# Vulnerability of Fake News Detectors under Flagship LLM Laundering

Jalal Mehdiyev
Institute of Information Technology, Baku, Azerbaijan
Azerbaijan Technical University, Baku, Azerbaijan
*jalal.mehdiyev.s@gmail.com*

***Abstract*— Flagship LLMs can launder fake news into fluent, claim-preserving text that bypasses classifiers trained on human-written corpora. We quantify how three current flagships - GPT-4o, Claude Sonnet 4.6, and Gemini 2.5 Pro - differ in this laundering and how the difference propagates to four detectors. Each of 1,000 ISOT fakes is laundered once per family; a three-judge ensemble isolates 906 claims preserved by all three. On this matched intersection, detector collapse is asymmetric: DeBERTa-v3-large falls from 0.985 to 0.738 F1-score on Claude, RoBERTa-large to 0.857 on Gemini, TF-IDF+LogReg 12.1 on Gemini, TF-IDF+SVM only to 0.936 on GPT. Each detector has a different worst-case family.**

***Keywords*— *fake news detection; generative artificial intelligence; large language models; laundering; natural language processing***

## I. INTRODUCTİON

In today's environment of generative artificial intelligence, a single API call to a flagship LLM is enough to rewrite a fake news article into fluent text. The cost is a fraction of a cent, the time is a few seconds, and the attacker does not need any specialised engineering. Because of this, the most dangerous threat to a deployed fake news detector is not a fine-tuned paraphraser or a chained pipeline, those are the research topics of yesterday, the threat now is one prompt and one API call.

Fake news detectors that are still in production were trained on pre-LLM human-written corpora, because of which they learned the surface features of old hyperpartisan English. A flagship LLM does not write in that style. When the LLM is asked to preserve the false claim and rewrite the article, the laundered output keeps the lie, but loses the surface markers that the detector was relying on. The detector then labels the laundered article as real, and the lie passes the gate.

One of the main problems in evaluation of those detectors is that prior work tends to test only one paraphraser at a time. Either DIPPER, recursive GPT, or a single commercial LLM. Different studies use different prompts, different judge models, and different filters for content drift, because of which numbers from one paper cannot be compared to numbers from another. It is still unclear whether the three current flagship families, OpenAI GPT-4o, Anthropic Claude Sonnet 4.6, and Google Gemini 2.5 Pro, behave the same way against a fixed set of detectors.

This paper closes that gap. We launder each of 1,000 ISOT fake articles once per family with the same prompt and the same temperature, run a three-judge factuality ensemble to isolate articles where all three families preserved the claim, and evaluate four detectors of different capacity on the resulting matched intersection. The contributions are three.

1) Single-round, multi-flagship attack characterisation. Four detectors against three flagship families on a matched intersection of n = 906 claim_ids, with content drift controlled by an ensemble factuality judge. This is the first comparison at this generation of LLMs.
2) Asymmetric collapse claim. Each detector has a different worst-case family. DeBERTa-v3-large breaks on Claude, RoBERTa-large on Gemini, TF-IDF+LogReg on Gemini, TF-IDF+SVM on GPT. Detector capacity does not predict which LLM family is the worst attacker for that detector.
3) Data-shift statistics behind the numbers. Twenty stylometric features measured on the matched intersection show that Claude shifts style most , Gemini least, GPT-4o in between. Style-shift magnitude does not predict attack effectiveness. The direction of the shift, against the detector training distribution, does.

The rest of the paper is organised as follows. Section II reviews the prior art. Section III fixes the threat model. Section IV describes the data and the laundering pipeline, including the data-shift statistics that show how the corpus changed after each LLM rewrite. Section V describes the detectors and the evaluation protocol. Section VI reports the results. Section VII discusses the asymmetric collapse claim and its operational implications. Section VIII concludes.

## II. RELATED WORK

The literature most relevant to this paper sits in three threads. Considering that flagship LLMs are now the cheap attack surface, those three threads have started to converge.

### A *Paraphrase-based evasion of AI-text detectors*

Krishna et al. proposed DIPPER, an 11-billion-parameter paraphraser fine-tuned to evade AI-text detectors [1]. Sadasivan et al. argued that recursive paraphrasing makes reliable detection theoretically impossible [2]. Mitchell et al. proposed DetectGPT, a zero-shot detector built on log-probability curvature, however the method needs white-box access to the source model and so does not apply to closed flagship APIs [3]. Cheng et al. extended the attack line with an adversarial

DOI: 10.25045/forensic.gov.2026.17

paraphraser trained against the detector [4], Fang et al. proposed CoPA, a contrastive decoding-time paraphrase attack [5], and Meng et al. proposed a gradient-based evader against AI-text detectors [6]. All three improve evasion strength but require either training against the detector or control of decoding, because of which they are not single API call attacks. The TempParaphraser of EMNLP 2025 increases evasion by raising decoding temperature [7], also a decoding-control assumption. Those works target detectors of "is this AI", not detectors of "is this fake".

B *Paraphrase-based evasion of fake news detectors*

Zellers et al. introduced Grover, the first generator-detector pair trained jointly on neural fake news, and demonstrated that the strongest detector for a given generator is often a same-family model, an early sign of the detector by family interaction we observe in this paper [8]. Wu, Guo, and Hooi introduced SheepDog, the first work to show that an LLM style rewrite can knock about 38% off the F1-score of a fake news classifier [9]. Park et al. proposed AdStyle, an adversarial-style data augmentation defence that uses LLM rewrites at training time [10]. Das and Dodge ran the closest prior measurement to this paper: GPT-4 plus Pegasus laundering applied to a fake news classifier, with a sentiment-shift explanation of the failures [11]. None of those four studies compared multiple flagship families on a matched intersection with a factuality judge. Przybyla et al. proposed TREPAT, an LLM-driven beam-search rephrasing attack against misinformation classifiers, again with a single paraphraser [12].

C *Linguistic and corpus-level studies*

Ma et al. characterised the linguistic footprint of AI mis/disinformation across eight LIWC-style dimensions with K-S significance tests, a methodology this paper follows in its data-shift section [13]. Guo et al. assembled HC3, the first Human-ChatGPT comparison corpus, and used it to characterise the stylometric gap between human and ChatGPT outputs [14], the same kind of gap we measure across three flagship families. Uchendu et al. proposed TURINGBENCH, the first benchmark for attributing text to its source generator across nineteen open-weights LLMs, which is the methodological precedent for treating per-family stylometric signatures as a forensic signal [15]. Chen and Shu showed that LLM-generated misinformation is harder to catch than human fake news, an effect of about 10 to 20 F1-score values on standard detectors [16]. Zhou et al. showed that AI-generated synthetic lies are judged more credible by humans than human-written fake news [17]. Those works establish that LLM-laundered text differs measurably from human fake news, because of which the detector failure observed in our experiments has a linguistic explanation.

D *Benchmarks for robustness*

RAID [18] and MAGE [19] compile millions of generations across many LLMs and adversarial attacks; those benchmarks show that current detectors are easily fooled by paraphrasing, sampling-strategy variation, and unseen models. Pu et al. previously showed that machine-generated text detectors do not generalise across generator families even at the GPT-2 / GPT-Neo tier, because of which the cross-family pattern we report at the flagship tier is not a surprise but the expected extension of that finding [20]. Our work is complementary to those benchmarks, instead of building a million-row benchmark, we run a controlled three-flagship comparison on a fixed fake news corpus with a content-drift filter, because of which the per-family signal is clean.

Considering all of the above, no prior work has compared the three current flagship families against a panel of fake news detectors of different capacity, on a matched intersection of articles whose false claim is preserved by all three families, with a corpus-level data-shift measurement attached to the F1-score values. This is the gap this paper fills.

## III. THREAT MODEL

### A. *Adversary capability*

The attacker has API access to one flagship LLM (GPT-4o, Claude Sonnet 4.6, or Gemini 2.5 Pro). The attacker can make exactly one API call per fake article. No fine-tuning, no decoding control, no detector-in-the-loop training, no multi-round paraphrasing, no prompt search. The temperature is 0.7. The prompt is generic and asks the model to rewrite the article in different style while preserving the factual claim. This is the cheapest realistic attack a non-specialist content farm can run.

### B. *Adversary goal*

For an article "x" labelled as fake by the detector, produce a rewrite "x'" such that two conditions hold. First, the false claim in "x" is the same false claim in "x'", the rewrite is not a different lie nor a softened version. Second, the detector classifies x' as real, or, in measurement terms, the F1-score of the detector on a population of laundered articles drops below an operationally useful threshold.

### C. *Defender model*

The defender is a fixed off-the-shelf classifier trained on the human-real and human-fake split of ISOT (90/10), then frozen. Four detectors are evaluated, two classical (TF-IDF+Logistic Regression, TF-IDF+Linear SVM) and two transformers (RoBERTa-large, DeBERTa-v3-large). The defender does not see the LLM prompt, does not know which family produced the rewrite, and does not retrain on laundered samples. Black-box defence on text only.

### D. *Out of scope*

Multi-round paraphrasing, prompt search, white-box gradient attacks against the detector, retrieval-grounded fact verification, watermark detection, cross-lingual rewriting, and detector retraining are all explicitly out of scope. Those each constitute a separate research question. The threat model in this paper is the floor of the attack surface, not the ceiling.

### E. *Why claim preservation is enforced externally*

A naive attacker would publish whatever the LLM returned. However, that confounds two failure modes: style laundering and content drift. If a Claude rewrite softens the lie into something close to true, and the detector then labels it real, that is not a successful attack on the detector, it is the LLM doing partial fact correction. To remove this confound, an external three-judge factuality ensemble re-scores every rewrite, and

only articles where the false claim survived in all three families are kept for detector evaluation. This is the matched intersection of $n$ = 906 claim_ids, the same 906 articles across the three families, because of which any F1-score difference between the GPT and Claude conditions is by construction a difference in style, not in content.

## IV. Data and Laundering Pipeline

### A. Source corpus

The ISOT fake news dataset of Ahmed et al. [21] is the source corpus. From the full dataset we sampled 1,000 human-real (HR) and 1,000 human-fake (HF) articles after removing Reuters dateline prefixes from HR (those datelines are a non-content shortcut for any classifier). The HR pool covers 2016-2017 mainstream US political reporting. The HF pool covers the same period, drawn from hyperpartisan publishers with high-affect registers. Each article is assigned a deterministic claim_id, which travels with the article through every later stage and supports the matched-intersection evaluation.

### B. Laundering *prompt* and decoding

Each of the 1,000 HF articles is sent to all three flagships once. The prompt is identical across families and reads: "Rewrite the following news article in a different style, while preserving the same factual claim. Output only the rewritten article." The decoding temperature is 0.7, top-p is the API default. No system prompt customisation, no retry on output, no length control. The output is the laundered V0 version of the HF article. The same claim_id is preserved on the V0 row.

### C. Three-judge factuality ensemble

Each (claim_id, family) pair is scored by three independent LLM judges. The judges are GPT-4o, Claude Sonnet 4.6, and Gemini 2.5 Flash-Lite. The Flash-Lite variant is used in the Gemini judge role because thinking-token truncation breaks JSON output in the larger Gemini 2.5 variants, because of which Flash-Lite is the only Gemini that returns reliable verdicts under the strict-JSON prompt. Each judge runs at temperature 0.0 with a different verification prompt and returns preserved or not_preserved for the claim. The ensemble accepts a rewrite if the majority of judges marked it preserved. Table I reports the per-family preservation outcome and the size of the strict three-way intersection.

TABLE I. PER-FAMILY PRESERVATION RATES AND MATCHED INTERSECTION SIZE

| Family | n_judged | n_preserved | Preservation rate | In 3-way intersection |
|---|---|---|---|---|
| GPT-4o | 995 | 993 | 0.998 | 906 |
| Claude Sonnet 4.6 | 993 | 991 | 0.998 | 906 |
| Gemini 2.5 Pro | 929 | 920 | 0.990 | 906 |

The strict three-way intersection has 906 claim_ids, the same 906 across all three families. All detector evaluations in Section VI are run on those 906. Gemini's slightly lower judged count (929 of 1,000) reflects API-side refusals on a small share of strongly hyperpartisan inputs. Considering that the preservation rates are all above 0.99, content drift is not the dominant variable in the experiment, the dominant variable is style.

### D. Preliminary notations and definitions

Before presenting the data-shift statistics we fix the notation used in Tables II and III and in formula 1. Three article populations are compared. HR is the human-real subset of ISOT (n = 906 on the matched intersection). HF is the human-fake subset of ISOT, the original pre-laundering articles (n = 906). V0 is the laundered output, the single rewrite produced by one flagship family for the same claim_id, also n = 906 per family. The three families are denoted GPT (GPT-4o), Claude (Claude Sonnet 4.6), and Gemini (Gemini 2.5 Pro).

The stylometric parameters in Table II are computed per document and then averaged within each population. Each parameter has a short identifier and a fixed unit:

- doc_len_tokens is document length measured in spaCy tokens, including punctuation:

$$\text{doc\_len\_tokens} = N \tag{1}$$

where $N$ is the total number of tokens in the document.

- avg_sent_len is the mean sentence length in tokens, averaged over sentences within a document:

$$\text{avg\_sent\_len} = \frac{N}{S} \tag{2}$$

where $S$ is the number of sentences.

- ttr is the type-token ratio, the count of unique lemmas divided by the total token count, a measure of lexical diversity:

$$ttr = \frac{number\ of\ unique\ lemmas}{total\ number\ of\ tokens} \tag{3}$$

- noun_rate, adj_rate, and pronoun_rate are the share of tokens with spaCy POS tag NOUN, ADJ, and PRON, respectively, expressed as a fraction of the document token count:

$$\text{noun\_rate} = \frac{\#Noun}{N} \tag{4}$$

$$\text{adj\_rate} = \frac{\#Adj}{N} \tag{5}$$

$$\text{pronoun\_rate} = \frac{\#Pron}{N} \tag{6}$$

- subord_clause_rate is the share of subordinate-clause heads ($advcl$, $ccomp$, $xcomp$ dependency labels) over all tokens:

$$\text{subord\_clause\_rate} = \frac{\#(advcl+ccomp+xcomp)}{N} \tag{7}$$

where $advcl$ is adverbial clause, $ccomp$ is clausal complement with subject, $xcomp$ is open clausal complement.

- punct_density, quote_density, and exclaim_density are the share of punctuation tokens, of single and double quotation marks, and of exclamation marks, respectively, again as a fraction of all tokens in the document.

$$\text{punct\_density} = \frac{\#punctuation}{N} \tag{8}$$

$$\text{quote\_density} = \frac{\#quotes}{N} \quad (9)$$

$$\text{exclaim\_density} = \frac{\#!}{N} \quad (10)$$

- vader_compound is the VADER compound sentiment score which ranges from -1 (strongly negative) to +1 (strongly positive).

$$-1 \leq \text{vader\_compund} \leq 1 \quad (11)$$

- vader_neg is the VADER negative-sentiment score with range 0 to 1.

$$-1 \leq \text{vader\_neg} \leq 1 \quad (12)$$

- certainty_rate is the share of certainty markers (definitely, certainly, surely, undoubtedly, and similar tokens) over all tokens of the document.

$$\text{certainty\_rate} = \frac{\#certainty\ markers}{N} \quad (13)$$

All features are computed with spaCy 3.7 (en_core_web_sm) and the VADER sentiment model; those tools are the same across HR, HF, and every V0 family.

Two derived quantities also appear in the tables. Δ% is the percent change of a parameter mean from the HF baseline to the V0 family as represented in Eq.(14):

$$\Delta\% = \frac{(\mu_{family} - \mu_{HF})}{\mu_{HF}} \times 100 \quad (14)$$

where Δ% is the percent change from the human-fake baseline, $\mu_{family}$ is the mean value of the stylometric feature, computed across all documents in the laundered V0 corpus of a given flagship LLM family, $\mu_{HF}$ is the mean value of the same stylometric feature, computed across all documents in the HF baseline corpus, restricted to the matched intersection used in the comparison.

The aggregate style distance $\sigma$ represented in Eq.(15) in Table III is expressed in sigma units of the HF baseline, the formula and the meaning of those $\sigma$ units are given in Eq.(15) below. Larger $\sigma$ means the laundered corpus sits further from the HF baseline in the 20-feature style space.

$$\sigma = \sqrt{\sum_{i=1}^{n} \left( \frac{\mu_{family,i} - \mu_{HF,i}}{\sigma_{HF,i}} \right)^2} \quad (15)$$

where $n$ is the number of stylometric features summed over, i is the feature index, running from 1 to n, $\mu_{family,i}$ is the mean of feature i, computed across all documents in the laundered V0 corpus of the given flagship LLM family, $\mu_{HF,i}$ is the mean of feature i, computed across all documents in the human-fake (HF) baseline corpus, $\sigma_{HF,i}$ is the standard deviation of feature i on the human-fake (HF) baseline corpus. Used as the per-feature z-normalisation scale so that features measured on different units (e.g., raw counts vs. ratios bounded in [0, 1]) contribute comparably to the sum. Eq. (16) shows the sample standard deviation (using M − 1 in the denominator, also known as Bessel's correction), which is the standard estimator used by numpy / pandas / scipy with ddof=1 and is appropriate when the human-fake corpus is treated as a sample rather than the full population.:

$$\sigma_{HF,i} = \sqrt{\frac{1}{M-1} \sum_{j=1}^{M} \left( x_{i,j} - \mu_{HF,i} \right)^2} \quad (16).$$

*E. Data-shift statistics: how the corpus changed after each LLM rewrite*

To make the attack mechanism concrete we measured 20 stylometric features per document on the matched intersection: length, sentence statistics, type-token ratio, five POS rates (noun, verb, adjective, adverb, pronoun), mean dependency distance, subordinate-clause and passive-voice rates, four punctuation-and-emphasis features (punctuation density, quote density, exclaim density, content-word ratio), three VADER affective features, and two epistemic features (hedging rate, certainty rate). Features are computed with spaCy 3.7 (en_core_web_sm) and the VADER sentiment model.

*F. Feature-level shift summary*

Fig.1 shows the per-document distribution of six representative parameters (document length, sentence length, type-token ratio, pronoun rate, punctuation density, VADER compound sentiment) as boxplots, side by side for the HF baseline and each laundered family. The dashed line on every panel marks the HF median, cause of which the magnitude and direction of the shift can be read directly.

Fig.2 condenses the change into a single Δ% heatmap over all 13 features and three families, which is the visual companion of Table II. Table II itself reports the mean per-feature value of the HF baseline alongside the laundered V0 of each family. All three families significantly shift the corpus across the majority of features (K-S two-sample test, V0 vs HF: GPT-4o 17 of 20 features at $p < 0.001$, Claude 15 of 20, Gemini 18 of 20).

Three observations. First, all three flagships shorten sentences by 26 to 32%, drop pronouns by 22 to 29%, drop exclaim density by about half, drop certainty markers by 35 to 50%, and add quotation marks and punctuation. Those changes are exactly the surface markers that distinguish hyperpartisan ISOT HF from neutral newswire registers, because of which a detector trained to look at those markers will lose its grip on the laundered text.

Second, the families differ in direction on affective features. Claude pushes compound sentiment further negative (-0.10 to -0.15), because of which the rewrite is more polarised than the original. Gemini neutralises compound sentiment to near zero (-0.10 to +0.02), which flips the sign of the article's affective tone. GPT-4o partially neutralises (-0.10 to -0.05). On negative sentiment specifically, Gemini cuts vader_neg by 19.6%, the other two leave it nearly unchanged.

Third, GPT-4o is the only family that lengthens the article (+17%). Claude is roughly length-neutral (+3%), Gemini is exactly length-neutral (-0.4%). Because of this, when a detector reacts to article length or length-correlated features, GPT-4o is the family that pushes the article furthest off the HF length distribution.

Aggregate style distance. To summarise the per-family shift in one number we computed the euclidean distance between each V0 family's mean feature vector and the HF baseline mean, in z-normalised space. This is the standardised style distance sigma.

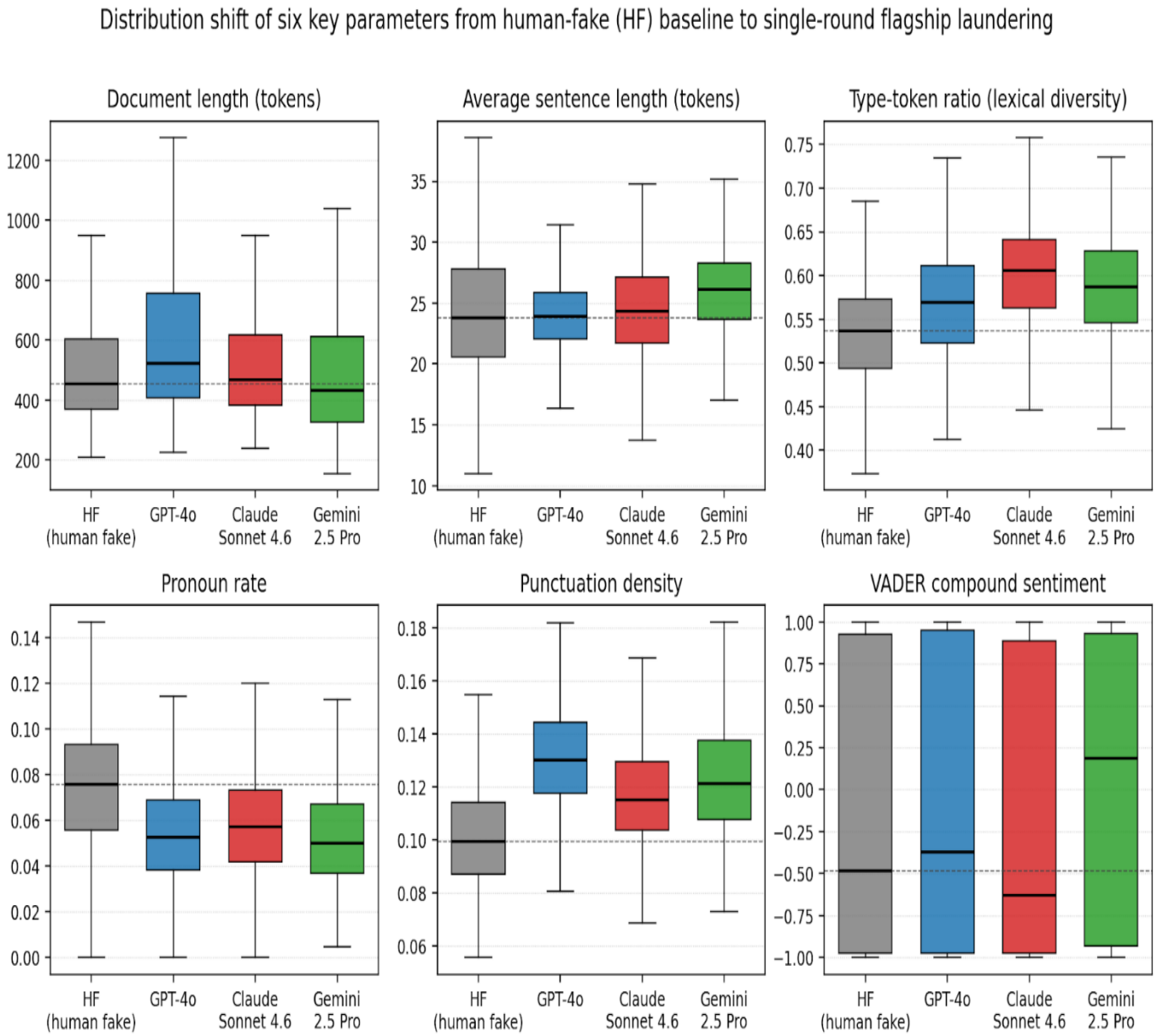


Figure 1. Per-document distributions of six key parameters baseline and each laundered V0 family.

TABLE II. STYLOMETRIC DRIFT FROM HF BASELINE TO LAUNDERED V0

| Feature | HF | GPT-4o (Δ%) | Claude (Δ%) | Gemini (Δ%) |
|---|---|---|---|---|
| doc_len_tokens | 519.6 | 609.7 (+17.3) | 535.8 (+3.1) | 517.3 (-0.4) |
| avg_sent_len | 35.40 | 23.93 (-32.4) | 24.55 (-30.6) | 26.03 (-26.4) |
| ttr | 0.533 | 0.565 (+6.1) | 0.602 (+13.0) | 0.585 (+9.8) |
| noun_rate | 0.168 | 0.191 (+13.9) | 0.182 (+8.5) | 0.190 (+13.3) |
| adj_rate | 0.060 | 0.069 (+14.2) | 0.072 (+20.1) | 0.071 (+18.3) |
| pronoun_rate | 0.075 | 0.054 (-27.7) | 0.059 (-22.0) | 0.053 (-29.1) |
| subord_clause_rate | 0.090 | 0.079 (-13.2) | 0.082 (-8.9) | 0.080 (-12.0) |
| punct_density | 0.099 | 0.132 (+33.1) | 0.118 (+18.9) | 0.124 (+25.4) |
| quote_density | 0.000475 | 0.008 (+1506) | 0.014 (+2856) | 0.004 (+774) |
| exclaim_density | 0.001 | 0.001 (-51.5) | 0.001 (-54.8) | 0.001 (-50.5) |
| vader_compound | -0.10 | -0.05 (-48.9) | -0.15 (+44.9) | +0.02 (-123.7) |
| vader_neg | 0.094 | 0.093 (-1.6) | 0.092 (-2.5) | 0.076 (-19.6) |
| certainty_rate | 0.005 | 0.003 (-47.1) | 0.003 (-35.1) | 0.003 (-49.6) |

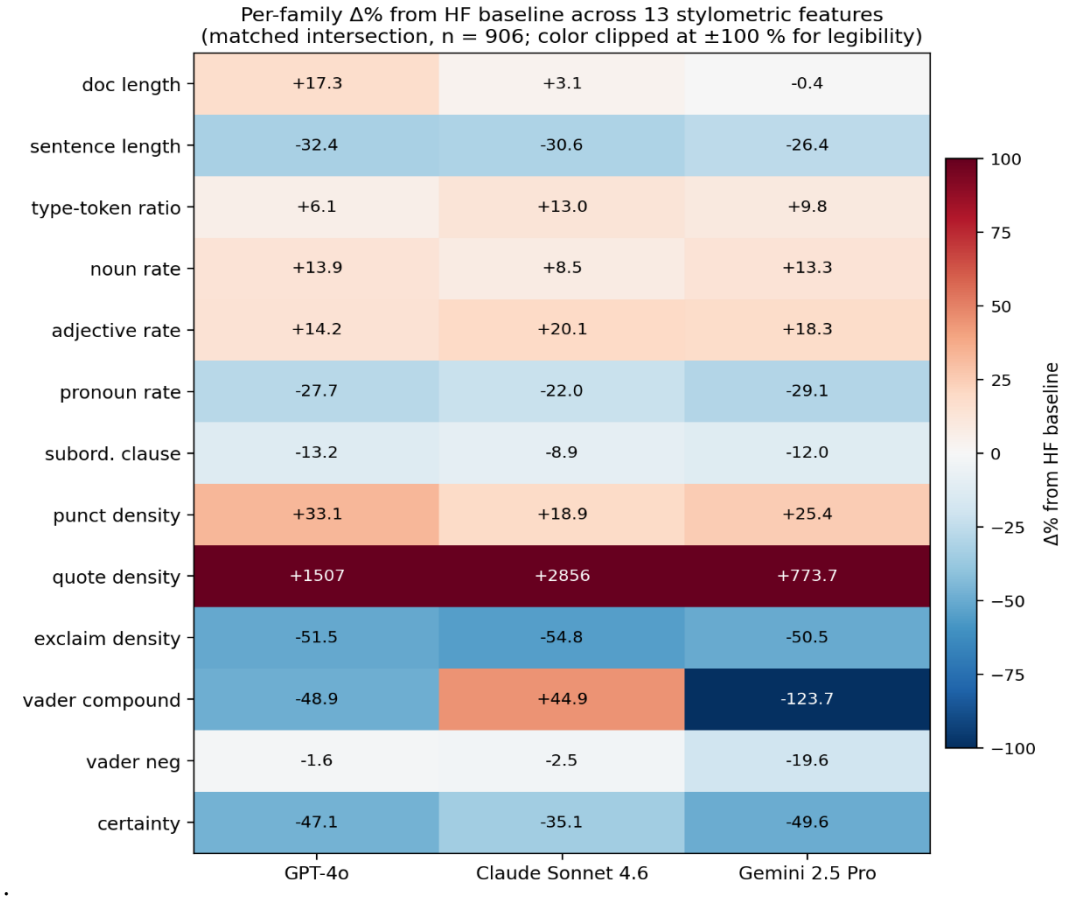


Figure 2. Δ% from the HF baseline across 13 stylometric features and three flagship laundering families.

TABLE III. AGGREGATE STYLE DISTANCE FROM HF BASELINE, Z-NORMALISED (MATCHED, N = 906)

| Family | Style distance σ | Interpretation |
|---|---|---|
| Gemini 2.5 Pro | 2.55 σ | Most conservative paraphraser |
| GPT-4o | 3.44 σ | Middle |
| Claude Sonnet 4.6 | 5.66 σ | Most aggressive style shift |

Considering Table III alongside the F1-score values in Section VI, the asymmetric-collapse claim emerges immediately. The largest style shift (Claude, 5.66 sigma) breaks the largest detector (DeBERTa-v3-large, drop of 24.7 F1-score values). The smallest style shift (Gemini, 2.55 sigma) breaks the second-largest detector (RoBERTa-large) and one of the classical baselines (TF-IDF+LogReg). The middle style shift (GPT-4o, 3.44 sigma) is the worst case for the most robust classical baseline (TF-IDF+SVM). Style-shift magnitude does not predict which detector breaks, the direction of the shift does, because of which detector capacity is not a reliable proxy for detector robustness.

## V. DETECTORS AND EVALUATİON PROTOCOL

Four detectors span four orders of magnitude in capacity. TF-IDF+Logistic Regression uses unigram and bigram TF-IDF features with a calibrated logistic regression head (classical baseline 1). TF-IDF+Linear SVM uses the same TF-IDF features with a calibrated linear SVM head (classical baseline 2). RoBERTa-large is a 355 M parameter transformer fine-tuned on HR vs HF for 3 epochs, fp16, fixed seed. DeBERTa-v3-large is a 304 M parameter transformer with the same fine-tuning protocol as RoBERTa-large.

All four detectors are trained once on the original ISOT HR vs HF training split, then frozen. No detector sees laundered V0 samples during training, those are held out for evaluation only. This matches a deployment scenario where the model was trained on pre-LLM data and is now attacked at inference time.

For each detector we report F1-score on the fake class under four conditions. HR vs HF (matched): the detector is evaluated on the matched intersection only, HR (n = 906) versus original HF (n = 906). This is the in-distribution baseline. HR vs V0-GPT, HR vs V0-Claude, HR vs V0-Gemini: HR (n = 906) versus the laundered V0 of that family (n = 906). The fake class label is held constant: V0 is treated as fake (because the false claim is preserved), HR remains real. The drop from the matched HR vs HF condition to each laundered condition is the attack effectiveness.

The matched 906 claim_ids, the laundering prompt, the three judge prompts, the detector training scripts, the feature-extraction scripts, and the random seeds are released with the paper. All experiments run on a single Colab T4 GPU in under two hours.

## VI. RESULTS

Table IV reports the headline F1-score values, four detectors, four conditions per detector. Fig.3 visualises the absolute F1-score of each detector under each condition.

TABLE IV. F1-SCORE ON THE FAKE CLASS ACROSS DETECTORS AND LAUNDERING FAMILIES

| Detector | HR vs HF | GPT | Claude | Gemini | Worst Δ (family) |
|---|---|---|---|---|---|
| TF-IDF + LogReg | 0.977 | 0.857 | 0.916 | 0.856 | -12.1 (Gemini) |
| TF-IDF + SVM | 0.996 | 0.936 | 0.975 | 0.955 | -6.0 (GPT) |
| RoBERTa-large | 0.987 | 0.908 | 0.929 | 0.857 | -13.0 (Gemin) |
| DeBERTa-v3-large | 0.985 | 0.894 | 0.738 | 0.863 | -24.7 (Claude) |

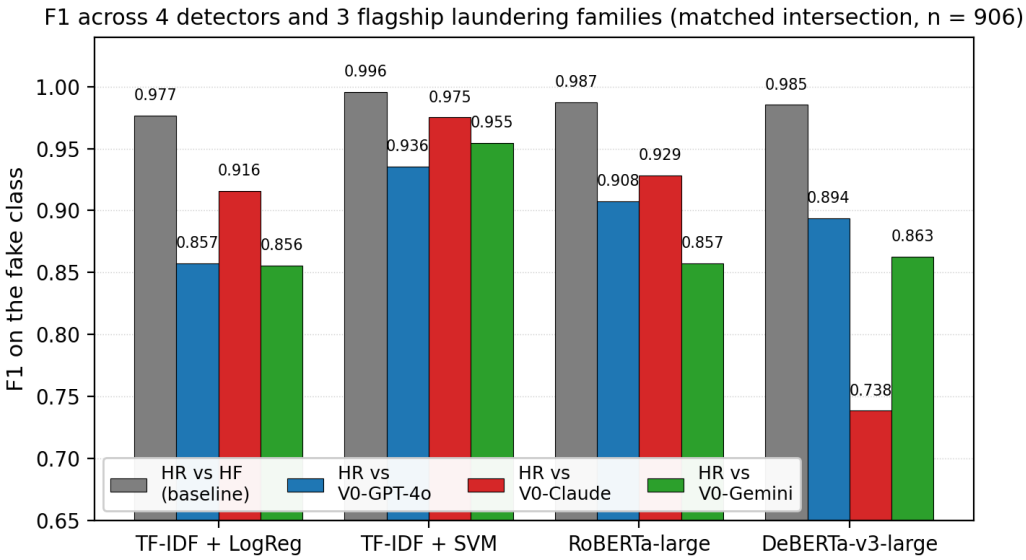


Figure 3. Absolute F1-score of four detectors on the fake class under the in-distribution baseline and the three single-round flagship laundering families (matched, n = 906)

Three findings follow:

**Finding 1**: every detector breaks, but capacity does not predict robustness. All four detectors lose F1-score under at least one laundering family. The largest loss is on the largest detector (DeBERTa-v3-large drops 24.7 points on Claude). The smallest loss is on the second classical baseline (TF-IDF+SVM drops only 6.0 points on GPT). Capacity alone, parameter count, is not a useful proxy for laundering robustness on this benchmark. The data-shift evidence supports this, the K-S tests in Section IV-D show that 15 to 18 of 20 features differ significantly between HF and each laundered family at $p < 0.001$, because of which every detector that reads surface features is exposed to a measurably shifted distribution regardless of the family.

**Finding 2**: each detector has a different worst-case family. DeBERTa-v3-large is broken hardest by Claude. RoBERTa-large is broken hardest by Gemini. TF-IDF+LogReg is also broken hardest by Gemini, however the absolute F1-score floor is far higher (0.856) than DeBERTa's floor on Claude (0.738). TF-IDF+SVM is broken hardest by GPT-4o, which is the only family that meaningfully changes article length. The four worst-case pairings span all three families, because of which an analyst cannot pick a single LLM to stress-test a pipeline, all three are needed.

**Finding 3**: style-shift magnitude does not match attack effectiveness. Claude has the largest style shift from HF (5.66 sigma, Table III) and is the worst case for DeBERTa-v3-large only. Gemini has the smallest style shift (2.55 sigma) and yet is the worst case for two of the four detectors. GPT-4o sits in the middle on style shift (3.44 sigma) and is the worst case for the most robust detector. The mapping from style-shift magnitude to F1-score loss is not monotonic, because of which the operational variable is not how much the LLM changes the text, it is in which feature direction.

Dataset shift as additional proof of the asymmetric collapse. Each of the three findings above can be read off Fig.1, Fig.2, and Table II directly, because of which the F1-score values in Table IV are not a stand-alone observation, they are predicted by the shape of the dataset shift.

On Finding 1, the boxplots in Fig.1 show that all three families compress the sentence-length distribution, raise the type-token ratio, drop the pronoun rate, and lift the punctuation density compared with HF. The HF median lines on Fig.1 sit clearly outside the laundered interquartile boxes for sentence length, pronoun rate, and punctuation density, which means the laundered V0 is no longer drawn from the same distribution the detectors were trained on. Because of this every detector loses ground, the question is only by how much.

On Finding 2, the heatmap in Fig.2 explains why each detector has a different worst-case family. Claude's row shows the largest absolute shifts on type-token ratio (+13%), adjective rate (+20%), and quote density (+2856%), those are the features that the largest transformer (DeBERTa-v3-large) had the headroom to learn during training, cause of which Claude is its worst case. Gemini's row is the most muted in absolute shift on most features, but it carries the only sentiment sign flip on the matched intersection (vader_compound from -0.10 to +0.02) and the only meaningful drop in vader_neg (-19.6%), those affective features are exactly what TF-IDF+LogReg and RoBERTa-large lean on, cause of which Gemini is the worst case for both. GPT-4o's row shows the only positive length shift (+17.3%) plus the largest punctuation and quote-density jumps relative to text size, those are the n-gram-count features that drive TF-IDF+SVM, cause of which GPT-4o is its worst case.

On Finding 3, Table III together with Table IV gives a direct

ranking inversion. The order of families by aggregate style distance is Gemini < GPT-4o < Claude, however the order of families by F1-score damage is detector dependent, with Gemini at the top of damage for two of the four detectors. The aggregate sigma from Table III is therefore not a sufficient statistic for attack strength, the per-feature pattern in Fig.2 is. Each detector reads a different subset of the 20 features (Fig.1 shows which subset moves most for that family), and each LLM family shifts a different subset of those features (Fig.2 shows the per-family heatmap), cause of which the worst-case pairing of LLM and detector is determined by overlap of those subsets, not by the size of either side.

## VII. DİSCUSSİON

Detector capacity is not detector robustness. The largest model in the panel (DeBERTa-v3-large) is the most damaged detector. The most robust detector is a classical baseline (TF-IDF+SVM). This contradicts the common assumption in fake news pipelines that scaling up the detector is a defence against laundering. On this benchmark, scaling up is a liability, because larger models read more of the surface features that flagship LLMs neutralise.

Each pipeline has its own worst-case LLM. Considering Finding 2 above, a deployed pipeline cannot be hardened by stress-testing against a single flagship. A pipeline that only tests against GPT-4o will think TF-IDF+LogReg is robust (drop of 12 points), it is not, Gemini also drops it 12 points but for different reasons. A pipeline that only tests against Claude will think TF-IDF+SVM is fine, it is, however the pipeline will miss that GPT-4o is its worst attacker. Because of this, multi-flagship stress testing should be standard.

The single-round threat model is the floor. Our numbers are obtained with one API call, generic prompt, no retries. Recursive laundering, prompt search, and decoding control would all increase the F1-score drops. The numbers in Table IV are therefore a lower bound on attack strength under flagship LLMs. The fact that DeBERTa-v3-large already drops 24.7 points under the cheapest attack is the operational warning, no fancy adversary is needed.

Limitations. Three limitations should be flagged. First, ISOT is a 2016-2017 corpus, because of which detectors trained on it are pre-LLM by construction. The laundering vulnerability we report is in part a property of training-distribution drift between 2017 and 2026, not only of the LLM itself. ReCOVery and FakeNewsNet replications would clarify how much of the drop is era-driven and how much is LLM-driven. Second, the three-judge ensemble is itself made of LLMs, because of which a small fraction of preservation decisions could be incorrect, those are bounded above by 1 to 2 % per family per Table I. Third, the matched-intersection design is conservative: it discards articles where any of the three families failed to preserve the claim, those discarded articles may contain harder cases.

Operational implication. A defender should not pick the largest detector and call the pipeline robust. The defender should run a multi-flagship stress test, look at per-family F1, and either retrain on laundered samples (the AdStyle line of work [10]) or build an ensemble that combines a classical model with a transformer. Considering the F1-score floor of TF-IDF+SVM under GPT-4o (0.936) and the F1-score ceiling of DeBERTa-v3-large under Claude (0.738), a simple ensemble of those two would already be more robust than either alone.

## CONCLUSİON

A single API call to a flagship LLM is enough to break fake news detectors trained on pre-LLM human-written data. We measured this break across three flagship families (GPT-4o, Claude Sonnet 4.6, Gemini 2.5 Pro) and four detectors (TF-IDF+LogReg, TF-IDF+SVM, RoBERTa-large, DeBERTa-v3-large) on a matched intersection of 906 ISOT articles whose false claim was preserved by all three flagships. Detector collapse is asymmetric: DeBERTa-v3-large drops 24.7 F1-score points on Claude, RoBERTa-large drops 13.0 on Gemini, TF-IDF+LogReg drops 12.1 on Gemini, TF-IDF+SVM drops 6.0 on GPT-4o. Each detector has a different worst-case family, because of which detector capacity is not a reliable proxy for detector robustness.

The data-shift statistics behind those numbers explain the mechanism. Claude shifts style most aggressively (5.66 sigma from HF baseline), Gemini least (2.55 sigma), GPT-4o in the middle (3.44 sigma). Style-shift magnitude does not predict attack effectiveness, the direction of the shift does. The most-damaging family for a given detector is the one whose shift direction overlaps with the feature subset that detector relied on during training.

For a digital-forensics practitioner the operational message is concrete. A deployed fake news pipeline should be stress-tested against all three flagship families, not one, and an ensemble of one classical and one transformer detector is a cheap path to laundering robustness. Future work will extend this comparison to ReCOVery and FakeNewsNet to disentangle era-effects from LLM-effects, add multi-round laundering to measure the ceiling of the attack, and integrate the data-shift signature into an attribution head that can label a laundered article with its source flagship family.

## REFERENCES

[1] K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer, "Paraphrasing Evades Detectors of AI-Generated Text, but Retrieval Is an Effective Defense (DIPPER)," in Proc. NeurIPS, 2023. https://doi.org/10.48550/arXiv.2303.13408

[2] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi, "Can AI-Generated Text Be Reliably Detected?", inTransactions on Machine Learning Research, 2024. https://doi.org/10.48550/arXiv.2303.11156

[3] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, "DetectGPT: Zero-Shot Machine-Generated Text Detection Using Probability Curvature," in Proc. ICML, 2023. https://doi.org/10.48550/arXiv.2301.11305

[4] Y. Cheng, V. S. Sadasivan, M. Saberi, S. Saha, and S. Feizi, "Adversarial Paraphrasing: A Universal Attack for Humanizing AI-Generated Text," in Proc. NeurIPS, 2025. https://doi.org/10.48550/arXiv.2506.07001

[5] H. Fang, J. Kong, T. Zhuang, Y. Qiu, K. Gao, B. Chen, S.-T. Xia, Y. Wang, and M. Zhang, "Your Language Model Can Secretly Write Like Humans: Contrastive Paraphrase Attacks on LLM-Generated Text Detectors (CoPA)," in Proc. EMNLP, 2025, pp. 8585-8602. https://doi.org/10.18653/v1/2025.emnlp-main.433

[6] W. Meng, S. Fan, C. Wei, M. Chen, Y. Li, Y. Zhang, Z. Zhang, and W.

Chen, "GradEscape: a gradient-based evader against AI-generated text detectors," in Proc. 34th USENIX Secur. Symp. (SEC '25), Seattle, WA, USA, Aug. 2025, art. no. 10. https://doi.org/10.5281/zenodo.15586856 .

[7] J. Huang, R. Zhang, J. Su, and Y. Chen, "TempParaphraser: 'Heating Up' Text to Evade AI-Text Detection through Paraphrasing," in Proc. EMNLP, 2025, paper 1607. https://doi.org/10.18653/v1/2025.emnlp-main.1607

[8] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, "Defending Against Neural Fake News," in Proc. NeurIPS, 2019. https://doi.org/10.48550/arXiv.1905.12616

[9] J. Wu, J. Guo, and B. Hooi, "Fake News in Sheep's Clothing: Robust Fake News Detection Against LLM-Empowered Style Attacks (SheepDog)," in Proc. KDD, 2024. https://doi.org/10.1145/3637528.3671977

[10] S. Park, S. Han, X. Xie, J.-G. Lee, and M. Cha, "AdStyle: Adversarial Style Augmentation via Large Language Model for Robust Fake News Detection," in Proc. ACM Web Conf. (WWW), 2025. https://doi.org/10.48550/arXiv.2406.11260

[11] R. K. Das and J. Dodge, "Fake News Detection After LLM Laundering: Measurement and Explanation," arXiv:2501.18649, 2025. https://doi.org/10.48550/arXiv.2501.18649

[12] P. Przybyla, E. McGill, and H. Saggion, "Attacking Misinformation Detection Using Adversarial Examples Generated by Language Models (TREPAT)," in Proc. EMNLP, 2025. https://doi.org/10.18653/v1/2025.emnlp-main.1405

[13] Y. Ma, X. Zhang, J. Ren, R. Wang, M. Wang, and Y. Chen, "Linguistic features of AI mis/disinformation and the detection limits of LLMs," Nature Communications, vol. 16, 2025. https://doi.org/10.1038/s41467-025-67145-1

[14] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, and Y. Wu, "How Close Is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection (HC3)," arXiv:2301.07597, 2023. https://doi.org/10.48550/arXiv.2301.07597

[15] A. Uchendu, Z. Ma, T. Le, R. Zhang, and D. Lee, "TURINGBENCH: A Benchmark Environment for Turing Test in the Age of Neural Text Generation," in Findings of EMNLP, 2021. https://doi.org/10.18653/v1/2021.findings-emnlp.172

[16] C. Chen and K. Shu, "Can LLM-Generated Misinformation Be Detected?" in Proc. ICLR, 2024. https://doi.org/10.48550/arXiv.2309.13788

[17] J. Zhou, Y. Zhang, Q. Luo, A. G. Parker, and M. De Choudhury, "Synthetic Lies: Understanding AI-Generated Misinformation and Evaluating Algorithmic and Human Solutions," in Proc. CHI, 2023. https://doi.org/10.1145/3544548.3581318

[18] L. Dugan, A. Hwang, F. Trhlik, 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 Proc. ACL, 2024. https://doi.org/10.18653/v1/2024.acl-long.674

[19] Y. Li, Q. Li, L. Cui, W. Bi, Z. Wang, L. Wang, L. Yang, S. Shi, and Y. Zhang, "MAGE: Machine-Generated Text Detection in the Wild," in Proc. ACL, 2024, pp. 36-53. https://doi.org/10.18653/v1/2024.acl-long.3

[20] X. Pu, J. Zhang, X. Han, Y. Tsvetkov, and T. He, "On the Zero-Shot Generalization of Machine-Generated Text Detectors," in Findings of EMNLP, 2023. https://doi.org/10.18653/v1/2023.findings-emnlp.318

[21] H. Ahmed, I. Traore, and S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," in Proc. Int. Conf. Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments (ISDDC), LNCS vol. 10618, Springer, 2017, pp. 127-138. https://doi.org/10.1007/978-3-319-69155-8_9