that the logit leaves the AUROC unchanged, for every detector whose score is a probability. Nine of

the fifteen configurations agree between the score and rank readings to within 0.009. Five of the six that do not are bounded classifier probabilities: the out-of-domain ensemble at $+0.294$ on the training-time extractors and $+0.207$ on the fifteen verified ones, the judge at $+0.070$, the in-domain nineteen-feature ensemble at -0.053 and the fine-tuned RoBERTa at $+0.053$. These place 89.9%, 92.4%, 69.2%, 65.5% and 88.8% of their mass outside $[0.01, 0.99]$ against 19.5% for the length-only control and 9.6% for the residualized ensemble. The sixth is DetectGPT at 100 perturbations, which is over the line by 0.0002 and is not a probability at all. The out-of-domain ensemble is the case that matters, and it is the reason we report Step 2 on ranks: on the score it reads 0.551, which invites the reading that nothing survives residualization, and on ranks it reads 0.844 against an AUROC of 0.855, which is the invariant reading and agrees with the 0.828 of Step 3. Where the two disagree, Step 3 is the reading to trust, because it is invariant by construction, and the practical rule we would give someone applying CAP is to report Step 2 on ranks and treat Step 3 as the control that settles the question. What the three protocols agree on is that residualization leaves a lot standing: 95.2%, 84.2% and 81.7% of the signal above chance respectively, measured against the 0.987 of the unresidualized ensemble. Step 2 therefore bounds the non-length signal from above without measuring it, for two reasons. Subtracting a linear fit removes only the component of each feature that is linear in length, and a feature such as bigram entropy is not linear in length, which is exactly why the raw-length and log-length protocols disagree by 0.053. And the residual features are still evaluated on documents whose lengths differ by class, so any remaining nonlinear length dependence is still usable. The log-length protocol has the better claim on the reader: it is the more conservative of the two, and its 0.898 sits within 0.007 of the 0.891 that Step 3 gives under an exact-length caliper, whereas the raw-length 0.964 sits above every length-matched evaluation of the same detector, which is hard to read as a bound on the same quantity. Step 3 has no such ambiguity, because a length-matched subset only selects rows and leaves the scores alone. What it does have is a caliper: the same detector reads 0.891 at exact length, 0.910 at the max(5 words, 5%) caliper the paper uses and 0.948 at max(20 words, 20%), so the caliper has to be stated with the number, and Appendix X varies it, varies the order in which candidates are matched, and reports a length-stratified alternative that needs no caliper at all.

D Bootstrap Confidence Intervals and Per-Domain OOD

Table 3 gives the per-domain breakdown for the ensemble and the two transformers, and Table 4 adds the zero-shot detectors, every cell with a 95% interval. Each interval is 2,000 resamples of that domain's own rows, stratified by class, at seed 42, written by `experiments/scripts/cap_step4_cross.py`. The release's shipped out-of-domain predictions, which a second implementation of the same nineteen features produced (Appendix V), are kept as the last row group so the difference is visible.[4] Two points matter. AG News is the one domain where synthetic text is longer than human, so word count alone is well below chance there (0.376 on the rows the recomputed ensemble covers, 0.190 on the full set); the shipped predictions were at their worst on that domain, at 0.345, and the recomputation is at its best, at 0.931. And the recomputed ensemble's spread across the four domains, 0.110, is indistinguishable from RoBERTa's 0.105, while every zero-shot detector spreads wider, up to 0.368 for the judge.

Table 3: Per-domain OOD on the domain-shift set of Section 3.3, with 95% percentile intervals. The H/S ratio and the word-count-only column are properties of the rows a detector covers rather than of the detector, so they are printed per row group. Row sets differ: the transformers read text directly and cover all 1,478 documents, the recomputed ensemble covers the 1,074 where all five ported feature families scored, and the shipped predictions cover 1,428. DeBERTa kept per-domain AUROCs only. Appendix W makes the cross-detector comparison on one shared row set instead.

[4] The original tree also contains `bootstrap_ci_results.json`, whose overall SynDet-Lite figure is 0.600 [0.571, 0.629]. That file was computed on a different row set, no table in the paper is drawn from it, and it is shipped only so the difference is visible.

Domain	H/S	Word count	AUROC	95% CI	n
SynDet-Lite, 19 features, training-time extractors					
AG News	0.89×	0.376	0.931	[0.871, 0.980]	97
Amazon Reviews	1.09×	0.594	0.821	[0.771, 0.869]	304
DBpedia	1.01×	0.528	0.869	[0.825, 0.906]	319
IMDb	1.71×	0.766	0.875	[0.836, 0.911]	354
Overall	1.30×	**0.577**	**0.855**	**[0.830, 0.879]**	**1,074**
RoBERTa-base, fine-tuned					
AG News	0.75×	0.190	0.886	[0.846, 0.925]	303
Amazon Reviews	1.22×	0.701	0.981	[0.968, 0.991]	393
DBpedia	1.07×	0.634	0.991	[0.983, 0.997]	386
IMDb	1.84×	0.804	0.985	[0.972, 0.995]	396
Overall	1.26×	**0.548**	**0.947**	**[0.937, 0.958]**	**1,478**
DeBERTa-base, fine-tuned, AUROC only					
AG News	0.75×	0.190	0.941	–	303
Amazon Reviews	1.22×	0.701	0.982	–	393
DBpedia	1.07×	0.634	0.971	–	386
IMDb	1.84×	0.804	0.995	–	396
Overall	1.26×	**0.548**	**0.972**	–	**1,478**
SynDet-Lite, 19 features, shipped predictions, second implementation					
AG News	0.80×	0.240	0.345	[0.277, 0.414]	259
Amazon Reviews	1.22×	0.699	0.588	[0.534, 0.642]	390
DBpedia	1.07×	0.628	0.534	[0.478, 0.592]	383
IMDb	1.84×	0.804	0.746	[0.696, 0.797]	396
Overall	1.31×	**0.578**	**0.566**	**[0.538, 0.594]**	**1,428**

Table 4: CAP Step 4 for every out-of-domain configuration: per-domain AUROC and the spread across the four domains. The first two lines are properties of the rows, not of any detector: the human-to-synthetic mean word-count ratio, and the AUROC word count alone reaches on that domain. Detectors are grouped by the rows they cover. Every cell carries a 95% percentile interval in `experiments/results/cap_step4_cross.json`, from 2,000 resamples of that domain's own rows stratified by class at seed 42; those intervals are 0.014 to 0.116 wide on the 1,478-document rows, 0.076 to 0.131 wide on the recomputed ensembles, 0.101 to 0.137 wide on the 1,428 shipped-prediction rows and 0.062 to 0.214 wide on the 500-document rows, which is why no cell here should be read to its last digit.

Detector	n	AG News	Amazon	DBpedia	IMDb	Spread
H/S word-count ratio	1,478	*0.75×*	*1.22×*	*1.07×*	*1.84×*	–
Word count alone	1,478	*0.190*	*0.701*	*0.634*	*0.804*	*0.614*
All 1,478 documents						
RoBERTa-base, fine-tuned	1,478	0.886	0.981	0.991	0.985	0.105
Binoculars, Falcon-7B pair	1,478	0.904	0.836	0.663	0.959	0.297
Fast-DetectGPT, Falcon-7B pair	1,478	0.910	0.826	0.667	0.940	0.273
Binoculars, GPT-2-large / GPT-2-XL	1,478	0.830	0.705	0.712	0.896	0.192
Fast-DetectGPT, GPT-2 white-box	1,478	0.899	0.663	0.756	0.818	0.236
Binoculars, GPT-2 / GPT-2-medium	1,478	0.718	0.648	0.687	0.831	0.182
The 500-document class-balanced subsample, where the ratios are 0.76, 1.31, 1.09 and 1.91						
LLM judge, Qwen2.5-32B	500	0.936	0.821	0.593	0.960	0.368
DetectGPT, 100 perturbations	500	0.719	0.593	0.689	0.646	0.126
DetectGPT, 25 perturbations	500	0.709	0.560	0.662	0.649	0.149
The rows the training-time feature extractors score						
SynDet-Lite, 19 feat., as trained	1,074	0.931	0.821	0.869	0.875	0.110
SynDet-Lite, 15 feat., as trained	1,174	0.829	0.742	0.773	0.820	0.088
Shipped release predictions, second feature implementation						
SynDet-Lite, 19 features	1,428	0.345	0.588	0.534	0.746	0.400

Read across Table 4, CAP Step 4 does not survive as a diagnostic. If per-domain AUROC followed the length gap, the four cells of a row would rise from AG News to IMDb in step with the top line,

and the detectors that read length hardest would spread widest. Neither holds. AG News is the one domain where the ratio is inverted, at 0.75×, so word count alone is far below chance there, at 0.190; it is nonetheless the easiest of the four domains for five of the twelve configurations. Amazon Reviews sits in the middle of the ratios, at 1.22×, and is the hardest domain for seven of the twelve. IMDb has the largest gap, at 1.84×, and is the easiest for six configurations but only second for the recomputed ensemble.

The correlation Step 4 asks for therefore runs from -0.58 to $+0.96$ across these twelve rows, and its sign is close to the opposite of what the diagnostic assumes. The largest positive value belongs to the shipped predictions, at $+0.96$, whose scores fall with word count ($\rho = -0.42$) and which are near chance overall; the next two are Binoculars at GPT-2 scale, at $+0.69$, and RoBERTa, at $+0.67$, whose scores are essentially uncorrelated with word count ($\rho = +0.03$ and -0.04). The two most negative values, -0.58 and -0.40, belong to the two DetectGPT rungs, which are the two detectors here whose scores track word count most strongly ($\rho = +0.50$ and $+0.51$). A diagnostic meant to expose length dependence assigns its extreme positive value to a detector that reads length backwards and its extreme negative values to the two that read it hardest. Four points cannot separate these hypotheses, and the spread across domains is a property of what each domain contains rather than of how much length a detector reads.

D.1 Intervals for every row of the cross-detector CAP table

Table 19 prints point estimates, and several sentences in Section 5.5 compare one row with another. Table 5 gives 95% intervals for all fifteen configurations that carry CAP cells, on the full row set and on the Step 3 matched subset, with 2,000 resamples, seed 42 and percentile endpoints. Row sets differ across detectors here, so the intervals in a row are compared with each other and not across rows; Appendix W is where cross-detector comparisons are made. The resampling unit differs by setting, because the rows do. Out-of-domain rows are independent documents, so human and synthetic rows are resampled separately and each resample keeps the class balance, as in Table 4. In-domain rows are not independent: a source document and its rewrites share text, so the in-domain intervals resample the 3,855 connected components of Appendix K with replacement and take all rows of a drawn component together. That is the wider and the correct choice, and it is why the in-domain intervals here are not the narrow ones a row bootstrap would report. The matched subset is the one the caliper rule produces from the rows alone, so it is the same subset for every detector on a given row set and is bootstrapped as its own row set; the two intervals in a row are computed on different rows and their overlap is the comparison, not a paired test. Written by `experiments/scripts/bootstrap_cap_table.py`, which ships with the package.

Three comparisons are paired, since the runs in each pair scored the same documents. Pairing takes one resample and scores both detectors on it, which removes the between-document variance that dominates an unpaired comparison. Quadrupling DetectGPT's perturbation budget from 25 to 100 moves its AUROC by $+0.005$ with an interval of $[-0.009, +0.020]$, so on these 500 documents the two budgets are not separated at all, and the conclusion that the detector is weak here at its intended setting, not only in a cheap one, does not depend on which of the two numbers is quoted. Fast-DetectGPT at Falcon scale sits -0.006 below Binoculars at the same scale with an interval of $[-0.011, -0.001]$, so the gap is small but resolved. The third pair is the one that matters most for this paper: on the same 9,462 in-domain rows, the nineteen-feature ensemble beats log word count alone by $+0.045$ with a component-bootstrap interval of $[+0.041, +0.050]$. The ensemble is therefore doing something word count does not, and that something is worth 0.045 AUROC.

The table separates the detectors cleanly into two groups. For ten of the twelve out-of-domain rows, length matching moves nothing beyond sampling noise: the full-set value falls inside the interval of its own matched subset, the largest movements being the two DetectGPT rungs at $+0.053$ and $+0.050$ on 318 rows. The two exceptions are both SynDet-Lite: the fifteen verified features drop -0.044 to 0.732, and the shipped predictions drop -0.042 to 0.524. The nineteen-feature recomputation drops -0.027 to 0.828, which its own interval covers. For the three in-domain rows every full-set value falls outside its own matched interval, and the size of the drop separates them: word count alone loses 0.343 and lands at 0.599, the raw ensemble loses 0.077 and lands at 0.910, and the residualized ensemble loses 0.043. A control that moved every detector it touched would be measuring the control, not the confound; a control that moves the positive control by four and a half times what it moves the ensemble is measuring the confound.

Table 5: Bootstrap 95% intervals for all fifteen configurations that carry CAP cells, on the full row set and on the CAP Step 3 matched subset (2,000 resamples, seed 42). Out-of-domain rows are resampled by row with the class balance preserved; in-domain rows are resampled by component, all rows of a component together. The three 500-document rows share one class-balanced subsample. The last column marks whether the full-set AUROC lies inside the matched interval, which is the test of whether matching moved the detector at all.

Detector	n	**AUROC**	**95% CI**	n **matched**	**Step 3**	**95% CI**	**In**
Out of domain, rows resampled by class							
RoBERTa-base, fine-tuned	1,478	0.947	[0.937, 0.958]	1,034	0.957	[0.945, 0.969]	yes
SynDet-Lite, 19 feat., as trained	1,074	0.855	[0.830, 0.879]	802	0.828	[0.797, 0.857]	yes
Binoculars, Falcon-7B pair	1,478	0.849	[0.829, 0.868]	1,034	0.847	[0.823, 0.870]	yes
Fast-DetectGPT, Falcon-7B pair	1,478	0.843	[0.823, 0.862]	1,034	0.845	[0.822, 0.869]	yes
LLM judge, Qwen2.5-32B	500	0.817	[0.778, 0.851]	318	0.794	[0.742, 0.839]	yes
Binoculars, GPT-2-large / GPT-2-XL	1,478	0.787	[0.764, 0.811]	1,034	0.779	[0.750, 0.807]	yes
SynDet-Lite, 15 feat., as trained	1,174	0.776	[0.747, 0.803]	852	0.732	[0.698, 0.766]	no
Fast-DetectGPT, GPT-2 white-box	1,478	0.768	[0.745, 0.790]	1,034	0.777	[0.749, 0.805]	yes
Binoculars, GPT-2 / GPT-2-medium	1,478	0.722	[0.696, 0.749]	1,034	0.731	[0.699, 0.762]	yes
DetectGPT, 100 perturbations	500	0.648	[0.600, 0.693]	318	0.701	[0.644, 0.756]	yes
DetectGPT, 25 perturbations	500	0.643	[0.595, 0.689]	318	0.693	[0.637, 0.748]	yes
SynDet-Lite, 19 feat., shipped	1,428	0.566	[0.538, 0.594]	1,026	0.524	[0.486, 0.558]	no
In domain, out of fold, components resampled							
SynDet-Lite, 19 features	9,462	0.987	[0.985, 0.989]	2,908	0.910	[0.897, 0.922]	no
Word count alone (positive control)	9,462	0.942	[0.937, 0.946]	2,908	0.599	[0.577, 0.620]	no
SynDet-Lite, residualized features	9,462	0.898	[0.891, 0.904]	2,908	0.854	[0.841, 0.868]	no

E Vocabulary Statistics by Generation Depth

Table 6: Vocabulary statistics by generation depth (full corpus). Average length declines monotonically with depth (193, 142, 133, 128, 120 words), the mechanism behind the length confound. Unique words are lowercased whitespace tokens, so punctuation stays attached; the totals track the number of documents at each depth and are not comparable across rows of different size. Every cell is regenerated from the released files by `audit/scripts/corpus_depth_stats.py`.

Corpus	**Docs**	**Avg. Words**	**Unique Words**
Human seeds (Gen 0)	2,000	439	83,076
Synthetic Gen 1	4,214	193	73,777
Synthetic Gen 2	296	142	8,983
Synthetic Gen 3	287	133	8,778
Synthetic Gen 4	287	128	8,759
Synthetic Gen 5	286	120	8,257

F Model Families

Table 7: Ten model families in WebMix-Collapse. All are instruction-tuned variants; base models were not available via the APIs used for all families. DeepSeek-R1 is a reasoning model with a different inference-time behavior from the other families. The released `model` field carries eleven distinct strings rather than ten, because Llama-3.1-8B was served under two names, `llama-3.1-8b` and `llama-3.1-8b-instant`; Appendix P keeps those two apart, and every count of families in this paper treats them as one model.

Model	Params	Tier	Source	Type
Llama-3.2-1B	1B	Small	Ollama local	Instruct
Qwen2.5-1.5B	1.5B	Small	Ollama local	Instruct
Gemma2-2B	2B	Small	Ollama local	Instruct
Llama-3.2-3B	3B	Small	Ollama local	Instruct
Qwen2.5-3B	3B	Small	Ollama local	Instruct
Phi3-Mini	3.8B	Small	Ollama local	Instruct
Llama-3.1-8B	8B	Medium	Groq free tier	Instruct
Llama-4-Maverick	17B	Large	SambaNova free	Instruct
Llama-3.3-70B	70B	Large	SambaNova free	Instruct
DeepSeek-R1	671B	Frontier	SambaNova free	Reasoning

G OOD Benchmark Construction and Length by Generator

The OOD set is drawn from four HuggingFace datasets: AG News, Amazon Reviews, DBpedia and IMDb. For each domain, human documents (800 in total) are sampled from the original dataset splits. The detector sweep uses 678 synthetic documents produced by three small open-weight models served through Ollama (Qwen2.5-3B, Gemma2-2B and Llama-3.2-3B) under the same rewrite prompt. Those three models are not held out. They also appear in the in-domain training set, though thinly: 103 of its 4,214 synthetic documents come from them, against 3,650 from Llama-3.1-8B. What this set changes relative to training is therefore the domain and the generator mix at once, and no result here separates the two. The released OOD set also contains rewrites from Llama-3.1-8B (through Groq, 799 documents), a second, separate Qwen2.5-3B run (local Ollama, 800 documents), OpenAI GPT-4o-mini (800 documents) and OpenAI GPT-4o (800 documents, through the official API). No detector is swept on these four files; Table 8 checks their lengths.

Table 8 asks two questions by counting words, with no model involved. In domain, does every generator shorten what it rewrites, or only the one that supplies most of Generation 1? We can answer this for the 1,303 rewrites whose seed is in the released 2,000-document pool. Every family shortens its input, by 1.86× to 2.92× in mean length. Out of domain, where the sources average 125 words, every released file, the commercial generators included, has synthetic text only slightly shorter than the human side (H/S 1.10 to 1.26), and word count alone is near chance on every file (0.500 to 0.558). The same Llama-3.1-8B that shortens long in-domain seeds 2.40× leaves short out-of-domain sources at 1.13×. The in-domain gap is therefore shared across generators, and it appears on the long in-domain sources but not on the short out-of-domain ones, which also differ in domain, so we do not attribute it to source length alone. Written by `experiments/scripts/generator_length_check.py`.

H RoBERTa Training Details

RoBERTa-base (125M parameters) is fine-tuned for 1 epoch on the full 9,512-doc corpus (lr 2×10^{-5}, batch size 4, weight decay 0.01, max sequence length 128, CPU-only for M2 stability). 3-fold GroupKFold CV: AUROC = 0.997 ± 0.001, F1 = 0.977 ± 0.002; approximately 21 minutes per fold on M2 CPU. The 128-token cap means RoBERTa sees approximately the first 100 words of each document. For longer documents it therefore cannot read total length; for shorter ones, which include most of the out-of-domain set (median 90 words), the whole document fits and its length is visible.

I DeBERTa Training Details

DeBERTa-base (He et al., 2021) (139M parameters, GPT-2 BPE tokenizer, disentangled attention) is fine-tuned under identical hyperparameters to RoBERTa: 1 epoch, lr 2×10^{-5}, batch size 4, weight decay 0.01, max sequence length 128, CPU-only on M2. 3-fold GroupKFold CV: AUROC = 0.997 ± 0.001, F1 = 0.984 ± 0.004; approximately 29 minutes per fold. OOD AUROC = 0.972 [0.964, 0.980]; original-generation AUROC = 0.990 [0.981, 0.997]. Results close to RoBERTa's in all three evaluation settings show that its transfer is not specific to one architecture; two encoders of similar size are not enough to say more.

Table 8: Length by generator, counted with the paper's word rule. Top: in-domain rewrites whose seed is in the released pool, seed and rewrite mean words and the ratio of the means; the two serving names of Llama-3.1-8B are counted as one family. The DeepSeek-R1 row covers 26 rewrites with no reasoning trace, because the seeds of its 217 traced rewrites are outside the released pool (Appendix J). Bottom: each released out-of-domain file, human and synthetic mean words, their ratio and, in parentheses, the AUROC of word count alone (shorter means synthetic). The swept file is the one the detector sweep uses, with 678 rewrites from Qwen2.5-3B, Gemma2-2B and Llama-3.2-3B.

In domain	n	**Seed words**	**Rewrite words**	**Ratio**
Llama-3.1-8B	958	437.7	182.1	2.40
Llama-4-Maverick	70	440.9	193.6	2.28
Llama-3.3-70B	42	469.0	206.4	2.27
Phi3-Mini	38	414.3	222.6	1.86
Qwen2.5-3B	38	450.1	183.7	2.45
Gemma2-2B	35	472.3	161.5	2.92
Qwen2.5-1.5B	35	408.2	202.9	2.01
Llama-3.2-1B	31	399.8	140.6	2.84
Llama-3.2-3B	30	442.4	177.7	2.49
DeepSeek-R1	26	465.9	165.2	2.82
Out of domain	n	**Human words**	**Synth. words**	**H/S (AUROC)**
Swept file, three models	1,478	124.8	98.7	1.26 (0.548)
Qwen2.5-3B, second run	1,600	124.8	104.4	1.19 (0.554)
Llama-3.1-8B	1,599	124.8	110.1	1.13 (0.500)
GPT-4o-mini	1,600	124.8	113.1	1.10 (0.531)
GPT-4o	1,600	124.8	105.2	1.19 (0.558)

J DeepSeek-R1 Reasoning Traces

Of the 245 DeepSeek-R1 documents, 217 retained a balanced pair of `<think>` tags, 27 carried no tags at all, and 1 had an unbalanced tag. The released rows are unchanged, so the checksums in `data/croissant.json` keep verifying, and a stripped view ships alongside them under `derived/r1/stripped/`. The 217 balanced documents are stripped, the 27 untagged ones are copied through, and the single unbalanced document is passed through unmodified with its status recorded rather than repaired by guesswork, which means one document in the stripped view still carries partial reasoning text. Table 14 and the corrected scale-generalization result use the stripped view; the raw AUROC of 0.562 uses the released rows.

K Duplicate Text in the In-Domain Set

The 9,462-document in-domain set contains 679 pairs of byte-identical human documents. Each pair is the same C4 seed published under two identifiers, one bare and one suffixed, for example `c4_000000` and `c4_000000_human`. The grouping rule derives a group from the identifier, so the two copies of a pair land in different groups, and under 5-fold GroupKFold 523 of the 679 pairs fall on opposite sides of a fold boundary. That leaves 1,046 test rows whose training half contains an exact copy of them. No pair mixes labels and no synthetic document is duplicated, so the effect is confined to the human class.

`audit/scripts/duplicate_text_leakage.py` measures it under three grouping rules on the same features and the same estimator:

Table 9: Exact-duplicate leakage in the in-domain set, measured under three grouping rules.

Grouping	**Rows**	**AUROC**	**F1**
By seed identifier (as released)	9,462	0.9883	0.9375
By seed identifier and by text hash	9,462	0.9871	0.9320
One copy of each pair dropped	8,783	0.9860	0.9344

The first row is the same protocol as the headline in-domain number and reaches 0.9883 rather than 0.9885 because the two runs differ in XGBoost thread count, which moves the fourth decimal place and nothing above it. Closing the leakage costs 0.001 to 0.002 AUROC, so no conclusion in the paper turns on it, but the released data files still contain both copies of each pair and a user of them should know. The evaluation path in the code release groups by text hash as well as by identifier, so it does not inherit the problem, and the data files are left byte-identical to what the published checksums cover.

The second row is also what the release's own pipeline produces end to end. Running `python -m webmix folds` and `python -m webmix evaluate-cached` on the released files, with the mojibake rows repaired into `derived/` as described in Appendix R, gives 3,855 connected components over the 9,462 documents and a 5-fold mean AUROC of 0.9871 for the 19 features, 0.9422 for log word count alone and 0.8978 after per-fold length residualization, with 0.8797 inside the 219 to 240 word band. That output ships under `results/verified_indomain/` so a fresh run can be diffed against it rather than compared against a claim in prose; two runs of the sequence were byte-identical, a third with the thread count pinned to one produced identical metrics and identical per-document predictions, and a fourth from a clean extraction of the archive at a different path diffed clean against the shipped copy. Reproducing those digits needs the pinned versions in `requirements.txt`: numpy 2.2.6 in place of 1.26.4 moves the full-model value by 0.0001.

L Signal Ablation

Every cell below is recomputed from the released feature caches by `audit/scripts/ablation_grouped.py`, on the 9,462 documents in the cached intersection, with the 19 features, the XGBoost settings and the missing-value handling of the original script. Both fold rules are reported: 5-fold GroupKFold by seed id, which is the protocol the paper describes, and the 5-fold ungrouped `StratifiedKFold` used by the shipped `historical/results/ablation_results.json`.

The two protocols agree to about 0.0003 on the full model, so keeping a seed and its rewrites in the same fold is not what separates this table from the shipped one. That file is not reproducible from the released caches by the script that wrote it. Its full model is 0.007 low, and the per-configuration differences run in both directions, which is what different feature values look like rather than a different fold rule: it has 0.752 where the recomputation has 0.832 for perplexity alone, and 0.906 where the recomputation has 0.835 for self-similarity alone. The most likely cause is an earlier version of the feature caches, which was not recorded. Nothing in the paper's argument turns on the difference, since the ordering of the signal families is the same in both, but the shipped numbers cannot be presented as reproducible, so the recomputation is reported instead and the discrepancy is stated here.

Table 10: Signal ablation, recomputed under both cross-validation protocols on the 9,462-document in-domain intersection. Each cell is mean AUROC over folds with mean F1 in parentheses. The last column is the corresponding cell of the shipped `ablation_results.json`, which neither protocol reproduces.

	Recomputed here		**Historical**
Configuration	**GroupKFold**	**Ungrouped**	**shipped file**
Full model	0.988 (0.938)	0.988 (0.936)	0.981 (0.907)
Remove S1	0.975 (0.904)	0.975 (0.901)	0.972 (0.884)
Remove S2	0.957 (0.871)	0.957 (0.873)	0.957 (0.836)
Remove S3	0.988 (0.937)	0.988 (0.933)	0.982 (0.907)
Remove S4	0.986 (0.933)	0.986 (0.933)	0.980 (0.900)
Remove S5	0.986 (0.930)	0.986 (0.931)	0.981 (0.914)
Only S1	0.832 (0.718)	0.832 (0.718)	0.752 (0.526)
Only S2	0.954 (0.855)	0.955 (0.856)	0.961 (0.841)
Only S3	0.764 (0.645)	0.762 (0.645)	0.826 (0.596)
Only S4	0.785 (0.668)	0.785 (0.671)	0.771 (0.543)
Only S5	0.835 (0.721)	0.833 (0.722)	0.906 (0.711)

Read as a ranking, the recomputed table says what the historical file says. Removing the lexical family costs the most (0.031 AUROC, 0.067 F1) and no other family costs more than 0.013, while the single-family runs put lexical far ahead of everything else at 0.954. The lexical family is also the most length-correlated of the five, which is the point Section 5.2 makes with the CAP steps.

M FEATURE IMPORTANCES

Table 11: Top-10 XGBoost feature importances and Pearson correlation with document length. The two features with $|r| > 0.5$ and highest importance account for roughly half of total importance, explaining the source of the length confound.

Rank	Feature	Signal	Importance	Length r
1	lex_bigram_entropy	S2	33.62%	0.93
2	lex_ttr	S2	14.91%	−0.67
3	ppl_mean	S1	6.87%	0.15
4	ppl_variance	S1	4.97%	0.10
5	ppl_kurtosis	S1	4.72%	0.10
6	sim_compression_ratio	S5	4.27%	−0.58
7	syn_sent_len_var	S4	4.08%	0.02
8	lex_yules_k	S2	3.58%	−0.04
9	lex_hapax_ratio	S2	2.96%	−0.59
10	sim_trigram_rep	S5	2.58%	0.13

N CONTAMINATION SPLITS

Table 12: WebMix-Collapse controlled contamination splits. Each split is a stratified sample of 500 documents at the stated synthetic rate. The six splits were drawn independently rather than as one partition, so they are not disjoint; the overlap is quantified below the table.

Split	Total	Human	Synthetic	Rate
Clean	500	500	0	0.0%
5%	500	475	25	5.0%
10%	500	450	50	10.0%
20%	500	400	100	20.0%
40%	500	300	200	40.0%
60%	500	200	300	60.0%
Total	3,000	2,325	675	22.5%

Overlap between splits. The 3,000 split rows cover 2,048 distinct documents. 773 documents occur in more than one split (688 human, 85 synthetic), and pairwise overlap ranges from 46 documents (5% with 60%) to 120 (5% with clean). Each split is internally free of duplicates, and within a split the stated rate is exact, so a split is a valid sample at its own contamination rate. Two splits are not independent samples of each other, so a difference between two split-level numbers should not be read as a difference between disjoint document sets. The stratified analysis in Appendix P avoids this by assigning each document one contamination label and computing out-of-fold predictions once on the 9,462-document in-domain set.

Lineage of the split documents. The splits draw on 588 distinct synthetic documents. 376 of these are generation-1 rewrites that also appear in the training pool: 3,838 of the pool's 4,214 generation-1 rewrites fall in no split, and the per-split counts of pool documents are 14, 34, 61, 117 and 198 for the 5%, 10%, 20%, 40% and 60% splits (these sum above 376 because of the overlap just described). The remaining 212 are recursive rewrites at generations 2 to 5 (56, 53, 50 and 53 documents respectively) that appear in no other released file, so the split files are the only place those depths are published. Anyone rebuilding a train/test division from the release should therefore group by seed document across all of these files instead of treating the pool and the splits as separate populations.

Table 14: Scale generalization: train on 1B to 4B models only, test on larger models. No test model contributes a document to this training set, but seed documents are shared across the two sides, which is why the out-of-fold table below is the leakage-safe reading. The DeepSeek-R1 row uses reasoning-trace-stripped text; its raw value of 0.562 is a length artifact discussed in the text.

Test Model	Params	AUROC
Llama-3.1-8B	8B	0.978
Llama-3.1-8B (instant)	8B	0.986
Llama-4-Maverick	17B	0.981
Llama-3.3-70B	70B	0.992
DeepSeek-R1 (stripped)	671B	0.994
Overall		**0.981**

O CONTAMINATION-LEVEL ROBUSTNESS

AUROC is rank-based and invariant to base rate, so it is not the right lens for the contamination axis. We therefore fix one decision threshold, chosen by Youden's J on pooled out-of-fold predictions over the whole in-domain set (5-fold GroupKFold, grouped by seed id; threshold 0.348, pooled AUROC 0.9884), and read threshold-sensitive metrics off that single operating point within each contamination level. The table covers the five contaminated levels. The clean level is omitted because it contains no synthetic documents, which leaves AUROC, precision and F1 undefined there.

Each level is evaluated on the documents carrying that contamination label in the 9,462-document in-domain set, not on the 500-row split files of Appendix N: the evaluated n is 228 to 396 because a document enters the pooled out-of-fold frame only if all five signal families extracted for it. The realized synthetic fraction in each of those subsets, reported in the table, runs below the nominal split rate at every level, from 0.044 at the 5% label to 0.495 at 60%, so the precision trend should be read against the realized fraction, not the label. Confidence intervals are 1,000-sample bootstrap percentile intervals at a fixed seed; at 5% contamination the interval on precision spans 0.28 to 0.76, so the level-to-level movement in precision is much better resolved than any single low-contamination point.

Table 13: Threshold-sensitive detection within each contaminated level at one fixed operating point. AUROC is nearly flat (range 0.030) while precision rises from 0.53 to 0.98, which is the quantity relevant to corpus auditing. n is the number of in-domain documents carrying that label with all five signal families present, and *Synth. frac.* is the realized synthetic fraction of that subset.

Contamination	n	Synth. frac.	AUROC	95% CI	Precision	Recall	F1
5%	228	0.044	0.973	[0.911, 1.000]	0.529	0.900	0.667
10%	297	0.071	0.974	[0.935, 0.999]	0.679	0.905	0.776
20%	362	0.130	0.947	[0.906, 0.979]	0.822	0.787	0.804
40%	369	0.257	0.974	[0.957, 0.987]	0.814	0.874	0.843
60%	396	0.495	0.978	[0.963, 0.990]	0.983	0.883	0.930

P STRATIFIED IN-DOMAIN DETECTION

As a leakage-safe complement to the held-out-by-scale analysis in Section 5.4, we compute out-of-fold detection with 5-fold GroupKFold (grouped by seed id) and stratify the resulting predictions by contamination level and by source model on the 9,462-document in-domain set (pooled out-of-fold AUROC 0.9885).

Table 15: Out-of-fold AUROC stratified by source model. Detection is stable from 1B to 70B; Phi3-Mini is the weakest, and DeepSeek-R1 is reported on reasoning-trace-stripped text.

Source model	OOF AUROC
Llama-3.2-1B	0.997
Qwen2.5-1.5B	0.941
Gemma2-2B	0.989
Llama-3.2-3B	0.982
Qwen2.5-3B	0.972
Phi3-Mini	0.886
Llama-3.1-8B	0.991
Llama-3.1-8B (instant)	0.986
Llama-4-Maverick	0.973
Llama-3.3-70B	0.985
DeepSeek-R1 (stripped)	0.976

Every value above is the run that was shipped, and the run is not exactly repeatable. The script intersects the five feature caches into an unordered set and builds its table by iterating it, so the row order of the table depends on Python's per-process string hashing, and row order changes which seed groups land in which fold and which rows the XGBoost subsample draw takes. Over six runs the pooled out-of-fold AUROC moved between 0.9884 and 0.9887, so the shipped 0.9885 sits inside the spread and the run is consistent with the script, unlike the ablation file in Appendix L. The small cells move more: Phi3-Mini spans 0.011 across the same runs and the 5%-contamination cell in Appendix O, which rests on 10 synthetic documents, spans 0.018. Three digits is more precision than this table supports, and the reading it carries, that detection is stable from 1B to 70B with Phi3-Mini weakest, holds across every run. Pinning the hash seed makes the script repeatable to the last digit, and the released wrapper does that by default rather than editing a script whose digest is what makes it auditable.

Q Comparison with Prior Benchmarks

Table 16: WebMix-Collapse against the two nearest benchmarks: RAID (Dugan et al., 2024), the largest detection benchmark, and HART (Bao et al., 2025), the closest rewrite benchmark, whose cells are taken from those papers. They differ on the axis this paper is about: RAID leaves length free and reports the gap, HART equalizes it by prompt and truncation, and neither reports what a length-only detector scores.

Feature	RAID	HART	WebMix-Collapse
Rewrites of human documents	× (continuations)	✓ (1 pass)	✓ (5 depths)
Contamination levels	×	×	✓ (6 levels)
Generator scale sweep	× (11 models)	× (6 models)	✓ (1B to 671B)
Length policy	free, gap reported	equalized	controlled variable
Word-count-only baseline	×	×	✓
OOD evaluation	✓	✓	✓

R Dataset Card

Following Gebru et al. (2021), we document the following properties of WebMix-Collapse.

Motivation. WebMix-Collapse was created to support evaluation of synthetic text detectors under conditions that reflect the model-collapse auditing scenario: varying contamination rates, varying source model scale, and iterative rewriting. The creating authors are academic researchers; no external funding was received for this dataset.

Composition. 10,668 plain-text documents: 5,298 human (sourced from C4) and 5,370 synthetic (generated via the rewrite protocol in Section 3). Generation-1 contains 4,214 synthetic documents; depths 2 to 5 contain 296, 287, 287 and 286 documents. Metadata fields per document: document ID, generation depth, source model, contamination split assignment, word count where available, and human/synthetic label. We ran no personal-information audit of our own. The human documents inherit whatever C4's construction left in place, and C4 applied no PII removal step beyond its

deduplication, language and boilerplate filters, so documents naming real people or organizations do occur; the synthetic documents are rewrites of those, and can carry the same content forward. Users who need a PII-free corpus should filter before use.

Collection process. Human documents were sampled from C4 with no additional filtering beyond the length constraint described in Section 3. Synthetic documents were generated using free-tier APIs (Ollama, Groq, SambaNova) and, for the OOD commercial subsets, the OpenAI API; full API version information is recorded in the provenance metadata. Provenance is recorded per document; the out-of-domain subset used in this paper's detector sweep is produced by the three Ollama small models named in Appendix G, which are not held out from the in-domain training set.

Preprocessing. No preprocessing was applied beyond tokenization for feature extraction. Raw text is released as-is. The DeepSeek-R1 rows keep their reasoning traces so the published checksums still verify; a separate stripped view ships under `derived/r1/stripped/` (Appendix J).

Known defects. Four properties of the released files are documented rather than repaired, because the published SHA-256 digests have to keep verifying; the code release ships repaired views beside the originals under `derived/` and a validator that reports every affected row. (1) 679 pairs of byte-identical human documents inside the in-domain set, discussed in Section 5.1. (2) 1,599 rows with no `id` field, all in `ood_multigen_llama8b.jsonl` and `ood_multigen_qwen3b.jsonl`; these duplicate the synthetic halves of the corresponding `ood_eval_*` files, which do carry IDs, and the IDs are recoverable by exact text match. (3) 114 rows whose text contains cp1252 mojibake control characters, spread over 14 files. These are Windows-1252 bytes that were decoded with the wrong codec upstream of the release, in two patterns: 48 runs of double-encoded UTF-8 and 411 single bytes in the 0x80 to 0x9F range. The strict reader in the code release rejects them, which is why the released benchmark files cannot be read without the repair; `tools/repair_mojibake_text.py` reverses the mis-decode into copies under `derived/text_repair/`, lists each of its 459 substitutions and 13 removals, and refuses to write unless only the `text` field changed and the partition of documents by normalized text is unchanged, so the connected components and therefore the fold assignment are provably unaffected. (4) The six controlled splits are not disjoint, quantified in Appendix N.

Uses. Intended uses: evaluation of synthetic text detectors, study of model-collapse auditing pipelines, benchmarking of OOD generalization in text classification. The benchmark is not appropriate for training detectors without held-out OOD evaluation; in-domain results should not be reported without the word-count baseline.

Distribution. Our generated documents, provenance metadata, Croissant metadata and code are released under CC BY 4.0. For review they are in the anonymized supplementary archive; on publication they move to a permanent public location with a DOI. The C4 source documents keep C4's own license, ODC-BY 1.0, and Common Crawl's terms of use apply to their content.

Maintenance. The authors will maintain the repository and address reported issues. The OOD evaluation set is released separately and can be extended; the authors welcome contributions adding new domains and generators.

Intended audience. Researchers evaluating synthetic text detectors, practitioners building corpus auditing pipelines, and developers of model-collapse mitigation systems.

Ethical review. No human subjects are involved. C4 applied no personal-information filtering beyond its own deduplication, language and boilerplate filters (see Composition), and we added none. The benchmark could in principle be misused to improve evasion of detection systems, but the length-padding vulnerability we document is already trivially discoverable.

Update plan. Planned extensions include detector sweeps over the released commercial-generator subsets and alternative generation regimes.

S Re-implemented Weak Baselines

For the in-domain baseline table, DetectGPT and Binoculars were re-implemented using GPT-2 117M as the scorer (rather than the original GPT-2-XL and Falcon-7B, respectively). These in-domain rows are **not comparable** to published performance from those methods and are included only to show

relative ordering of detection approaches on this benchmark at a fixed compute budget. No claim in this paper depends on these rows.

Table 17: Weak in-domain baselines (GPT-2 117M scorer). Not comparable to published results.

Method	AUROC
DetectGPT (500 docs; GPT-2 117M)	0.660
Binoculars (GPT-2 117M pair, in-domain)	0.486

T Published Statistical Detectors on OOD

We evaluate three published statistical detectors, at several scorer scales each, and an instruction-tuned judge on the out-of-domain subset of Appendix G, whose domains are held out and whose generators are not. Our first Binoculars run had an error in its cross-perplexity computation and scored 0.378 on these documents, well below chance. Every Binoculars value below uses the metric as published, with the score negated so that machine text ranks high.

Implementing the Binoculars metric of Hans et al. (2024) as published and scoring all 1,478 OOD documents in float32 on CPU, the detector reaches AUROC 0.722 with a GPT-2 observer and a GPT-2-medium performer, 0.787 with a GPT-2-large observer and a GPT-2-XL performer, and 0.849 with the intended pair of a Falcon-7B observer and a Falcon-7B-Instruct performer. At identical model size the correct implementation gives 0.722 against 0.378, so an apparent collapse of this detector would have been an implementation artifact. Scale then contributes a further 0.127 AUROC across the ladder, so this metric does benefit from the model pair it was designed for, and reporting it only at GPT-2 scale understates it. On a 500-document seeded subsample of the same set the three pairs give 0.774, 0.827 and 0.852, in the same order.

A first run reported 0.834 for the GPT-2 pair on that subsample. That number came from a revision of our runner which was edited later the same day and not retained, and the packaged runner does not reproduce it: it gives 0.774 on the same documents, which two independent implementations agree on. We therefore report the reproducible values and ship both artifacts so the discrepancy is auditable.

All three rungs have since been re-run on all 1,478 documents with the per-document scores kept, which is what puts them in Table 19. The GPT-2 pair reproduced the printed 0.722 to seven decimal places in float32 on CPU. The GPT-2-large and Falcon-7B pairs were re-run in float16 on an accelerator and landed within 5×10^{-5} of the values above, 0.7871552 against 0.787179 and 0.8486394 against 0.8485988, so the ladder no longer rests on any run whose scores were discarded. Appendix U gives the runner, the seeds and the input digest.

DetectGPT (GPT-2 scorer, T5-small perturber) was first run at 25 perturbations and reached AUROC 0.653 on a 500-document class-balanced subsample. It has since been run twice with the per-document scores kept, on the same 500 documents: 0.6429 at 25 perturbations and 0.6483 at the 100 that Mitchell et al. (2023) prescribe. Quadrupling the budget is worth 0.005 AUROC here, for 10,917 seconds of accelerator time against 3,898. DetectGPT is therefore weak on this benchmark at its intended budget, not only in a cheap configuration. The remaining gap between the shipped 0.653 and the 0.6429 rerun comes from perturbation sampling, not from a configuration change: DetectGPT draws its rewrites, the shipped run fixed no torch seed, and each rerun seeds every document from its own id so that a resumed or sharded run equals an uninterrupted one.

An instruction-tuned LLM judge (Qwen2.5-32B-Instruct, zero-shot, greedy decoding) was first run at AUROC 0.797 with a judge-score-to-word-count correlation of -0.08. That run kept neither the per-document scores nor the prompt, so the package cannot recompute it, and it is one of the three numbers the reproducibility statement flags. A second judge run does ship, on the same 500 documents as the DetectGPT reruns and with the same model: one yes-or-no question per document, the score read as the probability mass the model puts on the Yes tokens at the first answer position, so nothing is sampled and nothing is parsed out of free text. It reaches 0.8165 with a length correlation of -0.032 by Pearson and -0.023 by rank; the main text and Table 19 quote the rank value. The prompt, its sha256, the token id sets for Yes and No, and one score per document are all released. We report it as an independent measurement, not a reproduction of 0.797, and Table 19 uses it because it is the judge run whose scores exist.

The DetectGPT and judge runs share one class-balanced draw of 500 documents, whereas the Binoculars subsample used a seeded permutation that preserves the class proportions of the full set (227 synthetic of 500), so the subsample rows below are not scored on the same documents as the 500-document rows of Table 19.

Table 18: Published and contextual detectors on the OOD generator subset, including the configurations that Table 19 leaves out. Every full-set value has been reproduced with the per-document scores retained (Appendix U). The two rows marked as earlier runs kept only an AUROC; both are superseded by a released rerun on the same detector.

Detector	**Configuration**	n	**AUROC**
Binoculars	Falcon-7B / Falcon-7B-Instruct (intended scale)	1,478	0.849
Binoculars	GPT-2-large / GPT-2-XL	1,478	0.787
Binoculars	GPT-2 / GPT-2-medium (matched compute)	1,478	0.722
Binoculars	Falcon-7B / Falcon-7B-Instruct, subsample	500	0.852
Binoculars	GPT-2-large / GPT-2-XL, subsample	500	0.827
Binoculars	GPT-2 / GPT-2-medium, subsample	500	0.774
Fast-DetectGPT	Falcon-7B / Falcon-7B-Instruct	1,478	0.843
Fast-DetectGPT	GPT-2, sampling and scoring (white box)	1,478	0.768
DetectGPT	GPT-2 + T5-small, 100 perturbations	500	0.648
DetectGPT	GPT-2 + T5-small, 25 perturbations	500	0.643
DetectGPT	GPT-2 + T5-small, 25 perturbations, earlier run	500	0.653
LLM-as-a-judge	Qwen2.5-32B-Instruct, zero-shot	500	0.817
LLM-as-a-judge	Qwen2.5-32B-Instruct, earlier run	500	0.797

Takeaway. A correctly implemented cross-perplexity detector does not collapse on this OOD set, and neither does an LLM judge. Neither does a correctly implemented feature ensemble: the only near-chance out-of-domain number in this paper belongs to the shipped predictions, and Appendix V traces it to a unit convention, not to length dependence. What these rows do show is that out-of-domain AUROC on this set spans 0.566 to 0.972 without ordering the detectors by how much length they read, which is the point Section 5.3 draws from them.

U Reruns Behind the Cross-Detector CAP Table

Table 19 needs a score per document, and several of the detectors in it had only their AUROCs kept, so those runs were repeated from scratch with the per-document values written to disk as they were computed. The runner is a copy of the shipped `binoculars_scale_ladder.py` scoring function with the loop rewritten to append one JSON line per document; it is released as `binoculars_per_document.py` together with the score files.

Table 20: The three Binoculars reruns. Each scored all 1,478 OOD documents of the same input file, sha256 `450197d9...82939`, with a 512-token truncation limit. The rerun column is the AUROC of the negated ratio, as published; the first-run column is the AUROC from the original run, whose per-document scores were not retained.

Observer / performer	**Device, precision**	**Rerun**	**First run**
GPT-2 / GPT-2-medium	CPU, float32	0.7219487	0.7219487
GPT-2-large / GPT-2-XL	accelerator, float16	0.7871552	0.7871790
Falcon-7B / Falcon-7B-Instruct	accelerator, float16	0.8486394	0.8485988

The CPU run is bit-reproducible and matched to seven decimals. The two accelerator runs agree with the first-run numbers to within 5×10^{-5}, which is the size of a float16 reduction-order difference on 1,478 documents and is four orders of magnitude below the effects the paper reads off these values. Binoculars has no sampling step, so no seed enters; the only stochastic element in the table is the caliper matching, which draws from the documents rather than from the detector and is therefore shared by every row (`cap_cross_detector.py` recomputes each detector's headline AUROC from the score file and refuses to report a row that disagrees with its summary by more than 10^{-9}).

Two limits apply. The reruns settle whether the ladder is reproducible, not whether Binoculars is the best available detector: they use the metric as published, with no threshold tuning and no prompt

Table 19: Every detector we scored, with CAP Steps 1 to 4 wherever per-document scores were retained; no retraining is involved. Step 1 is word count alone on the same rows, oriented so that shorter means synthetic. Step 2 removes the score's least-squares fit on log word count, on the score and on its ranks; read the rank column, since only it is invariant to monotone rescaling. Step 3 is a caliper-matched subset ($n = 2{,}908$ in domain, 318 to 1,034 out of domain). Step 4 is the four-domain ratio-to-AUROC correlation, descriptive only, printed to show that it does not order the detectors as the other steps do; ρ is Spearman between score and word count. Rows marked *as trained* use the feature code that wrote the training caches; the last row is the release's own shipped predictions, printed to document the discrepancy rather than as a measurement (Appendix V). Row sets differ, so Appendix W repeats the comparison on one shared set. DeBERTa kept only its AUROC. The in-domain ensemble reads 0.987 here and throughout the paper, which is the release's component-grouped out-of-fold value rather than the 0.9885 the ID-grouped protocol gives (Appendix K).

Detector	**Set.**	n	**AUROC**	**St. 1**	**St. 2**	**St. 2$_{\text{rank}}$**	**St. 3**	**St. 4 / ρ**
Word count alone (positive control)	in-dom.	9,462	0.942	0.942	0.737	0.729	0.599	– / −1.00
SynDet-Lite, 19 features	in-dom.	9,462	0.987	0.942	0.842	0.788	0.910	– / −0.91
SynDet-Lite, residualized features	in-dom.	9,462	0.898	0.942	0.727	0.718	0.854	– / −0.60
DeBERTa-base, fine-tuned	OOD	1,478	0.972	–	–	–	–	– / –
RoBERTa-base, fine-tuned	OOD	1,478	0.947	0.548	0.893	0.946	0.957	+0.67 / −0.04
SynDet-Lite, 19 feat., as trained	OOD	1,074	0.855	0.577	0.551	0.844	0.828	−0.21 / −0.37
Binoculars, Falcon-7B pair	OOD	1,478	0.849	0.548	0.856	0.853	0.847	+0.39 / +0.11
Fast-DetectGPT, Falcon-7B pair	OOD	1,478	0.843	0.548	0.861	0.856	0.845	+0.32 / +0.19
LLM judge, Qwen2.5-32B	OOD	500	0.817	0.535	0.745	0.814	0.794	+0.29 / −0.02
Binoculars, GPT-2-large / GPT-2-XL	OOD	1,478	0.787	0.548	0.795	0.792	0.779	+0.46 / +0.10
SynDet-Lite, 15 feat., as trained	OOD	1,174	0.776	0.549	0.561	0.768	0.732	+0.09 / −0.26
Fast-DetectGPT, GPT-2 white-box	OOD	1,478	0.768	0.548	0.818	0.811	0.777	−0.21 / +0.38
Binoculars, GPT-2 / GPT-2-medium	OOD	1,478	0.722	0.548	0.726	0.723	0.731	+0.69 / +0.03
DetectGPT, 100 perturbations	OOD	500	0.648	0.535	0.704	0.695	0.701	−0.58 / +0.50
DetectGPT, 25 perturbations	OOD	500	0.643	0.535	0.698	0.690	0.693	−0.40 / +0.51
Shipped release predictions, second feature implementation, not a measurement:								
SynDet-Lite, 19 features	OOD	1,428	0.566	0.578	0.513	0.521	0.524	+0.96 / −0.42

engineering. And a float16 accelerator run is not the same object as a float32 CPU run, so the ladder is internally consistent at the precision each rung was computed at rather than at one precision throughout.

Fast-DetectGPT. Fast-DetectGPT's (Bao et al., 2024) conditional probability curvature costs one forward pass per model per document rather than a hundred perturbed rewrites, so it was run for the cross-detector table as a fourth detector family. The statistic is the analytic form, transcribed from `get_sampling_discrepancy_analytic` in the authors' released code: it needs no samples, so no seed enters and the score is a deterministic function of the models and the text. With GPT-2 as both the sampling and the scoring model it reaches AUROC 0.7678 on all 1,478 OOD documents in float32 on a CPU, in 116 seconds. Substituting the Falcon-7B and Falcon-7B-Instruct pair, the same two models Binoculars uses at its intended scale, lifts it to 0.8427, within 0.006 of Binoculars on the same documents, and that run is also float32 on a CPU, in 5,179 seconds. Both Fast-DetectGPT rungs therefore need no accelerator and no seed. Repeated on a second machine, the GPT-2 rung gave the same AUROC to every digit, with no per-document score moving by more than 2×10^{-5} (Appendix U.1). The runner documents two differences from the reference script, neither touching the statistic: it reads one document at a time from the benchmark file rather than paired passages, and it truncates at 512 tokens, the cap the Binoculars runner already used, so the zero-shot detectors all see the same prefix of each document.

DetectGPT at its prescribed budget. The 100-perturbation run is the most expensive row in the table, at 10,917 seconds against 3,898 for 25 perturbations, and it was sharded across four accelerators. Sharding is safe here only because each document's perturbations are seeded from `sha256(seed:id)`, so a document's score does not depend on which shard scored it or on how far into the loop it sat; `combine_shards.py` then checks that the shards agree on every configuration field, that their id sets are disjoint, and that together they cover the subset exactly before it writes a

merged file. The result is that a sharded or resumed run is the same object as an uninterrupted one, which matters for a detector whose score is an average over sampled rewrites.

The judge. The score is the probability the model puts on answering Yes to a single yes-or-no question about the passage, summed over the five token ids that spell an affirmative answer at the first answer position and normalized against the five that spell a negative one. Reading a probability rather than generating text removes two failure modes that make judge baselines hard to compare: there is no sampling, so the run is deterministic given the weights, and there is no parsing, so a verdict cannot be lost to an unexpected preamble. The run used float16 on one accelerator and took 172 seconds for 500 documents truncated at 1,024 tokens. Its scores are the reason Table 19 can put a judge through all three CAP steps, and they are also the case that exposed the Step 2 representation problem discussed in Appendix C, since 61.4% of them fall below 0.01.

U.1 CAP Step 5 for two zero-shot detectors

Step 5 is the one CAP step whose prediction can fail, so it should also be run on detectors we did not build. We scored all 787 documents of the reversed-length split with two zero-shot detectors, using the package's runners on the CPU in float32: Fast-DetectGPT with GPT-2 as sampling and scoring model, and Binoculars with a GPT-2 observer and a GPT-2-medium performer. The runners read the split as one file, `derived/original_generation/original_gen_combined.jsonl`, which concatenates the two released files. Before these runs, the recorded Fast-DetectGPT GPT-2 run on the 1,478 out-of-domain documents was repeated on the machine used for them. The repeat gave the recorded AUROC, 0.7677784, to every digit, and no per-document score moved by more than 2×10^{-5} (the script below checks this and writes it with its other results). Table 21 gives the results, written by `experiments/scripts/inversion_zero_shot.py`, with intervals from 2,000 class-stratified resamples at seed 42. The feature-ensemble rows are copied from the recomputation in Section 5.6.

Table 21: CAP Step 5 on the reversed-length split. Word count is the AUROC of word count alone on the same rows, shorter meaning synthetic. Step 3 is the length-matched AUROC with the caliper of Table 1, with the number of matched documents. ρ is Spearman between score and word count. The last column restricts to the four categories both classes contain.

Detector	n	AUROC [95% CI]	Word count	Step 3 (n)	ρ	Four cat.
SynDet-Lite, 19 features	680	0.289 [0.246, 0.333]	0.091	0.853 (180)	-0.81	0.293
SynDet-Lite, 15 features	722	0.231 [0.196, 0.270]	0.079	0.811 (180)	-0.85	0.240
Fast-DetectGPT, GPT-2	787	0.969 [0.955, 0.981]	0.068	0.963 (182)	$+0.66$	0.990
Binoculars, GPT-2 / GPT-2-medium	787	0.977 [0.968, 0.985]	0.068	0.988 (182)	$+0.58$	0.983

Neither zero-shot detector inverts. Both separate the classes on the split where the feature ensemble ranks them backwards, and length matching leaves each close to where it started, while it moves the ensemble from 0.289 to 0.853. On the 680 documents the ensemble scores, Fast-DetectGPT is $+0.673$ [$+0.627$, $+0.719$] above it and Binoculars $+0.691$ [$+0.648$, $+0.734$], from a paired bootstrap that resamples documents within class, 2,000 times at seed 42. The reversal is therefore not a property of the split. Both zero-shot scores do rise with word count here (ρ above), since the synthetic class is also the longer one, but on the length-matched documents, where word count alone scores 0.527, they reach 0.963 and 0.988, so length is not what separates the classes for them.

V Four Implementations of the Same Nineteen Features

The release computes the nineteen features in four places, and the four do not agree. `signal1_perplexity.py` through `signal5_similarity.py` wrote the five cache files that every in-domain result is trained and evaluated on. `eval_ood_v2.py` recomputes all nineteen inside itself and produced the shipped out-of-domain predictions. `extract_ood_features_v2.py` is a third extractor whose output is not in the release, so what it computed cannot be checked. `eval_original_gen.py` is a fourth, and it produced the 0.204 figure on the inversion split. All four ship under `historical/source/` with a `.txt` suffix, as reference copies a reader can read but not run against a layout they predate. The trained booster

is not released either, so every ensemble number in this paper comes from retraining on the caches: five-fold mean 0.9871, per fold 0.9856, 0.9842, 0.9860, 0.9879 and 0.9918, against the published 0.98688.

What the port reproduces. `syndet_features.py` is a single extractor written against the training-time scripts, and `verify_syndet_features.py` checks it feature by feature against the caches on 200 documents drawn at seed 42 from the 9,462 present in all five. Fifteen of the nineteen reproduce: the nine computed from the text alone match the caches exactly, apart from floating-point reassociation at the 10^{-16} level in mean burstiness, and the six that run a language model or a sentence embedder match to within a float32 tolerance of 10^{-3}, taken as absolute or relative, with Spearman 1.00 against the cache in every case. The four spaCy features fail: `syn_depth_mean` by up to 3.00 in absolute terms (Spearman 0.944), `syn_depth_var` by 41.94 (0.818), `syn_dep_dist_mean` by 1.98 (0.881) and `syn_sent_len_var` by 1111.83 (0.963). The cause is not recoverable from what shipped: `signal4_syntactic.py` loads `en_core_web_sm` with no version, and `requirements.txt` pins numpy, scipy, scikit-learn, xgboost-cpu, joblib, cloudpickle, threadpoolctl and pytest, and none of torch, transformers, spaCy or sentence-transformers, which are the four packages the five extractors need. That is why every out-of-domain figure the port produces is reported twice, once on all nineteen features and once on the fifteen that verify. The in-domain figures need no port, since they are trained and evaluated on the released caches themselves, so they are quoted on all nineteen features; their fifteen-feature counterpart, 0.9851 out of fold, appears only as the reference point of Appendix AA.

One unit convention accounts for the collapse. `signal1_perplexity.py` records the per-token cross-entropy in nats, straight from `CrossEntropyLoss`. `eval_ood_v2.py` exponentiates it. The four features carry the same names in both files and are on different scales. Feeding the exponentiated version to a booster trained on the nats version, with the other fifteen features untouched, takes out-of-domain AUROC from 0.855 to 0.615. On the fifteen verified features the same substitution gives 0.776 against 0.591, and on the inversion split 0.289 against 0.243. The near-chance out-of-domain AUROC of the shipped predictions is therefore what this substitution produces, and not a property of the detector.

The near-constant high confidence of the shipped predictions, by contrast, is not an artifact of the substitution. Saturation is a property of this ensemble under either implementation, and the substitution slightly reduces it. On the correct path 86.0% of the out-of-domain probabilities exceed 0.99, with a median of 0.9987 on human documents against 0.9999 on synthetic ones; on the exponentiated path 70.9% do, at 0.9931 against 0.9959. The fifteen verified features give 88.5% against 84.8%, and the inversion split 20.7% against 13.7%. All three comparisons run the same way, so the calibration failure reported in Section 5.5 stands on the corrected feature path, and it is only the ranking collapse that does not.

Like for like. On the 1,073 documents both the port and the shipped run scored, the ported features reach 0.855 and the shipped features 0.610, with word count alone at 0.577 on those same rows; on the fifteen-feature set, over 1,148 shared documents, 0.775 against 0.589. The row sets differ because the port declines a document whenever any family declines it, and the training-time signal 3 declines any document shorter than thirty words or with fewer than three topic words, a topic word being one that occurs two to ten times and is longer than three characters: 355 documents that the shipped run scored are declined by the nineteen-feature port and 280 by the fifteen-feature port. Restricting to the shared rows removes that difference and leaves the implementation as the only thing that varies.

The inversion script. `eval_original_gen.py` differs from the published in-domain path in three ways. It fits a `StandardScaler` on the training features and applies it out of domain, which the published path does not do. It replaces a missing or NaN feature with zero instead of declining the document, so a document whose parser or embedder failed still enters the fit with zeros in several columns. And its word-count baseline fits a logistic regression on a column of zeros whenever `word_count` is absent from the training frame, so the length baseline it prints is not a length baseline. Its 0.204 compares two implementations and does not measure the detector, which is why Section 5.6 reports the recomputation, 0.289 [0.246, 0.333], instead.

Three historical summaries that cannot be adjudicated. Three artifacts in the release report the nineteen-feature ensemble out of domain on 1,478 documents: a de-biasing summary at 0.794, a length-matching summary at 0.770, and a prefix-truncation summary at 0.802 with an in-domain

value of 0.798 at a 100-word prefix against 0.987 on full text. The shipped per-document predictions give 0.566 on 1,428 documents, and the recomputation gives 0.855 on the 1,074 the training-time extractors score. A fourth file, `bootstrap_ci_results.json`, reports 0.600 [0.571, 0.629] on a row set of its own. The provenance file marks the three summaries as historical and not revalidated, and the scripts that wrote them are not in the release, so the disagreement cannot be attributed to a feature path, a row set or a training subsample. What it does establish is that no out-of-domain figure for this ensemble can be quoted without naming the artifact it came from. This paper quotes the recomputation, whose extractor is checked against the caches feature by feature, keeps the shipped predictions beside it in every table, and leaves the three summaries in the release without building on them.

W Every Out-of-Domain Detector on One Shared Row Set

Detectors in Table 19 were scored on different rows, so a reader comparing two rows is reading a difference of method and a difference of documents at once. This appendix removes the second one. The reference set is the 1,074 documents where all five ported feature families scored, which is the row set the nineteen-feature recomputation is measured on; every other detector is restricted to it. The three runs that used the 500-document subsample cannot cover the reference set, so their intersection with it is 355 documents and is marked as such. Written by `experiments/scripts/ood_common_rows.py`.

Table 22: Each out-of-domain detector twice: on its own rows, and on the rows it shares with the reference set of 1,074 documents. Step 3 is the caliper-matched subset of the shared rows: the same 802 documents for every detector that covers the reference set, 800 for the shipped predictions and 236 for the three runs restricted to 355 rows. ρ is Spearman between the detector's score and word count on the shared rows. Intervals are 2,000 resamples of the shared rows stratified by class, seed 42.

Detector	Own n	Own	Shared	95% CI	Step 3	ρ	Covers
RoBERTa-base, fine-tuned	1,478	0.947	0.970	[0.961, 0.979]	0.967	$+0.05$	yes
SynDet-Lite, 19 feat., as trained	1,074	0.855	0.855	[0.830, 0.879]	0.828	-0.37	yes
Binoculars, Falcon-7B pair	1,478	0.849	0.852	[0.829, 0.874]	0.856	-0.01	yes
Fast-DetectGPT, Falcon-7B pair	1,478	0.843	0.848	[0.824, 0.870]	0.855	$+0.08$	yes
LLM judge, Qwen2.5-32B	500	0.817	0.817	[0.772, 0.859]	0.836	-0.23	no, 355
Binoculars, GPT-2-large / GPT-2-XL	1,478	0.787	0.780	[0.752, 0.807]	0.785	$+0.02$	yes
SynDet-Lite, 15 feat., as trained	1,174	0.776	0.779	[0.750, 0.807]	0.737	-0.32	yes
Fast-DetectGPT, GPT-2 white-box	1,478	0.768	0.752	[0.723, 0.779]	0.773	$+0.32$	yes
Binoculars, GPT-2 / GPT-2-medium	1,478	0.722	0.724	[0.694, 0.754]	0.736	-0.04	yes
DetectGPT, 100 perturbations	500	0.648	0.669	[0.615, 0.723]	0.703	$+0.58$	no, 355
DetectGPT, 25 perturbations	500	0.643	0.663	[0.607, 0.717]	0.697	$+0.57$	no, 355
SynDet-Lite, 19 feat., shipped	1,428	0.566	0.610	[0.577, 0.643]	0.542	-0.49	no, 1,073
Word count alone (floor)	1,074	0.577	0.577	[0.544, 0.610]	0.525	-1.00	yes

The restriction moves nothing far. Six of the eleven detectors it can be applied to move by less than 0.01, and the largest movements are the shipped predictions at $+0.044$, RoBERTa at $+0.023$, and the two DetectGPT rungs at $+0.021$ and $+0.020$. The reference rows are longer on average than the full 1,478, and the three configurations whose scores track word count most strongly are the shipped predictions at $\rho = -0.49$ and the two DetectGPT rungs at $+0.58$ and $+0.57$, all three of which gain. Length alone does not account for the pattern, though: the gains have the same sign either side of $\rho = 0$, and RoBERTa gains as much as the DetectGPT rungs at $\rho = +0.05$. One ordering changes. On their own rows the shipped predictions sit below the word-count floor, 0.566 against 0.577; on the shared rows they sit above it, 0.610 against 0.577. Every other pair of detectors keeps its order.

Five comparisons are paired, with one resample scoring both detectors on the documents both scored, which removes the between-document variance that dominates an unpaired comparison. Against Binoculars at Falcon scale the nineteen-feature ensemble is $+0.003$ with an interval of $[-0.029, +0.035]$ on 1,074 documents, so on these rows the two are not separated. Against RoBERTa it is -0.115 [-0.141, -0.089], so the gap to a fine-tuned transformer is real and is about a tenth of an AUROC. Against its own fifteen-feature restriction it is $+0.076$ [$+0.061$, $+0.091$], which is the price of dropping the four features the port cannot reproduce and is also the reason the paper

reports both. Against the shipped predictions of the same nineteen features it is $+0.244$ [$+0.215$, $+0.274$] on the 1,073 documents both scored, which is the implementation difference of Appendix V measured as a paired quantity. Against word count alone it is $+0.278$ [$+0.243$, $+0.313$]. The first and the last of those five are the two that matter for the paper's claims: out of domain the feature ensemble is level with the strongest zero-shot detector and is a long way above the length control, and it is well below a supervised transformer.

X Whether the Protocol Depends on Its Own Choices

CAP makes four choices a reader is entitled to call arbitrary. This appendix measures each one instead of defending it. Written by `experiments/scripts/protocol_robustness.py`, deterministic apart from the seeded permutations in the second check.

The caliper. Step 3 accepts a match when the word-count gap is at most $\max(5 \text{ words}, 5\%)$. Sweeping the tolerance from exact equality to 20% moves the matched out-of-domain AUROC of the nineteen-feature ensemble over 0.8411 (566 rows), 0.8278 (682), 0.8279 (802), 0.8319 (902) and 0.8350 (924), a spread of 0.013 across the five settings and 0.006 between exact equality and a 20% tolerance. The other out-of-domain detectors behave the same way: RoBERTa spans 0.9487 to 0.9570, Binoculars at Falcon scale 0.8380 to 0.8628, the fifteen-feature set 0.7323 to 0.7517. In domain the sweep is the informative one. The ensemble runs 0.8914, 0.8923, 0.9096, 0.9258, 0.9483 as the caliper loosens, and word count alone runs 0.4990, 0.5164, 0.5985, 0.6752, 0.7585. At exact word count the positive control sits at chance, which is what a length-only detector has to do when every pair is length-identical, and the ensemble keeps 0.891 of its 0.987. The looser the caliper, the more length leaks back in, and it leaks back into the control four and a half times faster than into the ensemble (0.2595 against 0.0569 over the sweep). Nothing in the paper's reading depends on the setting; the published one is the middle of five.

The matching order. Greedy matching walks synthetic rows in ascending word count and takes the nearest unused human row. Walking them in descending order or in five seeded random orders changes which rows survive. For the nineteen-feature ensemble out of domain the seven orders span 0.00754 of AUROC (ascending 0.8279 on 802 rows, descending 0.8270 on 814, the five random draws between 0.8204 and 0.8277). In domain the ensemble spans 0.0195 and word count alone spans 0.1117, from 0.5985 at ascending to 0.4869 at descending. The order therefore matters most where the confound is largest, which is again the control rather than the detector.

A replacement for the residualization step. Step 2 subtracts a least-squares fit, which is not rank preserving, so what it reports depends on how the score happens to be written down; the paper treats that as a defect of the step. The replacement has no such dependence: bin all rows into ten word-count deciles, and over every human/synthetic pair inside a decile count how often the detector ranks the synthetic document higher, pooling over deciles by the number of comparable pairs. It is an AUROC restricted to length-comparable pairs, so it depends only on ranks, and it uses every row, not just the subset a caliper admits. Out of domain the nineteen-feature ensemble reads 0.8699 over 24,045 comparable pairs, $+0.015$ above its unrestricted 0.855, and its ten deciles run from 0.801 to 0.922 with no trend in word count. RoBERTa reads 0.9400 (-0.007), Binoculars at Falcon scale 0.8563 ($+0.008$), the fifteen-feature set 0.7758 (-0.000), and the shipped predictions 0.5171 (-0.049). The two DetectGPT rungs gain the most, $+0.096$ and $+0.087$, consistent with their being the two detectors whose scores track length most strongly. In domain the same reading gives 0.9145 for the ensemble against 0.987 unrestricted, and 0.5435 for word count alone against 0.942, which is the same separation Step 3 reports without the caliper's row loss. One caveat belongs with the in-domain figure: because human documents there are 2.18 times longer than synthetic ones, four of the ten deciles hold only one class, so the 773,031 comparable pairs come from the six deciles that overlap, spanning 199 to 608 words. The in-domain stratified reading is a statement about that overlap, not about the whole corpus.

The 500-document subsample. Three runs were too expensive to score on all 1,478 out-of-domain documents and used a class-balanced 500-document subsample. Every detector that did score all 1,478 can be rescored on those same 500 rows, which puts a number on whether the subsample is representative: Binoculars at Falcon scale moves -0.013, Binoculars at GPT-2 scale $+0.009$, Binoculars at GPT-2-large scale $+0.001$, Fast-DetectGPT at Falcon scale -0.008, Fast-DetectGPT at GPT-2 scale $+0.001$ and RoBERTa -0.012. The largest movement is smaller than the width of

any 500-document interval in Table 5, so the subsample can be compared with the full set at the resolution this paper reads.

Y Operating Points

AUROC is a ranking summary and a corpus auditor does not deploy a ranking. Table 23 gives the true-positive rate each configuration reaches at 1%, 5% and 10% false positives. One caveat governs every number in it: the threshold for each target rate is chosen on the same rows it is then scored on, so each value is an upper bound on what the same detector would reach at a threshold chosen on other data. Written by `experiments/scripts/operating_points.py`.

Table 23: True-positive rate at 1%, 5% and 10% false positives, for every configuration with retained per-document scores, on the full row set and on the CAP Step 3 matched subset. Thresholds are chosen on the rows they are scored on, so every entry is an upper bound.

			Full set			**Step 3 matched**		
Detector	n	**AUROC**	**1%**	**5%**	**10%**	**1%**	**5%**	**10%**
In domain, out of fold								
SynDet-Lite, 19 features	9,462	0.987	0.820	0.929	0.965	0.400	0.624	0.740
Word count alone (control)	9,462	0.942	0.696	0.759	0.829	0.187	0.242	0.290
SynDet-Lite, residualized	9,462	0.898	0.349	0.581	0.708	0.263	0.467	0.605
Out of domain								
RoBERTa-base, fine-tuned	1,478	0.947	0.347	0.814	0.855	0.298	0.851	0.894
SynDet-Lite, 19 feat., as trained	1,074	0.855	0.312	0.558	0.660	0.289	0.499	0.618
Binoculars, Falcon-7B pair	1,478	0.849	0.298	0.491	0.584	0.298	0.470	0.582
Fast-DetectGPT, Falcon-7B pair	1,478	0.843	0.327	0.462	0.550	0.348	0.466	0.563
LLM judge, Qwen2.5-32B	500	0.817	0.016	0.284	0.456	0.025	0.258	0.428
Binoculars, GPT-2-large / GPT-2-XL	1,478	0.787	0.087	0.304	0.460	0.075	0.242	0.478
SynDet-Lite, 15 feat., as trained	1,174	0.776	0.060	0.270	0.423	0.063	0.223	0.347
Fast-DetectGPT, GPT-2 white-box	1,478	0.768	0.074	0.246	0.388	0.130	0.333	0.424
Binoculars, GPT-2 / GPT-2-medium	1,478	0.722	0.034	0.153	0.302	0.037	0.182	0.340
DetectGPT, 100 perturbations	500	0.648	0.032	0.104	0.236	0.044	0.201	0.308
DetectGPT, 25 perturbations	500	0.643	0.036	0.108	0.160	0.050	0.176	0.283
SynDet-Lite, 19 feat., shipped	1,428	0.566	0.003	0.037	0.079	0.006	0.045	0.097

Three points follow. First, the in-domain operating points are the ones a length confound flatters most: the ensemble catches 82.0% of rewrites at 1% false positives in domain, and word count alone catches 69.6%, so at the operating point a practitioner would actually pick, most of the ensemble's recall is available from a word counter. On length-matched rows the two separate cleanly, 40.0% against 18.7%. Second, out of domain the ordering by recall is not the ordering by AUROC: Fast-DetectGPT at Falcon scale sits below Binoculars at the same scale by AUROC, 0.843 against 0.849, and above it at 1% false positives, 0.327 against 0.298, then below it again at 5% and 10%. The judge is the sharper case: at AUROC 0.817 it catches 1.6% of synthetic documents at 1% false positives, because its scores pile up near zero and the top of its ranking is thin. AUROC alone would not have shown either. Third, the transfer of a threshold is worse than the transfer of a ranking. The probability threshold that gives the ensemble 1% false positives in domain is 0.9115, and carrying that same threshold to the out-of-domain rows gives a false-positive rate of 85.0% at 97.0% recall; at the 5% and 10% in-domain settings the same carry-over gives 86.8% and 87.7%. A detector calibrated on rewrites of one corpus and applied to another corpus flags almost every human document it sees.

Z De-biasing Analysis

Three de-biasing strategies were tested on the full corpus by retraining, without the cross-validation splits. Two qualifications belong before the table. The script that produced it is not in the release and its results file is marked historical and not revalidated, so the feature path it used cannot be identified; and its own nineteen-feature baseline, 0.794 out of domain on 1,478 documents, agrees with neither the shipped out-of-domain predictions, at 0.566 on 1,428, nor the recomputation, at 0.855 on 1,074 (Appendix V). The rows below are therefore comparable with each other and with no other table

in this paper, and none of the paper's claims rests on them. They are kept because the ordering they report is the one the recomputed results also produce: removing length from the features costs in-domain accuracy without buying out-of-domain transfer.

Table 24: De-biasing SynDet-Lite. The four strategy rows come from one historical summary, `debiased_syndet_results.json`, whose script is not in the release, and are to be read against each other only; the 8-feature row is signal family S1's four perplexity features together with S4's four syntactic features. No strategy recovers RoBERTa-level OOD performance. The two transformer rows are not in that summary and are printed only as a reference scale: RoBERTa from `roberta_metrics.json` in domain and `bootstrap_ci_results.json` out of domain, DeBERTa from `deberta_results.json`.

Strategy	In-Dom.	OOD
Original (19 feat.)	0.989	0.794
Length-normalised (÷ wc)	0.988	0.629
S1+S4 only (8 feat.)	0.894	0.720
Length-residualised	0.964	0.639
RoBERTa (reference)	0.997	0.970
DeBERTa (reference)	0.997	0.972

A second historical summary, with the same status and the same missing script, matched 804 of the 1,478 out-of-domain documents on length (human mean 102 words against synthetic 103) and reports AUROC 0.771 against 0.770 on the unmatched 1,478. Read on its own terms that says length matching costs this ensemble nothing out of domain, and the recomputation agrees in direction while disagreeing in level: CAP Step 3 moves the nineteen-feature ensemble from 0.855 to 0.828, which its own interval covers (Table 5). Both readings point the same way. Out of domain the length gap is not what limits this ensemble; content is. In domain the reverse holds, and word count alone reaches 0.942.

AA Generation Depth Analysis

Table 25: Document counts and average length by generation depth. Average length declines monotonically with depth. Only depths 0 and 1 are in the in-domain evaluation file, so depths 2 to 5 carry no released detector result; Table 26 supplies one.

Depth	n (synth)	Avg. words
1	4,214	193
2	296	142
3	287	133
4	287	128
5	286	120

The monotonic decline in average word count across depths (193, 142, 133, 128, 120) is consistent with the length confound: deeper rewrites produce shorter documents, which should make length-based discrimination progressively easier. That is a prediction, not an observation, so we measured it, and doing so first required noticing that the depth axis had never been evaluated at all: `webmix_collapse_v5_train.jsonl`, the file every in-domain number in this paper and in the release is computed on, holds only generations 0 and 1. The 1,156 documents at depths 2 to 5 are in the master file and in no detector evaluation.

What we measured, and the asymmetry it carries. The positives at depth d are the depth-d synthetic documents, featurized with `syndet_features.py` and scored by a model fit on all 9,462 in-domain rows from the released caches, which is the train-on-caches, test-on-port protocol of Appendix V and the same reason to read the fifteen verified features instead of all nineteen. The negatives are the 5,295 human rows at their published out-of-fold scores, at a mean of 422 words. No row on either side is scored by a model that saw it, but the two sides are scored by different fits of the same estimator, because the depth documents have no human counterpart of their own; and a depth-5 document is an iterated rewrite of a seed whose human original is in the training set, so seed

content is shared by construction. The four depths are also narrower than their row counts suggest. All 1,156 documents descend from the same 100 seed documents, the depth-5 seed set being a subset of the depth-2 one, and each depth was produced by a single rewriter: `gemma2:2b` at depths 2 and 5, `llama3.2:3b` at depth 3, `qwen2.5:3b` at depth 4. Depth is therefore confounded with rewriter identity, no depth contributes an independent sample of seeds, and a difference between two depths is a difference between two rewriters applied at two depths. The trend below is one instance of iterated rewriting, not an estimate of a general depth effect, which is also why we read the margins against the length control and not the absolute values. The numbers below are therefore a comparison across depths, where both of those hold constant, and not a clean held-out measurement. The port declines 11, 22, 20 and 24 documents at the four depths, all of them under its thirty-word or three-sentence floors. Word count alone is fit and scored the same way, as CAP Step 1.

The asymmetry can be bounded rather than only declared, by running the same arrangement where the answer is already known. Holding out the components of a tenth of the depth-1 synthetic rows, fitting on the rest and scoring those 454 rows against the human out-of-fold scores gives 0.9882 on the fifteen verified features, against 0.9867 for the same rows read out of fold: the protocol is optimistic by 0.0015, and by 0.0006 on all nineteen. That is of the same order as the margins in the table, which is why the margins are the quantity to read and not their exact size: both sides of each margin are scored under the identical arrangement, so the bias applies to both, whereas the above-chance signal in the absolute values is some three hundred times larger than the bias itself.

Table 26: Detection at recursive generation depths 2 to 5, against the human class at their out-of-fold scores, with 95% percentile intervals from 2,000 resamples of connected components at seed 42. The fifteen verified features are the set to read; all nineteen are given for continuity. Both the ensemble and the length control climb monotonically with depth, and the ensemble's margin over the control is under 0.003 at every depth, against 0.043 in domain at depth 1 (0.9851 against 0.9418).

Depth	n	**Avg. words**	**15 feat.**	**Word count alone**	**Margin**
2	285	143.7	0.9972 [0.9946, 0.9992]	0.9959 [0.9933, 0.9981]	+0.0013
3	265	136.3	0.9991 [0.9982, 0.9998]	0.9965 [0.9940, 0.9986]	+0.0026
4	267	131.2	0.9993 [0.9984, 0.9999]	0.9985 [0.9970, 0.9996]	+0.0007
5	262	122.7	0.9998 [0.9995, 1.0000]	0.9996 [0.9988, 1.0000]	+0.0003
All nineteen features, for continuity					
2	285	143.7	0.9975	0.9959	+0.0016
3	265	136.3	0.9990	0.9965	+0.0025
4	267	131.2	0.9994	0.9985	+0.0009
5	262	122.7	0.9999	0.9996	+0.0003

The prediction holds, and it holds in the way that is bad for the detector. Accuracy does rise with depth, monotonically, from 0.9972 to 0.9998; so does the accuracy of word count alone, from 0.9959 to 0.9996, and the gap between them is small at every depth without being monotone in it: 0.0013 at depth 2, 0.0026 at depth 3, then 0.0007 and 0.0003 at depth 5. In domain at depth 1 the same ensemble is 0.043 above the same control; against the shortest, most heavily rewritten documents in the corpus it is 0.0003 above it. Read as a claim about model collapse, a detector that reports 0.9998 on depth-5 rewrites is not detecting iterated generation, it is measuring that iterated rewrites are 300 words shorter than the web documents they came from. The score-to-word-count rank correlation on the depth rows runs -0.47, -0.44, -0.39 and -0.21, so the dependence is visible inside each depth as well as across them. `experiments/scripts/depth_evaluation.py` writes `depth_evaluation.json` and caches its features, and the whole run is a few minutes on a CPU.

AB Per-Signal OOD Analysis

Each family is retrained on its own features alone, in domain with the same five folds as the full ensemble and out of domain on the rows its own extractors score, so a family that declines more documents is measured on fewer of them. All values come from `recompute_syndet_ood.json`, under the single feature implementation of Appendix V. The length column is the mean absolute Spearman correlation between a feature in that family and word count, taken in domain over all 9,462 documents, which is a property of the features rather than of the trained model.

Table 27: Per-family OOD analysis, recomputed. The drop is in domain minus out of domain, so a positive value is a loss. Out-of-domain row counts differ because each family declines a different set of documents. Over the five families the drop and the mean length correlation rank together at Spearman $+0.90$ on five points, which is descriptive and carries no inferential weight.

Family	In-dom.	OOD	OOD n	Drop	Mean $\lvert r \rvert$	Score ρ
S1: Perplexity	0.824	0.699	1,478	$+0.125$	0.19	-0.45
S2: Lexical	0.951	0.634	1,478	$+0.316$	0.60	-0.94
S3: Burstiness	0.750	0.580	1,174	$+0.170$	0.35	-0.57
S4: Syntactic*	0.768	0.794	1,258	-0.025	0.18	-0.46
S5: Similarity	0.820	0.629	1,478	$+0.192$	0.28	-0.72
Full ensemble, 19 features	0.987	0.855	1,074	$+0.132$	–	-0.37

*The four syntactic features are the four the port cannot reproduce against the released caches, and the release pins no spaCy version (Appendix V), so this row mixes domain shift with a parser difference and is the one row in the table that cannot be verified.

Three points stand out. The lexical family is the one that both reads length hardest and transfers worst: its features correlate with word count at a mean absolute Spearman of 0.60, its own score correlates at -0.94 in domain, and it loses 0.316 of AUROC out of domain, 1.6 times the next largest loss (0.192, self-similarity). Perplexity is at the other end, with a mean feature correlation of 0.19 and a loss of 0.125. Across the five families the ranking by length dependence and the ranking by loss agree at Spearman $+0.90$, which on five points is a consistency check and not evidence. The syntactic family gains 0.025 out of domain, and it is also the unverifiable row: the one family that appears to transfer is the one family whose values cannot be reproduced, so we draw no recommendation from it. Finally, the full ensemble loses 0.132, between perplexity's 0.125 and burstiness's 0.170, but it starts far higher and ends far higher: 0.855 out of domain against 0.794 for the unverifiable syntactic family and 0.699 for the best of the four that verify.

AC OOD Domain Visualization

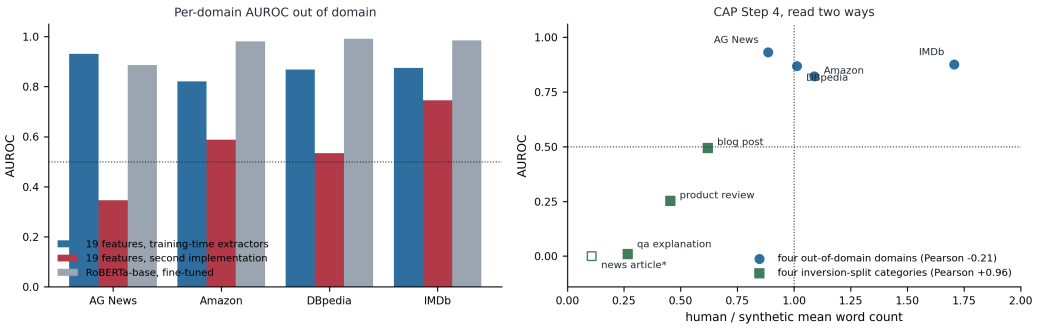

Figure 2: Left: per-domain out-of-domain AUROC, for the ensemble under both feature implementations and for RoBERTa. Under the extractors the model was trained on, AG News is the ensemble's *best* domain at 0.931; under the shipped implementation it is 0.345, below chance, a pattern Appendix V traces to a unit convention. The spread across the four domains is 0.110 for the recomputation and 0.105 for RoBERTa, against 0.400 for the shipped predictions. Right: CAP Step 4 read two ways, both on recomputed scores. Over the four out-of-domain domains the length-ratio-to-AUROC correlation is Pearson -0.21, the wrong sign for the step's premise; over the four categories of the inversion split, where the ratio actually varies from 0.11 to 0.62, it is $+0.96$. Neither reading is established by four points, and the news-article category (open marker) rests on three human documents. Both panels are drawn from the recomputations in `experiments/results/` by `experiments/scripts/make_paper_figures.py`.