DeBERTa-ConPara: Attack-Aware and Deployment-Realistic Detection of AI-Generated Text

Mohamed Mady[1,2] **Yupei Li**[3] **Johannes Reschke**[2] **Björn W. Schuller**[1,3]

[1]CHI, Chair of Health Informatics, Technical University of Munich, Germany
[2]Smart Embedded Systems Lab, OTH Regensburg, Germany
[3]GLAM, Group on Language, Audio & Music, Imperial College London, UK

mohamed.mady@tum.de yupei.li22@imperial.ac.uk
johannes.reschke@oth-regensburg.de schuller@tum.de

Abstract

Robust detection of AI-generated text under deployment conditions is challenging: distribution shifts across domains and generators, adversarial perturbations of the input surface, and the absence of target-domain labels for threshold calibration all degrade detectors that perform well in-domain.

We present **DeBERTa-ConPara**, a deployment-oriented detector combining attack-aware Unicode preprocessing with a contextual transformer encoder trained over HC3 Plus, M4, MAGE and RAID. Our central finding is that preprocessing acts in opposite directions depending on where it is applied: normalising the *training* corpus deduplicates it, collapsing 35.4 % of RAID rows into copies of their clean siblings and deleting the adversarial supervision, whereas normalising at *inference* is an effective defence. A factorial varying the two placements independently identifies raw training with normalised inference as the best configuration, reaching **99.61 % AUROC**, **99.01 %** TPR@5 % FPR and **96.57 %** TPR@1 % FPR on the official RAID hidden test, alongside **93.14 %** average balanced accuracy across HC3 Plus and MAGE under a fixed threshold. The gain is confined to two of twelve attack classes: homoglyph and zero-width-space insertion rise from 11.05 % and 1.12 % to 96.98 %. The same signature reproduces in a zero-shot detector of different architecture, showing the effect belongs to the attacks rather than to our model. We additionally report two negative results: semantic-invariance augmentation through paraphrasing and supervised contrastive learning (ConPara) does not improve the best configuration, and the handcrafted feature-fusion branch is inert in distribution and harmful outside it.

1 Introduction

Recent advances in large language models (LLMs) have dramatically improved the fluency and realism of AI-generated text, making machine-written content increasingly difficult to distinguish from human writing across domains such as education, journalism, scientific publishing, and online communication (Achiam et al., 2023; Gemini Team, 2023; Stanford Institute for Human-Centered Artificial Intelligence (HAI), 2025). While these capabilities enable powerful generative applications, they also introduce major concerns including misinformation propagation, academic dishonesty, synthetic reviews, and coordinated inauthentic behaviour (Turnitin LLC, 2024; Chakraborty et al., 2024; Goldstein et al., 2023; Stokel-Walker, 2022). Consequently, robust AI-generated text detection has become a critical challenge for trustworthy deployment of generative AI systems.

Existing detection approaches fall into three families, surveyed in Section 2: supervised classifiers (Li et al., 2025), zero-shot statistical detectors (Bao et al., 2024), and watermarking (Wouters, 2024). Watermarking requires cooperation from the generator and leaves already-generated text untouched, so post-hoc detection remains the practical setting and is the one we address. Although many detectors achieve strong in-domain performance, robustness degrades substantially under unseen generators, domain shifts, decoding variations, and adversarial perturbations (Dugan et al., 2024; Chakraborty et al., 2024). Lightweight surface-form attacks such as homoglyph substitutions and zero-width Unicode characters can severely disrupt tokenisation-sensitive pipelines and distort linguistic and statistical feature distributions while preserving semantic content.

Despite recent progress, two deployment challenges remain insufficiently addressed. First, most evaluation protocols implicitly assume access to target-domain labels by independently tuning decision thresholds for each dataset or domain at test time, an assumption that rarely holds in practical deployment.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 1792–1824
November 6-10, 2026 ©2026 Association for Computational Linguistics

Second, many adversarial attacks operate primarily at the character and formatting level, exploiting weaknesses in Unicode handling and tokenisation before semantic modelling begins. Existing detectors therefore often focus on representation learning while overlooking vulnerabilities in preprocessing and tokenisation pipelines.

We introduce **DeBERTa-ConPara**, a deployment-oriented detector built on `microsoft/deberta-v3-large` (He et al., 2023) that combines attack-aware preprocessing with a multi-dataset adversarial training corpus.

On the official RAID hidden-test benchmark, DeBERTa-ConPara reaches $99.01\,\%$ TPR@5 % FPR while sustaining $93.14\,\%$ average balanced accuracy across HC3 Plus and MAGE under a fixed threshold. Several leaderboard systems score higher on RAID alone; our claim is the combination, restricted to the English domains, generators and attacks evaluated here.

Our contributions are:

- A controlled $2 \times 2 \times 2$ factorial, all eight cells submitted to the official RAID hidden test, showing that Unicode preprocessing defends at inference time but removes adversarial supervision when applied to the training corpus.
- An attribution of that robustness to two of twelve attack classes, reproduced in a zero-shot detector of different architecture: the effect belongs to the attacks, not to our model.
- A multi-dataset curation strategy that keeps cross-dataset generalisation alongside that robustness, where systems tuned to one benchmark do not.
- Two documented negative results: ConPara augmentation does not improve the best configuration, and the feature-fusion branch is inert in distribution and harmful out of it.

2 Related Work

AI-generated text detection has rapidly evolved alongside advances in large language models (LLMs), leading to a broad spectrum of detection paradigms including zero-shot statistical methods, supervised detectors, watermarking approaches, and adversarially robust detection frameworks (Wu et al., 2025). Despite substantial progress, existing methods continue to struggle under realistic deployment conditions involving unseen generators, domain shifts, and adversarial perturbations.

2.1 Zero-Shot and Statistical Detection

Zero-shot detectors identify machine-generated text without task-specific supervised training by exploiting statistical irregularities in language-model outputs. Representative approaches include GLTR (Gehrmann et al., 2019), DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2024), and Binoculars (Hans et al., 2024), which analyse token distributions, probability curvature, or cross-model likelihood patterns. These methods are computationally lightweight and avoid costly supervised training, making them attractive for rapid deployment and unseen-generator evaluation. Feature-based approaches such as Ghostbuster (Verma et al., 2024) additionally leverage stylometric and probabilistic signals for cross-domain generalisation. More recent work targets the proxy-model mismatch that arises when the scoring model differs from the unknown generator: SurpMark (Chen and Khisti, 2026) summarises a passage by the dynamics of its token surprisals and scores it against fixed human and machine references, avoiding the per-input contrastive generation that makes curvature-based methods expensive.

Although zero-shot methods avoid costly labeled training data and may generalise to unseen generators, prior work shows substantial degradation under domain shifts, decoding variations, and adversarial rewriting attacks (Chakraborty et al., 2024; Dugan et al., 2024; Sadasivan et al., 2025). These limitations become particularly pronounced under fixed-threshold evaluation where target-domain calibration is unavailable.

2.2 Supervised Detection Methods

Supervised detectors fine-tune pretrained language encoders on labeled human-versus-AI text pairs. Early work leveraged transformer architectures such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) trained on benchmarks including HC3 (Guo et al., 2023) and HC3 Plus (Su et al., 2023). More recent approaches improve robustness through adversarial training, representation learning, and generator-aware adaptation. For example, RADAR (Hu et al., 2023) trains detectors adversarially against paraphrasing attacks. RepreGuard (Chen et al., 2025) instead detects machine-generated text from hidden representation patterns rather than output probabilities, and MGT-Prism (Liu et al., 2026) aligns spectral features to improve generalisation across domains.

Recent studies further show that paraphrasing alone can evade many existing detectors (Krishna et al., 2023), motivating both adversarial training approaches and semantic-invariance augmentation strategies.

Although supervised methods often achieve near-ceiling in-domain accuracy, large-scale benchmarks reveal substantial degradation under cross-generator and cross-domain evaluation (Wang et al., 2024b; Li et al., 2024; Dugan et al., 2024). Many detectors implicitly overfit to generator-specific artefacts, stylistic cues, or decoding signatures that fail to generalise under distribution shift.

2.3 Watermarking-Based Detection

Watermarking approaches embed detectable signals directly during text generation rather than relying on post-hoc analysis (Kirchenbauer et al., 2023). Subsequent work explored probabilistic and cryptographic watermarking strategies for LLM outputs (Wouters, 2024; Zhang et al., 2024). However, watermark robustness degrades substantially under paraphrasing and semantic-preserving rewriting attacks (Zhang et al., 2024).

2.4 Domain Generalisation and Adversarial Robustness

Several recent works explicitly target robustness under domain shift and adversarial perturbations. EAGLE (Bhattacharjee et al., 2024) employs domain-adversarial and contrastive learning objectives to encourage generator-invariant representations, while MGT-Prism (Liu et al., 2026) explores spectral-alignment techniques for improving robustness across domains and generators.

Recent benchmark efforts further highlight the importance of deployment-realistic evaluation. DetectRL (Wu et al., 2024) showed that many detectors degrade substantially under natural distribution shifts and adversarial conditions. Similarly, RAID (Dugan et al., 2024) established a large-scale adversarial robustness benchmark covering paraphrasing, homoglyph attacks, whitespace perturbations, zero-width Unicode characters, and structural manipulations. Most existing approaches primarily address semantic attacks through adversarial training or paraphrase augmentation, while comparatively little attention has been paid to explicit mitigation of surface-form perturbations before tokenisation.

3 Datasets

DeBERTa-ConPara is trained and evaluated on four complementary large-scale benchmarks spanning semantic invariance, cross-domain generalisation, fluent multi-domain writing, and adversarial robustness: HC3 Plus (Su et al., 2023), M4 (Wang et al., 2024b), MAGE (Li et al., 2024), and RAID (Dugan et al., 2024). Together they expose the detector to substantial variability across generators, domains, writing styles and attacks.

The training corpus holds 1,549,358 rows, balanced exactly between human and AI-generated text, with 172,826 further rows held out for validation. Table 1 gives its composition; Appendix A.1.2 describes the human-written filler drawn from M4 to achieve that balance. Two subsets, PeerRead and WikiHow, are generated by us rather than drawn from a published benchmark, and they matter out of proportion to their size: the four public benchmarks use generators current in 2023, so these 43,059 rows are the corpus's only exposure to newer models.

Detailed dataset descriptions, split configurations, attack categories, and per-domain statistics are provided in Appendix A.1.

Dataset	Rows	Domains	AI generators
RAID	530,534	8	11
M4 (human filler)	438,305	5	0
MAGE	319,071	7	27
HC3 Plus	148,040	4	1
M4	70,349	5	8
PeerRead†	28,047	1	5
WikiHow†	15,012	1	7
Total	1,549,358		

Table 1: Composition of the 1.55M-row training corpus. RAID contributes 398,834 AI-generated and 131,700 human rows; the M4 human filler carries no AI text. Domain and generator counts characterise each benchmark, not the sampled subset. A further 172,826 rows are held out for validation. † Generated by us: PeerRead with ChatGPT, LLaMA and LLaMA-2-chat at 7B, 13B and 70B; WikiHow with Claude, GPT-4o, Grok, Qwen-14B, LLaMA-3-8B, Mistral-7B and Cohere.

4 Method

We propose **DeBERTa-ConPara**, a deployment-oriented detector combining attack-aware preprocessing with contextual transformer representations.

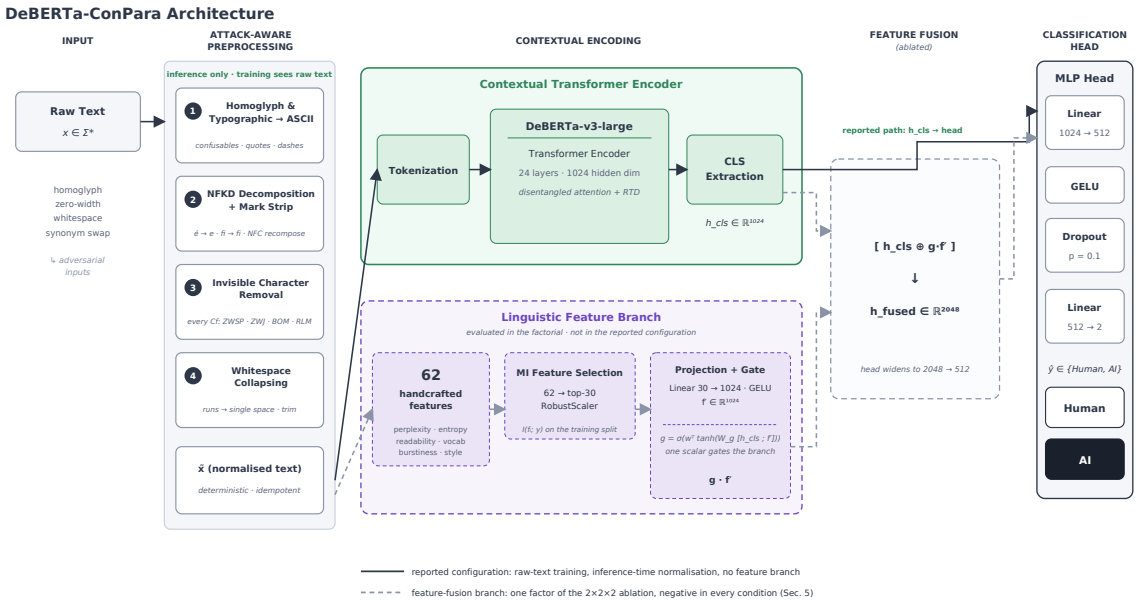


Figure 1: DeBERTa-ConPara architecture. Solid components form the reported configuration: attack-aware preprocessing applied at inference only, a DeBERTa-v3-large encoder, and a two-layer head on the classification embedding. Dashed components are the linguistic feature branch and its gated fusion, one factor of the $2 \times 2 \times 2$ ablation in Section 5, which is negative in every condition and is not part of the reported model.

We also describe a linguistic feature-fusion branch and semantic-invariance augmentation (ConPara), both central to the submitted version and both reported as negative results in Section 5. The configuration we report trains on raw text, normalises at inference, and carries no feature branch.

4.1 Overall Architecture

Figure 1 presents an overview of DeBERTa-ConPara. The input text, normalised at inference by the pipeline of Section 4.2, is encoded by a contextual transformer whose classification embedding feeds the head; the linguistic feature branch shown dashed is the ablated component described next.

Contextual Transformer Encoder. We employ `microsoft/deberta-v3-large` (He et al., 2023) as the contextual encoder backbone. The submitted version used `deberta-v3-base`, appropriate for the smaller corpus then available. With the corpus extended to 1.55M rows we re-ran the comparison with every other factor held fixed (identical data, hyperparameters and seed, five epochs), and the larger backbone is consistently better: balanced accuracy at the optimal threshold rises from $94.56\,\%$ to $96.58\,\%$ and AUROC from 0.98655 to 0.99328. We therefore report the larger backbone throughout; backbone change and corpus extension were adopted together. DeBERTa-v3 combines disentangled attention mechanisms with ELECTRA-style replaced-token detection (RTD) pretraining (Clark et al., 2020), providing strong contextual representations and robust transfer performance across diverse language understanding tasks.

The final hidden-state representation corresponding to the special classification token is extracted as a 1024-dimensional contextual embedding, denoted $\mathbf{h}_{\text{cls}}$ below; in the reported configuration it is the only input to the classifier.

Linguistic Feature Encoder. In parallel, we extract 62 handcrafted linguistic and statistical features designed to capture stylometric properties that may be weakly represented in transformer embeddings. The feature categories include probabilistic, stylometric, readability, and lexical-diversity signals.

To reduce redundancy and improve robustness under distribution shift, we select the top $k = 30$ features using Mutual Information (Cover and Thomas, 2006) on the training set; the full feature list and the selection procedure are in Appendix A.2.2.

Selected features are normalised using RobustScaler, projected to the encoder width, and combined with the contextual representation through a learned gate.

Formally, given the selected feature vector $\mathbf{f}(x) \in \mathbb{R}^{30}$ and the encoder's classification embedding $\mathbf{h}_{\text{cls}} \in \mathbb{R}^{1024}$, the branch computes

$$\begin{aligned}
\mathbf{f}' &= \text{Dropout}\big(\phi(W_p\,\mathbf{f}(x))\big), \quad W_p \in \mathbb{R}^{1024\times 30}, \\
g &= \sigma\Big(\mathbf{w}^\top \tanh\big(W_g\,[\mathbf{h}_{\text{cls}}; \mathbf{f}']\big)\Big), \\
\mathbf{h} &= \big[\,\mathbf{h}_{\text{cls}}\;;\; g\,\mathbf{f}'\,\big] \in \mathbb{R}^{2048},
\end{aligned} \tag{1}$$

where ϕ is GELU and σ the logistic function. Note that g is a *single scalar*: the gate admits or suppresses the projected feature vector as a whole rather than reweighting its coordinates. The gated features are concatenated with the 1024-dimensional DeBERTa representation, forming a **2048-dimensional fused representation** passed to the classifier.

Binary Classification Head. The fused representation is processed using a lightweight two-layer MLP:

$$\begin{aligned}
&2048 \to 512 \to \text{GELU} \\
&\quad \to \text{Dropout}(0.1) \to 2.
\end{aligned} \tag{2}$$

In the configuration we report, which carries no feature branch, the classification embedding is passed to the same head directly, giving $1024 \to 512 \to \text{GELU} \to \text{Dropout}(0.1) \to 2$. The released checkpoint implements exactly this head.

4.2 Attack-Aware Preprocessing Pipeline

We introduce an attack-aware preprocessing pipeline applied prior to tokenisation to improve robustness against Unicode-based and formatting-based adversarial perturbations such as homoglyph substitutions and zero-width character insertions.

The pipeline applies, in order: targeted homoglyph substitution using a curated confusables table spanning Cyrillic, Greek, fullwidth and stylised Latin variants; typographic normalisation of smart quotes, dashes and fractions to ASCII; NFKD decomposition with combining-mark removal, recomposed with NFC; removal of every Unicode format character (category Cf), which covers zero-width and directional marks; and whitespace collapsing. Homoglyphs are handled explicitly because they are canonically distinct and decomposition alone does not repair them. Full stage descriptions are given in Table 8.

The pipeline is deterministic and idempotent, with negligible overhead.

Where it is applied, however, matters more than whether it is applied: Section 5.2 shows that normalising the training corpus removes the adversarial supervision that corpus provides, whereas normalising at inference is an effective defence. The configuration we report therefore trains on raw text and normalises only at inference, in contrast to the submitted version, which applied the pipeline at both stages.

4.3 Semantic-Invariance Augmentation

To improve robustness against paraphrasing and semantic-preserving rewriting attacks, we investigate semantic-invariance augmentation through paraphrase generation and supervised contrastive learning.

Paraphrase Augmentation. For selected training samples, paraphrases are generated using `flan-t5-base` (Chung et al., 2024). Generated paraphrases are filtered using a cosine-similarity threshold of ≥ 0.85, computed over sentence embeddings from a pretrained encoder (Krishna et al., 2023). This threshold empirically retains paraphrases that preserve the original semantic intent while introducing sufficient lexical and syntactic variation; lower thresholds admitted meaning-altering rewrites, while higher thresholds eliminated useful diversity in preliminary experiments.

Supervised Contrastive Learning. We additionally experiment with supervised contrastive learning (SupCon) (Khosla et al., 2020) using paraphrased positive pairs and opposite-class negatives. The training objective combines weighted cross-entropy with SupCon loss weighted by $\lambda \in \{0.05, 0.1, 0.2\}$, selected via ablation (Equation 3, Appendix A.2.3).

4.4 Training Configuration

Training uses AdamW (Loshchilov and Hutter, 2019) with weight decay 0.01, peak learning rate 2×10^{-5}, batch size 32 and maximum sequence length 512, under a OneCycleLR schedule with cosine annealing (Smith and Topin, 2019); the reported checkpoint is the epoch with the best validation balanced accuracy. Full hyperparameters are listed in Appendix A.2.4. The development sequence that preceded the factorial, from Cohere-focused oversampling through adversarial data scaling to a generator-difficulty weighted loss, is recorded in Appendix E; the controlled comparisons of Section 5 supersede it.

5 Experiments and Results

We evaluate DeBERTa-ConPara under a deployment-realistic fixed-threshold protocol. Following the official RAID hidden-test evaluation procedure (Dugan et al., 2024), we report TPR@5 % FPR, TPR@1 % FPR, and AUROC. In addition to RAID leaderboard evaluation, we assess cross-dataset generalisation on HC3 Plus and MAGE, perform external academic-domain validation, analyse multi-seed stability, and compare against strong zero-shot and supervised baselines.

5.1 Evaluation Protocol

We adopt a **fixed-threshold evaluation protocol** designed to better reflect realistic deployment conditions. A single decision threshold τ^* is calibrated on source validation data by maximising balanced accuracy (Youden, 1950) and is then held constant across all target distributions.

Unlike standard evaluation settings that independently re-tune thresholds for each target dataset, the fixed-threshold setting exposes operating-point trade-offs and generator-specific failure modes that emerge when target-domain labels are unavailable. This setup more closely reflects practical deployment scenarios encountered in moderation systems, educational platforms, and misinformation-filtering pipelines.

For RAID, we follow the official hidden-test leaderboard protocol (Dugan et al., 2024), where thresholds are calibrated to achieve exactly 5 % false positive rate (FPR) on human-written text within each domain. Performance is evaluated using TPR@5 % FPR (primary metric), TPR@1 % FPR, and AUROC (Fawcett, 2006).

5.2 Ablation Study: Preprocessing at Training Time and at Inference Time

Preprocessing can be applied when the training corpus is built, when a test document is scored, or both. The two are distinct interventions that, as we show below, act in opposite directions, so we vary them independently.

Normalisation deduplicates the training corpus. RAID contains 13,371 unique human articles under twelve attack variants. Three attacks, homoglyph substitution, zero-width-space insertion and whitespace manipulation, are pure character substitutions, so Unicode normalisation returns the original text exactly and the attacked row becomes a byte-identical duplicate that deduplication discards. On a matched 1,892-row sample, the normalised pipeline retains 1,223 distinct texts against 1,892 for the raw one: normalisation collapses 35.4 % of RAID and removes all three attack classes from the corpus. Raw-text training is therefore implicitly adversarial; normalising at training time deletes that supervision.

Ft	Tr	Inf	AUR	T5	T1	$T1_{cl}$	pen
✓	n	n	99.48	98.84	87.45	88.83	1.38
✓	n	r	97.32	92.76	75.59	88.38	12.79
✓	r	r	99.39	98.83	93.64	95.77	2.13
✓	r	n	99.51	99.02	94.72	95.28	0.56
✗	n	n	99.54	98.50	94.52	95.44	0.92
✗	n	r	95.77	86.14	79.48	95.28	15.80
✗	r	r†	99.35	98.39	93.78	97.17	3.39
✗	**r**	**n**	99.61	99.01	96.57	96.98	**0.41**

Table 2: The complete $2\times2\times2$ factorial on the 672,000-row RAID hidden test. Ft = feature-fusion branch; Tr/Inf = normalisation at training and inference (n/r); AUR = AUROC; T5/T1 = TPR at 5 %/1 % FPR; $T1_{cl}$ = TPR@1 % on unattacked rows; pen = attack penalty $T1_{cl}-T1$. Shading is a fixed scale shared with Appendix G. The last row is the configuration we report. † This cell is a vanilla fine-tuned DeBERTa-v3-large, with neither attack-aware preprocessing nor handcrafted features; it is the supervised baseline requested during review.

An eight-cell factorial. Crossing preprocessing at training time with preprocessing at inference time, with and without the feature-fusion branch, gives eight configurations, trained from the same source corpus with identical hyperparameters and a fixed seed. The training factor is not normalisation alone: because normalisation collapses attacked rows into duplicates, the four *n*-trained cells see roughly 35 % fewer RAID rows than the *r*-trained cells. Tr should therefore be read as normalisation *and* the deduplication it causes. All eight cells were submitted to the RAID hidden test, whose labels we never observe; Table 2 reports them.

The $-$Fr$\rightarrow$r cell is a vanilla fine-tuned DeBERTa-v3-large with neither attack-aware preprocessing nor handcrafted features, and is the supervised baseline requested during review. Every comparison below is therefore on labels we never observe. Three main effects follow, each consistent across the conditions it is averaged over.

Inference-time normalisation always helps, by $+1.08$ to $+15.04$ pp of TPR@1 % FPR. The gain is largest exactly where the model was trained on normalised text and so never met an attack ($+11.86$ and $+15.04$), and small where raw training already supplied that exposure ($+1.08$ and $+2.79$).

Raw-corpus training always helps, by $+2.05$ to $+18.05$ pp, and most when inference is also raw ($+14.30$, $+18.05$), where it is the only defence present.

The feature branch never helps. It costs 7.07, 3.89, 1.85 and 0.14 pp in the four conditions: negative in all of them, and most expensive in the fully normalised setting the submitted version used. That aggregate conceals a far larger domain-specific effect, which we return to below.

In the attack-penalty column, six of eight cells lose under 3.4 pp; the two that lose 12.79 and 15.80 pp are precisely the two trained on normalised text and evaluated on raw. Attack robustness is therefore not a property of the model but of the agreement between its training corpus and its inference pipeline.

The feature branch fails on one domain. Per-domain results (Appendix G, Table 20) show that the branch's cost is not spread evenly. On seven of eight RAID domains it changes TPR@1 % FPR by at most a few points, and on abstracts it is slightly beneficial. On *poetry* it costs 49.8 pp: 38.1 % with the branch against 87.9 % without, in the otherwise identical fully normalised configuration. The penalty falls to 7.8 pp once training uses raw text, so adversarial supervision partly compensates for it, but poetry remains the domain where the branch is most damaging in every condition.

The effect is confined to two attack classes. For ten of the twelve RAID categories the *penalty relative to unattacked text* is within roughly a point across every configuration; the aggregate differences between cells are driven by homoglyph substitution and zero-width-space insertion. Absolute rates still vary by attack, with paraphrase ranging from 75.9 to 90.4 %, but that variation is common to all eight cells. Without inference-time normalisation a normalised-trained model detects 11.05 % of homoglyph and 1.12 % of zero-width-space attacks at 1 % FPR; with it, both reach 96.98 %, identical to its unattacked score. Normalisation does not reduce these attacks, it eliminates them.

Neither placement alone suffices: raw training lifts homoglyph detection to 85.78 % and zero-width to 73.80 %; inference-time normalisation carries both to 96.98 %.

Backbone	AUROC	T5	T1	$T1_{cl}$	pen
DeBERTa-v3-large	99.61	99.01	96.57	96.98	0.41
BERT-large	99.07	96.77	90.54	92.31	1.77
RoBERTa-large	94.32	87.08	71.23	69.68	−1.55

Table 3: Three backbones, identical corpus, hyperparameters and seed, all raw-trained with normalised inference. T5 and T1 are TPR at 5 % and 1 % FPR over all rows; $T1_{cl}$ is TPR@1 % FPR over unattacked rows only; pen $= T1_{cl} - T1$ is the attack penalty, i.e. how much adversarial perturbation costs. RoBERTa is the only system whose penalty is negative, scoring *worse* on unattacked text than overall, which follows from its collapse on one domain rather than from any attack.

The mismatch penalty is also asymmetric, and the same two-attack signature reproduces in Fast-DetectGPT (Bao et al., 2024), which has a different tokeniser and no trained head (Appendix B).

The feature-fusion branch. Four interventions applied to one checkpoint on an identical $20{,}000$-row sample isolate the 30-feature branch; zeroing the projected representation and forcing the gate to zero are mathematically equivalent and produce bit-identical margins, confirming they act where intended. Removing the branch shifts the logit margin by 1.751 on average, yet AUROC is unchanged to four decimals (0.99326 against 0.99336) and TPR@1 % FPR falls 0.40 pp: saturation absorbs the shift. A model trained *without* the branch scores 92.30 % against 92.31 %, so its training signal is not needed either. Out of distribution it is harmful, costing 23.4 pp of TPR@1 % FPR on SemEval-2024 and 7.7 pp on HC3-SI. Note that zeroing the raw inputs is not equivalent to removing the branch: the projection is affine, so a zero input still emits its bias, worth 0.172 logits.

Relation to the submitted version. This design supersedes the A1/A3 comparison in the submitted paper, whose two arms shared a corpus that had already been normalised at construction time; their difference therefore reflected a train/test mismatch rather than the effect of preprocessing.

5.3 Backbone Comparison

Is the result specific to DeBERTa? We trained `bert-large-cased` and `roberta-large` on the identical raw corpus with identical hyperparameters and seed and submitted all three to the hidden test (Table 3; details in Appendix C).

All three tokenisers fragment homoglyph-substituted words into subword pieces rather than emitting unknown tokens, so the vulnerability is a property of subword tokenisation rather than of one model. DeBERTa leads by six points of TPR@1 % FPR over BERT and by twenty-five over RoBERTa (99.61, 99.07 and 94.32 % AUROC), and the backbone ranking on our validation pool does not survive the hidden test: RoBERTa leads BERT by nine points there and trails it by nineteen on the hidden test.

5.4 Impact of the Training Data Scaling and Validation Design

Scaling to the full RAID corpus. To test whether the gains continue, we trained the best configuration on the complete RAID training split (9.38M balanced rows, roughly six times the adversarial exposure), selecting the checkpoint on validation AUROC, which peaked at the second epoch. Hidden-test performance improves marginally, to 99.63 % AUROC and 97.12 % TPR@1 % FPR, an increase of 0.55 pp over the 1.55M-row configuration. On a leakage-free evaluation restricted to sources absent from both training corpora, however, the same model loses 3.42 pp of TPR@1 % FPR on MAGE (91.99 → 88.57, disjoint 95 % intervals), and its cross-dataset average falls from 93.14 to 92.45 % (Table 4). Sixfold adversarial scaling therefore buys about half a point on the benchmark and costs several points of cross-dataset generalisation, which is the trade-off our multi-dataset design is intended to avoid. We report the 1.55M-row configuration as our primary system for this reason.

Validation composition also matters: an early configuration used a validation split in which MAGE was over-represented relative to training, biasing early stopping and costing three points of HC3-SI. Correcting to proportional validation restored it (Appendix F).

5.5 Cross-Dataset Generalisation

Table 4 evaluates the four no-feature cells on three external corpora under the fixed-threshold protocol, with $\tau^\star$ calibrated once on the source validation split and held constant.

The result is that preprocessing placement, which separates the cells by 12.9 pp on RAID, separates them by at most 0.41 pp here: the cross-dataset averages are 93.35, 93.55, 93.17 and 93.14 %. Inference normalisation in particular is inert: N-nn against N-nr, and N-rr against N-rn, are the same weights differing only in inference, and their AUROC and TPR@1 % FPR agree to four decimal places on every benchmark. This is what the mechanism predicts: HC3 Plus and MAGE contain no Unicode perturbations, so normalisation has nothing to remove. Attack-aware preprocessing is not a general preprocessing benefit but a defence that is exactly inert in its absence.

M4GT-Bench. On M4GT-Bench (Wang et al., 2024a), which was not part of training, the four cells reach 97.67, 94.55, 96.70 and 93.59 % TPR@1 % FPR for $+\mathrm{F}\,\mathrm{n}{\to}\mathrm{n}$, $-\mathrm{F}\,\mathrm{n}{\to}\mathrm{n}$, $+\mathrm{F}\,\mathrm{r}{\to}\mathrm{r}$ and $-\mathrm{F}\,\mathrm{r}{\to}\mathrm{n}$ respectively. This is the one external corpus on which the feature branch helps, by about three points in both training conditions, the opposite sign to its effect on SemEval-2024 and HC3-SI, and a reminder that the branch is not uniformly harmful but unpredictable.

Raw-corpus training is close to neutral rather than free: it costs about two points on HC3-SI and one on M4 while gaining one on MAGE, for a net change within half a point on the average. Set against a 17.1 pp gain in RAID TPR@1 % FPR, we regard that as an acceptable exchange, and we report it rather than the average alone.

What freezing the operating point costs. Because $\tau^\star$ is fixed before any target data is seen, the relevant question is what that constraint costs. Sweeping the threshold on each benchmark and comparing balanced accuracy at the frozen $\tau^\star$ against the best achievable on that target gives an oracle gap for each of the 24 cell-benchmark pairs (Figure 2).

The cost is negligible almost everywhere and concentrated on one split. Mean gaps are 0.11 pp on HC3-QA, 0.25 on M4 and 0.35 on MAGE; on HC3-SI the mean is 4.15 pp and the worst case 13.58. Excluding HC3-SI, all 18 remaining pairs fall between 0.02 and 0.99 pp. Fixed-threshold evaluation is therefore close to free on the benchmarks where scores are high, and expensive exactly where meaning-preserving rewriting has already compressed the score distribution. Details, including the released configuration's own gaps, are in Appendix D.

Further out-of-distribution evaluation (Appendix A.3.1) and the development sequence behind the submitted version (Appendix E) are consistent with these findings.

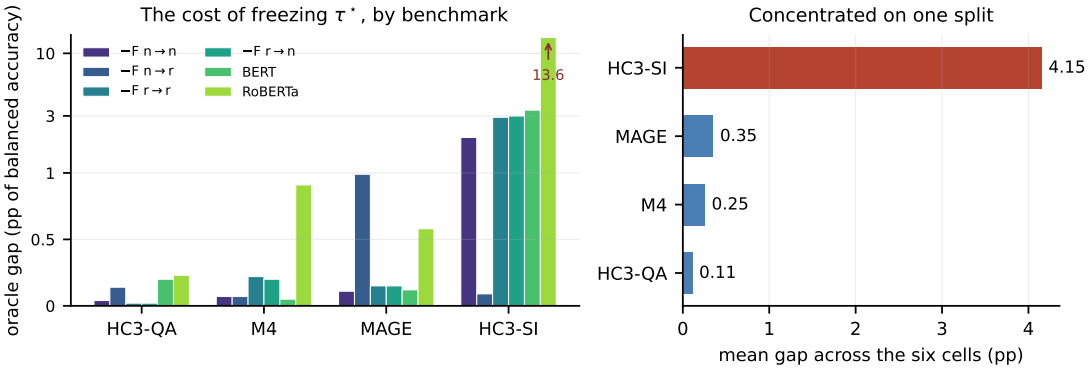


Figure 2: Oracle gap, the balanced accuracy given up by fixing $\tau^\star$ in advance rather than tuning per target, across 24 cell-benchmark pairs. The cost is under one point everywhere except HC3-SI. Per-cell sweep curves are in Appendix D (Figure 3).

Method	HC3-QA	HC3-SI	MAGE	Avg	RAID
RAID leaderboard systems with public checkpoints					
MELD	94.64	66.86	96.17	85.89	99.78
ModernBERT*	95.40	52.54	93.51	80.48	94.14
Desklib v1.01	97.87	56.38	83.44	79.23	91.17
SuperAnnotate	99.04	56.25	60.01	71.77	64.87
e5-small-lora	87.29	59.45	67.12	71.29	85.69
TMR	84.85	55.92	70.99	70.59	95.79
BERT-tiny-4M	69.84	55.60	61.49	62.31	84.18
ADAL	59.89	44.89	61.23	55.34	96.25
RADAR	53.32	49.09	60.40	54.27	63.91
Ours, no-feature cells (DeBERTa-v3-large)					
−F n→n	99.72	84.79	95.55	93.35	98.50
−F n→r	99.67	85.77	95.22	93.55	86.14
−F r→r	99.68	83.64	96.20	93.17	98.39
−F r→n	99.69	83.50	96.23	93.14	99.01
Ours, other backbones (raw/norm)					
BERT-large	98.85	81.39	92.59	90.94	96.77
RoBERTa-large	99.28	72.38	92.70	88.12	87.08
Ours, trained on the full RAID split (Sec. 5.4)					
−F r→n	99.72	83.71	93.94	92.45	99.13

Table 4: Balanced accuracy at the fixed threshold $\tau^\star$ on the external benchmarks, and RAID TPR@5 % FPR from the official leaderboard (Dugan et al., 2024). Avg is the mean over HC3-QA, HC3-SI and MAGE. Competitors are every RAID entry with a public checkpoint, each scored under our protocol. The feature cells were not evaluated externally. * Trained on MAGE, so that column is in-distribution for it.

5.6 Comparison Against Strong Baselines

We compare against every RAID leaderboard system that publishes a checkpoint (Table 4), each downloaded and evaluated under our fixed-threshold protocol rather than at numbers reported elsewhere. MELD is the only open system above ours on RAID alone (99.78 against 99.01); on adversarial robustness in isolation it is the stronger detector. And HC3, MAGE and M4 are training sources for our system, so those columns are held-out splits for us but external data for every competitor, which favours us by construction. What the table does show is that no open system combines both: our configuration leads the cross-dataset average by 7.3 points over MELD, and the systems nearest us on RAID (ADAL, TMR) fall to near chance on HC3-SI. Several commercial systems score above 99 on RAID but publish no checkpoint.

6 Discussion and Conclusion

In-domain accuracy alone is an insufficient indicator of robustness. Among systems with public checkpoints, ours is the only one that is both within a point of the best RAID score and first on the cross-dataset average.

Two findings generalise. Normalisation before tokenisation neutralises zero-width and homoglyph attacks, but only at inference: applying it to the training corpus deletes the adversarial supervision it provides. And robustness here proved data-centric: the feature branch reduced TPR@1 % FPR in every condition, by 49.8 pp on poetry alone.

Both findings are cheap to act on. Inference-time normalisation is a deterministic pre-tokenisation step with no learned parameters, and the same signature appears in a zero-shot detector of different architecture, so existing detectors can adopt it without retraining; it cannot replace adversarial exposure during training.

Code, checkpoint and live demo:
`github.com/MohamedMady19/deberta-conpara`
`huggingface.co/mohamedmady/deberta-conpara`
`huggingface.co/spaces/mohamedmady/deberta-conpara`

Limitations

While DeBERTa-ConPara demonstrates strong robustness across multiple benchmarks and adversarial settings, several limitations remain.

First, each configuration is a single training run. Three seeds of one configuration spanned 10.68 points of hidden-test TPR@1 % FPR. That spread belongs to the metric as much as to the model: TPR@1 % FPR is fixed by one percentile of the human score distribution, so small calibration shifts displace it disproportionately, whereas balanced accuracy on the same system varies by ± 0.75 pp across five seeds (Appendix A.3.3). Our factorial comparisons are paired (identical corpus and seed, one factor changed), so they are not subject to the full between-run spread, but we could not quantify the paired variance without repeating the factorial. We therefore rest our conclusions on effects that are large relative to that spread (raw-corpus training, $+2.05$ to $+18.05$ pp; the poetry penalty, 49.8 pp) or consistent in sign across all four conditions (the feature branch is negative in every one). Differences of a few points between cells are indicative only; where we draw a conclusion in that range, as in the scaling analysis, it rests on the cross-dataset loss (3.42 pp, disjoint 95 % intervals), not on the small hidden-test gain.

Second, all experiments focus exclusively on English-language datasets, and all robustness claims should be read as applying only to the English domains, generators and attacks evaluated here. Extending the framework to multilingual settings remains an important direction, particularly since Unicode-based attack patterns and linguistic structures vary substantially across languages. Multilingual benchmarks such as DetectRL-X (Wu et al., 2026), which spans eight languages and six domains, report that detector performance degrades unevenly across scripts and that short texts are substantially harder in morphologically richer languages than in English; whether attack-aware normalisation transfers to non-Latin scripts is therefore an open question our results do not address.

Third, although attack-aware preprocessing substantially improves robustness against surface-level perturbations, semantically preserving transformations such as sophisticated paraphrasing remain challenging for current detection systems.

Fourth, the human-written training data originates from pre-2022 sources, predating the widespread availability of large-scale generative AI. While this temporal boundary provides provenance guarantees, human writing style is evolving in response to AI use, and AI-assisted composition is emerging as a challenging intermediate category that binary detection frameworks do not address. We consider the development of three-class detection (human, AI-assisted, fully AI-generated) an important direction for future work.

Finally, robustness evaluation is necessarily bounded by currently available benchmarks and attack strategies. AI text generation is advancing rapidly, including the emergence of post-processing humanisation filters that deliberately reduce the statistical detectability of AI-generated text (Chakraborty et al., 2024). A particularly challenging emerging scenario involves personally adapted AI systems that deliberately imitate an individual's writing style including their characteristic errors, vocabulary, and syntactic patterns making stylometric detection ineffective by design. Addressing such threats will require author-adaptive detection approaches and continued benchmark refreshment as both human writing and AI generation capabilities evolve.

Ethical Considerations

Reliable AI-generated text detection can support important applications including academic integrity, misinformation mitigation, content moderation, and quality control in human-annotated dataset construction.

Detection systems carry false-positive risks, particularly for non-native speakers, minority dialects and stylistically atypical human writing, while false negatives cause harm in academic-integrity and misinformation settings. We therefore recommend that detectors serve as decision support rather than definitive evidence in high-stakes settings, accompanied by human oversight, transparent confidence reporting and clear appeal mechanisms.

The released system is intended for defensive detection research rather than surveillance or censorship applications.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, and 1 others. 2023. GPT-4 technical report. arXiv preprint arXiv:2303.08774. *Preprint*, arXiv:2303.08774.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *12th International Conference on Learning Representations*, Vienna, Austria.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. EAGLE: A domain generalization framework for AI-generated text detection. arXiv preprint arXiv:2403.15690. *Preprint*, arXiv:2403.15690.

Souradip Chakraborty, Amrit Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. 2024. Position: On the possibilities of AI-generated text detection. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6093–6115, Vienna, Austria. PMLR.

Shuangyi Chen and Ashish J. Khisti. 2026. Black-box detection of LLM-generated text using generalized Jensen Shannon divergence. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 14913–14955, Seoul, South Korea. PMLR.

Xin Chen, Junchao Wu, Shu Yang, Runzhe Zhan, Zeyu Wu, Ziyang Luo, Di Wang, Min Yang, Lidia S. Chao, and Derek F. Wong. 2025. RepreGuard: Detecting LLM-generated text by revealing hidden representation patterns. *Transactions of the Association for Computational Linguistics*, 13:1812–1831.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations*, Addis Ababa, Ethiopia.

Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory*, 2nd edition. John Wiley & Sons, Hoboken, New Jersey, USA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Tom Fawcett. 2006. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116, Florence, Italy. Association for Computational Linguistics.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models. arXiv preprint arXiv:2312.11805. *Preprint*, arXiv:2312.11805.

Josh A. Goldstein, Girish Sastry, Micah Musser, Renée DiResta, Matthew Gentzel, and Katerina Sedova. 2023. Generative language models and automated influence operations: Emerging threats and potential mitigations. *arXiv preprint arXiv:2301.04246*.

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

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 17519–17537, Vienna, Austria. PMLR.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda.

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

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33*, pages 18661–18673, Virtual. Curran Associates, Inc.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084, Honolulu, Hawaii, USA. PMLR.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems 36*, pages 27469–27500, New Orleans, Louisiana, USA. Curran Associates, Inc.

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

Yupei Li, Manuel Milling, Lucia Specia, and Björn W. Schuller. 2025. Discourse features enhance detection of document-level machine-generated content. In *Proceedings of the 2025 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Rome, Italy. IEEE.

Shengchao Liu, Xiaoming Liu, Chengzhengxu Li, Zhaohan Zhang, Guoxin Ma, Yu Lan, and Shuai Xiao. 2026. MGT-Prism: Enhancing domain generalization for machine-generated text detection via spectral alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32132–32140.

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations*, New Orleans, Louisiana, USA.

Mohamed Mady. 2026a. Academic-Text-arxiv-gpt-gemini. Hugging Face Datasets.

Mohamed Mady. 2026b. HC3-Gemini-Flash-Responses. Hugging Face Datasets.

Mohamed Mady, Yupei Li, Johannes Reschke, and Björn W. Schuller. 2026. Feature-augmented transformers for robust AI-text detection across domains and generators. *Preprint*, arXiv:2605.03969.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962, Honolulu, Hawaii, USA. PMLR.

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

Leslie N. Smith and Nicholay Topin. 2019. Super-convergence: Very fast training of neural networks using large learning rates. *Preprint*, arXiv:1708.07120. Also published in Proc. SPIE 11006, Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, 2019.

Stanford Institute for Human-Centered Artificial Intelligence (HAI). 2025. The 2025 AI index report. https://hai.stanford.edu/ai-index/2025-ai-index-report. Accessed: 2025-05-01.

Chris Stokel-Walker. 2022. AI bot ChatGPT writes smart essays — should professors worry? *Nature*. News feature, 9 December 2022.

Zhenpeng Su, Xing Wu, Wei Zhou, Guangyuan Ma, and Songlin Hu. 2023. HC3 plus: A semantic-invariant human ChatGPT comparison corpus. arXiv preprint arXiv:2309.02731. *Preprint*, arXiv:2309.02731.

Turnitin LLC. 2024. 2024 turnitin wrapped. https://www.turnitin.com/blog/2024-turnitin-wrapped. Accessed: 2025-05-01.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1702–1717, Mexico City, Mexico. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024a. M4GT-Bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3964–3992, Bangkok, Thailand. Association for Computational Linguistics.

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

Bram Wouters. 2024. Optimizing watermarks for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 53251–53269, Vienna, Austria. PMLR.

Junchao Wu, Yefeng Liu, Chenyu Zhu, Hao Zhang, Zeyu Wu, Tianqi Shi, Yichao Du, Longyue Wang, Weihua Luo, Jinsong Su, and Derek F. Wong. 2026. DetectRL-X: Towards reliable multilingual and real-world LLM-generated text detection. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 38247–38294, San Diego, California, USA. Association for Computational Linguistics.

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

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. 2024. DetectRL: Benchmarking LLM-generated text detection in real-world scenarios. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, pages 100369–100401, Vancouver, BC, Canada. Curran Associates, Inc.

W. J. Youden. 1950. Index for rating diagnostic tests. *Cancer*, 3(1):32–35.

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. 2024. Watermarks in the sand: Impossibility of strong watermarking for language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 58851–58880, Vienna, Austria. PMLR.

A Appendix

This appendix provides additional details on dataset composition, benchmark characteristics, preprocessing, feature engineering, and robustness evaluation settings used throughout this work. The supplementary material is intended to support reproducibility and provide deeper insight into the deployment-oriented robustness properties of DeBERTa-ConPara.

A.1 Detailed Dataset and Benchmarks Descriptions

DeBERTa-ConPara is trained and evaluated on four complementary large-scale datasets spanning semantic invariance, cross-domain generalisation, highly fluent writing, and adversarial robustness settings.

A.1.1 HC3 and HC3 Plus

The **HC3** benchmark (Guo et al., 2023) (Human ChatGPT Comparison Corpus) is one of the earliest large-scale datasets for AI-generated text detection. It contains approximately 40,000 questions and responses from human experts collected through social media and Wikipedia, with ChatGPT (GPT-3.5 series) used to generate corresponding answers. HC3 is available in both English and Chinese, with the English subset covering five domains and the Chinese subset covering seven domains, spanning areas including open-domain question answering, medicine, finance, and law.

HC3 Plus (Su et al., 2023) extends HC3 with semantic-invariant tasks beyond question answering. The semantic-invariant split incorporates summarisation, translation, and paraphrasing tasks, with human-written text sourced from CNN/DailyMail, XSum, LCSTS, the CLUE benchmark, and datasets from the Workshop on Machine Translation (WMT). AI-generated outputs were produced using GPT-3.5-Turbo for all tasks in the semantic-invariant split. Unlike the original HC3 question-answering tasks, semantic-invariant tasks are substantially more challenging for detection because the AI output preserves the full semantic content of the human source, eliminating the stylistic differences that make standard QA detection tractable.

We use only the English subset throughout this work. The official HC3 Plus training split contains 148,040 samples, while the SI split is reserved exclusively for semantic robustness evaluation.

A.1.2 M4

M4 (Wang et al., 2024b) (Multi-Generator, Multi-Domain, and Multi-Lingual Black-Box Machine-Generated Text Detection) is designed specifically for evaluating cross-generator and cross-domain robustness, and received the Best Resource Paper Award at EACL 2024.

The full M4 corpus spans **seven languages** (Arabic, Bulgarian, Chinese, English, Indonesian, Russian, and Urdu) and contains approximately 147,000 human-machine parallel text pairs, of which 102,000 are English and 45,000 cover the remaining six languages. In addition, the corpus contains over 10 million non-parallel human-written texts. AI-generated outputs are produced by six generators: GPT-4, ChatGPT, GPT-3.5 (`text-davinci-003`), Cohere, Dolly-v2, and BLOOMz 176B. Human-written English texts are sourced from Wikipedia (March 2022 snapshot), WikiHow, Reddit ELI5, arXiv, and PeerRead.

We use the **English subset only**, covering five domains (Wikipedia, WikiHow, Reddit ELI5, PeerRead, and arXiv) and eight AI generators (ChatGPT, text-davinci-003, Cohere, BLOOMZ, Dolly, Dolly-2, Flan-T5, and LLaMA), as detailed in Table 6. After preprocessing, filtering, and deduplication, the English subset contains 78,766 samples (12,583 human-written and 66,183 AI-generated). Because the English subset is heavily AI-dominated (approximately 84 % AI-generated), we rebalance the training corpus by supplementing its human-written side with additional human-written texts drawn exclusively from the Wikipedia (March 2022 snapshot) and WikiHow subsets of M4. These sources predate the public release of ChatGPT (November 2022) and are therefore verifiably human-authored under a temporal guarantee. This filler contributes 438,305 rows, the single largest component of the 1.55M-row corpus, and is reported as its own line in Table 1 rather than folded into the M4 total. Full split statistics are provided in Table 6.

A.1.3 MAGE

MAGE (Li et al., 2024) (MAchine-GEnerated text detection in the Wild) is a large-scale benchmark designed to evaluate AI-generated text detection under realistic, highly heterogeneous conditions resembling practical deployment scenarios.

The benchmark collects human-written texts from **seven distinct writing tasks**, including story generation, news writing, scientific abstract

Split	Subset	Total	Human	AI	Purpose
Train	HC3 Plus QA	148,040	83,261	64,779	Main training set
Validation	HC3 QA	5,839	2,919	2,920	Standard validation
Validation	HC3 SI	5,839	2,919	2,920	Semantic-invariant validation
Test	HC3 QA	24,969	16,951	8,018	Standard evaluation
Test	HC3 SI	38,110	19,047	19,063	Semantic-invariant evaluation

Table 5: Detailed composition of the English splits of HC3 Plus. All counts denote number of text samples. The HC3 Plus QA test split is human-heavy following the official benchmark configuration.

Domain	Human	ChatGPT	Davinci003	Cohere	BLOOMZ	Dolly	Dolly2	FlanT5	LLaMA
Wikipedia	3,000	2,995	3,000	2,336	3,000	2,702			
Reddit ELI5	3,000	3,000	3,000	3,000	3,000			4,220	
WikiHow	3,001	3,000	3,000	3,000	3,000		3,000		
PeerRead	586	586	586	586		586			586
arXiv	2,996	3,000	3,000	1,220	3,000	3,000		1,780	
Total	**12,583**	**12,581**	**12,586**	**10,142**	**12,000**	**6,288**	**3,000**	**6,000**	**586**

Table 6: Sample counts for the English subset of M4 used in this work, covering five domains and eight AI generators. A blank cell means M4 provides no text for that generator in that domain.

writing, peer review, and biomedical question answering, among others. For each human-written text, corresponding machine-generated texts are produced using **27 large language models** under three representative prompt types (continuation, topical, and specified), covering a wide spectrum of model families including OpenAI GPT (text-davinci-002/003, GPT-3.5-Turbo), LLaMA (6B/13B/30B/65B), GLM-130B, FLAN-T5 (small/base/large/xl/xxl), OPT (125M to 30B), and BigScience models (T0, BLOOM-7B1) (Li et al., 2024).

The data is organised into **eight testbeds** of progressively increasing detection complexity, ranging from in-domain detection of a single white-box generator to the most challenging setting involving texts from entirely unseen domains and generators under paraphrasing attacks. The full corpus contains 447,674 samples. Following the official split configuration, we use 319,071 training samples.

Compared with HC3 Plus and M4, MAGE exhibits substantially weaker stylistic separation between human and AI text, particularly in formal and polished writing domains. This is partly because the dataset explicitly captures the "in-the-wild" regime where surface-level lexical cues are insufficient for reliable detection. This makes MAGE especially challenging under deployment-realistic fixed-threshold evaluation.

A.1.4 RAID

RAID (Dugan et al., 2024) (Robust AI-Text Detection Benchmark) is the primary benchmark used in this work for adversarial robustness evaluation, and the largest machine-generated text detection benchmark available at the time of writing.

Corpus Structure. The full RAID corpus contains over **6.2 million AI-generated texts**, constructed by generating 2,000 continuations for every combination of domain, model, decoding strategy, repetition penalty, and adversarial attack. Human-written reference texts are sourced from publicly available pre-2022 datasets across the same eight domains.

Generators (11): ChatGPT, GPT-4, GPT-3 (`text-davinci-003`), GPT-2 XL, LLaMA-2-70B (Chat), Cohere, Cohere (Chat), MPT-30B, MPT-30B (Chat), Mistral-7B, and Mistral-7B (Chat).

Domains (8): arXiv abstracts, book summaries, news articles (NYT), poetry, recipes, Reddit posts, IMDb movie reviews, and Wikipedia.

Adversarial attacks (11): paraphrasing, synonym substitution, homoglyph replacement, zero-width character insertion, whitespace addition, upper/lower case swap, article deletion, number swap, alternative spelling, perplexity-based misspelling, and paragraph insertion.

Decoding strategies (4): greedy (T=0), sampling (T=1), greedy with repetition penalty (T=0, Θ=1.2), and sampling with repetition penalty (T=1, Θ=1.2). Repetition penalty variants are

Attack Type	Example (Attacked)	After Preprocessing	Effectiveness
Homoglyph substitution	Thıs ıs ã test	This is a test	Fully neutralised
Zero-width insertion	This[ZWS]is[ZWS]a test	This is a test	Fully neutralised
Whitespace perturbation	This is a test	This is a test	Fully neutralised
Case perturbation	tHiS iS a TeSt	this is a test	Mostly neutralised
Number/letter swap	Th1s 1s a test	This is a test	Partially neutralised
Alternative spelling	colour → color	normalised	Partial
Synonym substitution	test → examination	semantic change remains	Limited
Paraphrasing	rewritten sentence	semantic change remains	Limited
Article deletion	sentence without articles	unchanged	Limited
Misspelling	artificail inteligence	partially corrected	Partial
Paragraph insertion	sentence + inserted para.	unchanged	Limited

Table 7: Representative RAID attacks and the effect of the proposed preprocessing pipeline. Surface-form attacks are largely neutralised before tokenisation, while semantic attacks remain challenging.

applied to open-source models only.

Train/Test split. Ten per cent of the RAID corpus is withheld without labels as the **official hidden test set**, used exclusively through the public leaderboard at `raid-bench.xyz`. The remaining 90 % is released with labels and used for training. All leaderboard results are evaluated on the official hidden test set.

RAID rows used in this work. The training corpus of Table 1 draws 398,834 AI-generated texts from the labelled RAID split, sampled across the attack × domain cells, together with all 131,700 human reference texts, 530,534 rows in all. The submitted version used smaller subsets of the same construction with a fixed number of AI texts per cell and Cohere-focused oversampling (v2.2: ∼205k rows, 1,671 per cell; v2.4-3k: ∼322k, 3,000; v2.4-5k: ∼498k, 5,000; Appendix E). The full-split run of Section 5.4 uses every labelled RAID training row (9.38M balanced rows).

Data currency note. The RAID benchmark represents the state of adversarial AI-text detection as of its 2024 release, covering 11 generators and 11 attack categories available at that time. AI text generation capabilities are advancing rapidly, including the emergence of post-processing humanisation filters designed to reduce the statistical detectability of AI-generated text (Chakraborty et al., 2024; Wu et al., 2025). Detectors trained on current benchmarks may therefore face accelerating distribution shift as newer, more human-like generators enter deployment. This is a known open challenge shared across the AI-text detection community, and motivates continued benchmark refreshment alongside the targeted adversarial data curation strategy introduced in this work.

A.2 Additional Method Details

This section provides additional implementation, preprocessing, feature-engineering, and training details to facilitate reproducibility. Additional analyses are included to clarify the behaviour of the proposed attack-aware preprocessing pipeline, feature-fusion mechanism, and semantic-invariance augmentation strategies under adversarial and cross-domain conditions.

A.2.1 Attack Taxonomy and Preprocessing Pipeline

RAID includes 11 adversarial attack categories covering both surface-form perturbations and semantic-preserving rewriting attacks. Many of these attacks exploit vulnerabilities in Unicode handling, tokenisation, whitespace processing, and stylometric feature extraction rather than high-level semantic modelling itself. Table 7 summarises all attack categories and their response to the proposed preprocessing pipeline.

The attack-aware preprocessing pipeline mitigates surface-form perturbations before tokenisation and feature extraction, explicitly targeting Unicode-level and formatting-level manipulations that destabilise tokenisation, entropy estimation, burstiness statistics, and stylometric feature extraction. The pipeline performs six stages applied consistently during both training and inference. Table 8 details each stage; Table 9 provides representative normalisation examples.

The pipeline additionally stabilises linguistic-feature distributions. Without normalisation, Unicode perturbations artificially inflate entropy, burstiness, and vocabulary-richness estimates, leading to unstable detector behaviour under adversarial attacks.

Stage	Purpose
1. Homoglyph substitution	Cyrillic, Greek and fullwidth characters mapped to their Latin look-alikes through a curated confusables table. NFKD cannot repair these, as they are canonically distinct.
2. Typographic normalisation	Smart quotes, dashes, fractions and related symbols mapped to ASCII.
3. NFKD decomposition	Decomposition followed by removal of combining marks (Mn), reducing accented Latin to ASCII (é → e) and expanding compatibility forms; the result is re-composed with NFC.
4. Invisible char. removal	Removal of every Unicode format character (category Cf): zero-width space, ZWNJ, ZWJ, BOM, directional marks and soft hyphen.
5. Whitespace collapsing	Runs of spaces collapsed to one. At inference all whitespace is collapsed and the string trimmed.

Table 8: Stages of the attack-aware preprocessing pipeline, in the order applied. The pipeline is deterministic, idempotent and carries no learned parameters.

Original Adversarial Text	After Preprocessing	Attack Type
[Cyrillic] AI-generated text	AI-generated text	Homoglyph
This[ZWS]is-[ZWS]a test	This is a test	Zero-width
T h i s i s A I	This is AI	Whitespace
Th1s t3xt c0nta1ns subst.	This text...	Char. subst.
tHiS Is An aDvErSaRiAl SaMpLe	this is an adv.	Case pert.

Table 9: Representative adversarial-text normalisation examples performed by the preprocessing pipeline.

A.2.2 Linguistic Features

The handcrafted feature space contains 62 linguistic and statistical features organised into eight categories. The top 30 features are selected for feature-attention fusion via Mutual Information (Cover and Thomas, 2006) computed on source-side training data, as detailed below.

Feature Normalisation and Selection. Feature selection is performed exclusively on source-side training data to avoid target-domain leakage. All features are normalised using `RobustScaler` fitted on source training data only. The final subset (k=30) maximises Mutual Information with the binary human/AI label on the full training mix.

Complete Feature List (62). Table 11 lists all 62 features with descriptions, organised by category.

Category	**N**	**Description**
Perplexity	8	GPT-2 perplexity stats (doc/token)
Entropy	10	Char., word, n-gram, POS entropy
Burstiness	6	Repetition and interval burstiness
Repetition	10	Self-BLEU, unique n-grams, compression
Vocabulary	8	Zipf, Yule, Hapax, TTR, MATTR
Coherence	6	Sentence similarity, topic drift
Readability	6	Flesch, FK-grade, ARI, SMOG
Stylometric	8	Punctuation, pronouns, caps ratio
Total	**62**	
Selected	**30**	Via MI on training data

Table 10: Handcrafted feature categories (m=62 total; k=30 selected via Mutual Information).

A.2.3 Contrastive Learning and Paraphrase Augmentation

To improve robustness against semantic-preserving rewriting, we additionally investigate paraphrase augmentation and supervised contrastive learning (SupCon). Paraphrases are generated offline using `flan-t5-base` (Chung et al., 2024). Generated samples are filtered using a cosine-similarity threshold of ≥ 0.85 over sentence embeddings. Following Krishna et al. (2023), this value was selected to balance semantic fidelity against lexical diversity: thresholds below 0.80 admitted meaning-altering rewrites that degraded classification signal, while thresholds above 0.90 produced near-identical paraphrases with insufficient diversity to improve robustness. For supervised contrastive learning, positive pairs consist of original and paraphrased samples from the same class, while in-batch samples from opposite classes serve as negatives. The SupCon objective is combined with weighted cross-entropy loss:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{SupCon}}, \tag{3}$$

where $\lambda \in \{0.05, 0.1, 0.2\}$ is tuned experimentally.

A.2.4 Training Hyperparameters

All experiments use the same configuration. Hyperparameters are selected and kept fixed to ensure consistent comparison.

Feature	Description
Perplexity (8)	
ppl_text	Document-level GPT-2 perplexity
ppl_mean_token	Mean token-level perplexity
ppl_max_token	Maximum token perplexity
ppl_std_token	Std. of token perplexity
ppl_skew_token	Skewness of token perplexity
ppl_low_ratio	Fraction of low-perplexity tokens
ppl_high_ratio	Fraction of high-perplexity tokens
ppl_trend	Perplexity trend across sentences
Entropy (10)	
ent_char	Character-level Shannon entropy
ent_word	Word-level Shannon entropy
ent_bigram	Bigram entropy
ent_trigram	Trigram entropy
ent_conditional	Conditional word entropy
ent_rate	Entropy rate across document
ent_pos	Part-of-speech tag entropy
ent_punct	Punctuation entropy
ent_sent_len	Sentence-length entropy
ent_word_len	Word-length entropy
Burstiness (6)	
burst_word	Word occurrence burstiness
burst_sent_len	Sentence-length burstiness
burst_punct	Punctuation burstiness
burst_memory	Inter-arrival memory coefficient
burst_fano	Fano factor of word frequency
burst_iat_var	Inter-arrival time variance
Repetition (10)	
rep_self_bleu2	Self-BLEU at bigram level
rep_self_bleu3	Self-BLEU at trigram level
rep_self_bleu4	Self-BLEU at 4-gram level
rep_unique_bigram	Unique bigram ratio
rep_unique_trigram	Unique trigram ratio
rep_bigram_rate	Bigram occurrence rate
rep_trigram_rate	Trigram occurrence rate
rep_max_ngram_freq	Maximum n-gram frequency
rep_compression	Text compression ratio (zlib)
rep_dup_sent	Duplicate sentence ratio

Feature	Description
Vocabulary (8)	
vocab_zipf_coef	Zipf law coefficient
vocab_yules_k	Yule's K richness measure
vocab_hapax	Hapax legomena ratio
vocab_dis	Dis legomena ratio
vocab_heaps	Heaps' law exponent
vocab_ttr	Type-token ratio (TTR)
vocab_mattr	Moving-average TTR (MATTR)
vocab_lexical_density	Content word ratio
Coherence (6)	
coh_sent_sim_mean	Mean adjacent sentence similarity
coh_sent_sim_var	Variance of sentence similarity
coh_topic_drift	Topic shift across sentences
coh_self_sim	Global document self-similarity
coh_semantic_density	Average semantic density
coh_consistency	Cross-sentence consistency score
Readability (6)	
read_flesch	Flesch Reading Ease
read_fk_grade	Flesch-Kincaid grade level
read_ari	Automated Readability Index
read_coleman	Coleman-Liau index
read_smog	SMOG grade
read_dale_chall	Dale-Chall readability score
Stylometric (8)	
style_func_word	Function word ratio
style_pronoun	Pronoun usage ratio
style_conjunction	Conjunction density
style_avg_word_len	Mean word length (characters)
style_avg_sent_len	Mean sentence length (words)
style_sent_len_var	Sentence length variance
style_punct_ratio	Punctuation density
style_cap_ratio	Capitalisation ratio

Table 11: Complete list of all 62 handcrafted features with descriptions, organised by category. Left: Perplexity, Entropy, Burstiness, Repetition (38 features). Right: Vocabulary, Coherence, Readability, Stylometric (28 features). The top k=30 are selected via Mutual Information (Section A.2.2).

Parameter	Value
Backbone	DeBERTa-v3-large
Max sequence length	512
Batch size	32
Optimiser	AdamW
Learning rate	2×10^{-5}
Weight decay	0.01
Scheduler	OneCycleLR + cosine annealing
Dropout	0.1
Gradient clipping	1.0
SupCon temperature	0.07
Hardware	NVIDIA A100-SXM4-40GB
Framework	PyTorch + Transformers

Table 12: Training Hyperparameters.

A.3 Additional Experimental Results

This section reports the out-of-distribution evaluation, the semantic-invariance (ConPara) ablation and the multi-seed check carried out on the submitted version of the system; the version sequence itself is in Appendix E.

A.3.1 Extended Out-of-Distribution Evaluation

To further assess generalisation beyond the training distribution, we evaluate DeBERTa-ConPara v2.4-3k ($\tau^* = 0.920$) on two large-scale benchmarks under the fixed-threshold protocol without any target-domain re-calibration. Table 13 summarises the results.

DeBERTa-ConPara achieves strong generalisation across diverse unseen distributions. On 23K Gemini 2.0 Flash HC3-style responses (Mady, 2026b), the detector achieves 98.64 % recall despite Gemini outputs being entirely absent from training data. On the 669K academic benchmark spanning human arXiv paragraphs and GPT-3.5/Gemini-generated scientific prose (Mady, 2026a), the model achieves 94.69 % balanced accuracy and 99.90 % AUROC, with near-perfect AI recall (99.98 %) and 89.41 % human recall.

A.3.2 ConPara Ablations

To evaluate the contribution of semantic-invariance objectives, we investigate supervised contrastive learning (SupCon) and paraphrasing augmentation under the heterogeneous multi-domain setting used throughout this work.

Supervised contrastive objectives produce only modest gains on semantic-invariant HC3 evaluation (+0.47 to +1.41 pp), while consistently degrading robustness on MAGE and RAID. The strongest HC3-SI performance (85.45 %) is achieved with SupCon ($\lambda = 0.05$), but this configuration simultaneously reduces MAGE performance by more than three percentage points.

Similarly, offline paraphrasing augmentation slightly improves HC3-SI robustness but substantially decreases cross-domain adversarial robustness. The combined v3 configuration (SupCon + paraphrasing) produces the largest overall degradation, reducing RAID TPR@5 % FPR from 97.65 % to 95.88 %.

These results suggest that semantic-invariance objectives partially conflict with the AI-vs-human discrimination objective under heterogeneous multi-generator settings: contrastive clustering of semantically related samples can reduce sensitivity to subtle stylistic and statistical signals that remain informative for adversarial detection.

A.3.3 Multi-Seed Stability

To evaluate robustness against random initialisation and stochastic optimisation effects, we repeat v2.2 training across five random seeds and evaluate on a held-out RAID subset containing 53,738 samples.

Metric	Mean	Std. Dev.
Balanced Accuracy	92.20 %	±0.75 pp
AUROC	94.53 %	±1.20 pp

Table 15: Multi-seed stability results for DeBERTa-ConPara v2.2 on a 53k held-out RAID evaluation subset.

The results demonstrate that the observed robustness improvements are stable across repeated training runs, with relatively low variance despite the highly heterogeneous multi-domain and multi-generator setting.

In particular, the low balanced-accuracy variance (±0.75 pp) suggests that attack-aware preprocessing and adversarial data scaling produce consistent robustness gains that are not strongly dependent on favourable random initialisation. The AUROC variance (±1.20 pp) is slightly higher, consistent with known sensitivity of ranking-based metrics to probability calibration across seeds. Together, these results indicate that the primary robustness gains are attributable to architectural and data-centric design choices rather than fortunate random initialisation.

Dataset	N	BA (%)	AI Rec. (%)	Hum. Rec. (%)	AUROC (%)
Gemini 2.0 Flash HC3 (Mady, 2026b)	23,463	98.64	98.64	n/a	n/a
Academic Text (Mady, 2026a)	669,008	94.69	99.98	89.41	99.90

Table 13: Out-of-distribution evaluation under fixed-threshold protocol ($\tau^* = 0.920$, DeBERTa-ConPara v2.4-3k) without target-domain re-calibration. The Gemini set contains AI-generated text only, so human recall and AUROC are undefined (n/a).

Configuration	HC3-QA	HC3-SI	MAGE	Average	RAID TPR@5 %
v2 (no ConPara)	99.43 %	82.18 %	92.84 %	91.91 %	97.11 %
v2.2 (baseline)	99.56 %	84.04 %	93.15 %	92.25 %	**97.65 %**
+ SupCon ($\lambda = 0.05$)	99.24 %	85.45 %	90.10 %	91.60 %	n/s
+ SupCon ($\lambda = 0.1$)	99.42 %	85.01 %	92.59 %	92.34 %	97.18 %
+ SupCon ($\lambda = 0.2$)	99.32 %	84.97 %	90.78 %	91.69 %	n/s
+ Offline paraphrasing (T5)	99.20 %	84.51 %	90.35 %	91.35 %	n/s
v3 (SupCon + paraphrasing)	99.14 %	84.35 %	89.30 %	90.93 %	95.88 %

Table 14: Ablation analysis of supervised contrastive learning and paraphrasing augmentation. Semantic-invariance objectives moderately improve HC3-SI but consistently reduce MAGE and RAID robustness under heterogeneous adversarial training conditions. n/s: variant not submitted to the RAID hidden test.

B Factorial Ablation: Extended Results

Submission provenance. All eight cells were submitted individually to the RAID hidden test between 16 and 25 August 2026. The naming convention in the leaderboard history is: `v218` denotes a model trained on the normalised corpus and `v218b` one trained on raw text; `nofeat` denotes removal of the feature-fusion branch; and the suffix records the inference-time pipeline. Table 16 maps each submission to its cell.

Submission	Feat	Train	Infer
`v2.18`	yes	norm	norm
`v218-rawinf`	yes	norm	raw
`v2.18b-raw`	yes	raw	raw
`v2.18b-norm`	yes	raw	norm
`v218-nofeat-norm`	no	norm	norm
`v218-nofeat-rawinf`	no	norm	raw
`v218b-nofeat-raw`	no	raw	raw
`v218b-nofeat-norminf`	no	raw	norm

Table 16: Mapping from leaderboard submission to factorial cell.

Interaction between the two placements. The two preprocessing placements are not independent. Averaged over the feature factor, inference-time normalisation is worth $+13.45$ pp after normalised-corpus training but only $+1.94$ pp after raw training; equivalently, raw training is worth $+16.18$ pp under raw inference and $+4.66$ pp under normalised inference.

Each intervention substitutes for the other: whichever is applied first captures most of the available gain, and the second adds comparatively little. What is not viable is applying normalisation at training time only, which removes the adversarial supervision and leaves nothing at inference to compensate.

Clean-text performance. The unattacked column separates attack robustness from baseline quality. On unattacked rows alone the eight cells span 88.38 to $97.17\,\%$ TPR@1 % FPR, and the ordering differs from the aggregate: the best clean score belongs to raw training with raw inference (97.17), not to the configuration we report (96.98). The configuration we report wins on the aggregate because its attack penalty is the smallest of the eight (0.41 pp).

The feature branch under every condition. The branch reduces TPR@1 % FPR in all four train/inference combinations, by 7.07, 3.89, 1.85 and 0.14 pp. Its cost is largest in the fully normalised configuration, the one used in the submitted version of this paper, and shrinks as raw training is introduced, consistent with the branch supplying a signal that adversarial supervision provides more reliably.

Neither half of the defence is sufficient alone. The two components decompose cleanly. For homoglyph attacks, detection rises from $11.05\,\%$ with no defence to $85.78\,\%$ with raw-corpus training alone, and to $96.98\,\%$ when inference-time normalisation is added. For zero-width space the corresponding figures are $1.12\,\%$, $73.80\,\%$ and

96.98 %. Adversarial supervision does most of the work; normalisation closes the remainder. Notably, inference-time normalisation slightly *reduces* performance on nine of the twelve attack classes (by 0.09 to 0.24 pp); it is worth applying only because the two classes it repairs are worth 11 and 23 points.

The mismatch penalty is asymmetric. Withholding preprocessing at test time from a model trained with it costs 0.108 of mean submitted score across the hidden test. Adding preprocessing at test time to a model trained without it costs 0.007, fifteen times less and in the beneficial direction. This asymmetry is the signature of adversarial training: a model that has seen perturbations is undisturbed when they are removed, because the clean text was in its training set too, while a model that has never seen them is badly disturbed when they appear.

Generalisation beyond this architecture. The same two-attack signature appears in a detector with no trained classification head and a different tokeniser. Scoring the identical validation rows with Fast-DetectGPT (Bao et al., 2024) using a GPT-Neo-2.7B scoring model, inference-time normalisation raises AUROC on zero-width-space attacks from 0.604 to 0.875 and on homoglyph attacks from 0.683 to 0.873, while the remaining ten classes move by between -0.025 and $+0.005$. The effect therefore belongs to the attacks and to the tokenisation failures they induce, not to our architecture.

C Backbone Comparison: Extended Results

The two observations below extend the backbone comparison of Section 5.3 (Table 3); they explain why the attack is not specific to DeBERTa and why validation-set ranking is an unreliable guide to the hidden test.

The attack mechanism is architecture-general. All three tokenisers fragment homoglyph-substituted words into subword pieces rather than emitting an unknown-token symbol: measured token inflation on a matched sample is 3.40× for BERT, 4.00× for RoBERTa and 4.12× for DeBERTa, with zero unknown tokens in any of them. The vulnerability that inference-time normalisation repairs is therefore a property of subword tokenisation itself.

Validation ranking does not transfer to the hidden test. On our own RAID validation pool, RoBERTa reaches 85.29 % TPR@1 % FPR against BERT's 76.09 %, a nine-point lead. On the hidden test the ordering reverses and the gap widens in the opposite direction: BERT reaches 90.54 % and RoBERTa 71.23 %. Selecting a backbone on validation would have chosen the worse system by nineteen points.

Two measurements explain the instability. RoBERTa is the most overconfident of the three, with a fitted temperature of 2.946 against 2.204 for DeBERTa and 2.154 for BERT, and correspondingly the widest bootstrap interval on validation TPR@1 % FPR (15.7 points against 2.1 for DeBERTa). Its hidden-test failure is also domain-localised rather than uniform: per-domain AUROC is 0.9913 on news and 0.9996 on recipes but 0.6562 on abstracts, and on books the ranking is sound (0.9820 AUROC) while the strict operating point collapses (35.45 % TPR@1 % FPR against 99.13 % at 5 %). A single global threshold is not viable for a model whose scores are that compressed. We report this as a caution about single-split model selection at low false-positive rates, and as motivation for the fixed-threshold protocol used throughout this paper.

D Threshold Sensitivity

Curve shape does not explain the gap. Section 5.5 and Figure 2 give the oracle gap of every cell-benchmark pair; Figure 3 shows the sweep curves behind it. We tested whether the gap is explained by the shape of the threshold curve, as reported for a different model family (Mady et al., 2026): the width of the band keeping balanced accuracy within one point of its maximum correlates at $r = -0.35$ across all pairs, but the sign is inconsistent within benchmarks (-0.19, $+0.48$, $+0.52$, -0.90 for the four splits), so that account does not carry over here. The benchmark, not the curve, governs the cost.

The configuration we report. For the released configuration the gap is 0.01 pp on HC3-QA, 0.10 pp on MAGE and 0.25 pp on M4, so oracle access to the operating point is worth almost nothing on three of the four benchmarks. HC3-SI is again the exception at 2.97 pp, which localises the cost of the protocol to the split that is hardest in absolute terms.

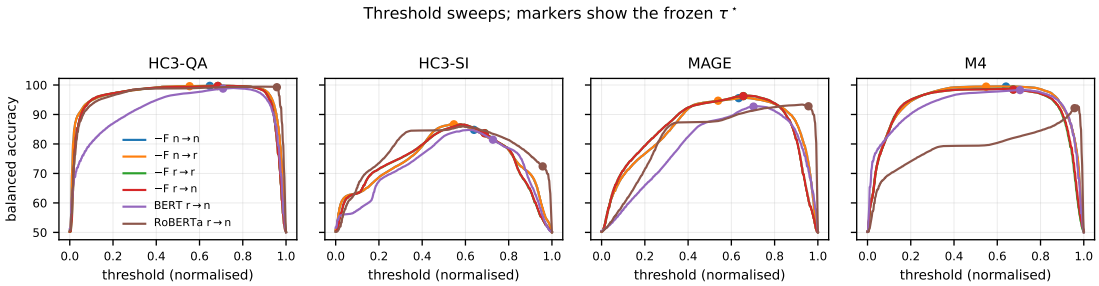


Figure 3: Balanced accuracy as the decision threshold is swept across its range, for each cell on each benchmark. Markers show where the frozen $\tau^\star$ falls. HC3-QA and M4 give broad plateaux, so the frozen threshold lands almost anywhere without cost; HC3-SI is sharply peaked for every cell, which is where the oracle gap comes from.

E Version History

The submitted version of this paper was built on `deberta-v3-base` and went through the development sequence in Table 17, which also lists the full-RAID run of Section 5.4. The sequence is reported for completeness; the factorial of Section 5.2 supersedes it as ablation evidence. Three observations from the sequence still hold.

The gains were data-centric. The largest single step, v1 to v2 (+2.82 pp TPR@5 % FPR), changed preprocessing, corpus composition and adversarial coverage together, and every later step changed data or loss while the backbone stayed fixed. Scaling the RAID training coverage from 1,671 to 3,000 rows per attack × domain cell (v2.2 to v2.4-3k) added +0.14 pp at 5 % and +0.96 pp at 1 % FPR, with the largest gains on Reddit (+1.78 pp) and the Cohere generators (+0.99 to +1.27 pp).

Ranking and operating-point quality diverge. v2.2 has the highest AUROC of the sequence but v2.4-3k the best TPR at both operating points, and the generator-weighted loss of v2.3 raised TPR@5 % FPR slightly while cutting TPR@1 % FPR by 2.35 pp. AUROC alone hides calibration behaviour at strict operating points, which is why the paper reports fixed-threshold metrics throughout. The v2.4-5k row shows the complementary trade-off: proportional validation improved HC3-SI (Appendix F) at the cost of RAID operating-point robustness.

The A1 and A3 arms are not a measure of preprocessing. Both were trained on a corpus normalised at construction time, and A1 was then evaluated on raw text, so their 13 pp gap is a mismatch between training corpus and inference pipeline rather than the contribution of the pipeline. The factorial isolates the same effect: its two cells trained on normalised text and evaluated on raw lose 12.79 and 15.80 pp to adversarial perturbation.

F Validation-Set Composition

Effect of validation-set composition. Our initial v2.4-3k runs used a validation set in which MAGE made up 62.6 % of samples against ∼34 % of the training data; the resulting early-stopping bias cut HC3-SI from 84.04 % (v2.2) to 81.18 %. Correcting to proportional validation restored HC3-SI to **85.33 %** while preserving RAID robustness, so validation composition is a hyperparameter in its own right.

G Factorial Ablation: Complete Breakdowns

Cell labels: first letter, feature-fusion branch (**F** present, **N** absent); second and third, normalisation at training and at inference (**n** normalised, **r** raw). All numbers are TPR at the stated FPR on the 672,000-row RAID hidden test.

Why the feature branch fails on poetry. Plausibly, the 30 features are surface statistics (sentence length, punctuation density, burstiness, type-token ratio) whose distributions poetry violates systematically, so the gated fusion supplies a confidently wrong contribution.

G.1 Per-Cell Detail

Tables 19 to 25 and 22 give all eight cells on every breakdown. Table 18 reproduces, at full precision, only the configuration we report; the corresponding numbers for the other seven cells are the columns of those tables.

Version	TPR@5 %	TPR@1 %	AUROC	Change
Reported (Sec. 5.2)	**99.01 %**	**96.57 %**	**99.61 %**	`deberta-v3-large`, no features, raw training, normalised inference, 1.55M rows
Full RAID (Sec. 5.4)	99.13 %	97.12 %	99.63 %	Same configuration on the complete RAID split (9.38M rows); loses 3.42 pp TPR@1 % FPR on MAGE
v1	94.29 %	87.36 %	**98.58 %**	`deberta-v3-base`, feature fusion
v2	97.11 %	93.34 %	97.38 %	Attack-diverse training data
v2.2	97.65 %	93.99 %	97.87 %	Cohere balancing, revised RAID validation
v2.2-Seed42	97.29 %	93.55 %	96.86 %	v2.2, second seed
v2.3 (GDWL)	97.69 %	91.64 %	97.41 %	Generator-difficulty weighted loss
v2.4-3k	**97.79 %**	**94.95 %**	97.35 %	RAID scaled to 3,000 rows per cell
v2.4-5k	96.70 %	90.98 %	96.01 %	5,000 rows per cell, proportional validation (App. F)
A3 (norm. train, norm. inference)	97.39 %	92.05 %	96.79 %	Submitted-version ablation arm, no features
A1 (norm. train, raw inference)	84.00 %	78.76 %	89.78 %	A3 without inference-time normalisation

Table 17: RAID hidden-test results of the configuration we report, of the same configuration trained on the complete RAID split, and of the development sequence behind the submitted version (all on `deberta-v3-base`). TPR at 5 % and 1 % FPR. Bold marks the configuration we report and, within the development sequence, the best value in each column. A1 and A3 differ only by a train/inference mismatch; the controlled comparison of the two preprocessing placements is the eight-cell factorial of Section 5.2.

Breakdown	**Level**	**TPR@1 %**	**TPR@5 %**
Attack	alternative_spelling	97.13	99.03
	article_deletion	97.87	99.09
	homoglyph	96.98	99.04
	insert_paragraphs	96.98	99.04
	none	96.98	99.04
	number	97.08	99.09
	paraphrase	89.61	98.33
	perplexity_misspelling	97.41	99.03
	synonym	96.78	99.10
	upper_lower	98.10	99.27
	whitespace	96.98	99.04
	zero_width_space	96.98	99.04
Domain	abstracts	96.12	98.59
	books	98.72	99.50
	news	95.62	99.02
	poetry	94.48	98.75
	recipes	99.45	99.84
	reddit	93.45	98.12
	reviews	97.83	99.46
	wiki	96.91	98.83
Generator	chatgpt	99.52	99.72
	cohere	89.83	96.86
	cohere-chat	92.48	98.04
	gpt2	97.62	99.50
	gpt3	94.57	98.29
	gpt4	98.94	99.79
	llama-chat	99.64	99.96
	mistral	93.90	98.32
	mistral-chat	98.74	99.67
	mpt	94.96	98.11
	mpt-chat	98.35	99.71
Decoding	greedy	98.26	99.51
	sampling	94.89	98.52
Rep. penalty	no	96.03	98.88
	yes	97.57	99.25

Table 18: Cell $-\mathbf{F}$ $\mathbf{r}{\to}\mathbf{n}$: without the feature branch, raw training, normalised inference; the configuration we report.

H Per-Cell Performance Matrices

The ten configurations are laid out two per page, each with three breakdowns (generator×domain, generator×attack and domain×attack) on a colour scale fixed to $[0, 100]$, so panels are comparable across pages. The configuration we report appears again at the end, one breakdown per page at full size.

Attack	+F n→n	+F n→r	+F r→r	+F r→n	−F n→n	−F n→r	−F r→r	−F r→n
alternative_spelling	88.8	88.4	95.5	95.0	95.6	95.5	**97.3**	97.1
article_deletion	89.1	88.8	95.9	95.3	96.3	96.1	**98.0**	97.9
homoglyph	88.8	32.8	89.7	95.3	95.4	11.0	85.8	**97.0**
insert_paragraphs	89.3	88.7	96.4	96.2	95.4	95.3	**97.2**	97.0
none	88.8	88.4	95.8	95.3	95.4	95.3	**97.2**	97.0
number	88.7	88.2	95.8	95.3	95.4	95.3	**97.3**	97.1
paraphrase	77.6	75.9	90.1	**90.4**	83.1	82.8	88.9	89.6
perplexity_misspelling	89.0	88.5	95.8	95.3	95.9	95.7	**97.5**	97.4
synonym	83.7	83.0	94.7	93.8	93.5	93.3	**97.0**	96.8
upper_lower	87.8	87.5	95.2	94.3	97.2	97.1	**98.2**	98.1
whitespace	88.8	86.0	95.5	95.3	95.4	95.3	**97.2**	97.0
zero_width_space	88.8	10.9	83.3	95.3	95.4	1.1	73.8	**97.0**

Table 19: TPR@1 % FPR by attack, all eight factorial cells (cell labels as defined in Appendix G). Best per row in bold.

Domain	+F n→n	+F n→r	+F r→r	+F r→n	−F n→n	−F n→r	−F r→r	−F r→n
abstracts	97.5	93.3	**99.3**	99.1	95.2	82.6	95.4	96.1
books	90.2	78.3	91.5	93.1	96.2	80.8	95.5	**98.7**
news	94.5	79.5	94.1	**96.4**	94.5	79.2	89.2	95.6
poetry	38.1	30.4	83.7	82.4	87.9	71.3	91.5	**94.5**
recipes	99.5	91.7	99.5	**99.6**	99.1	84.0	99.3	99.5
reddit	89.8	70.2	94.1	**95.4**	90.9	76.4	91.5	93.5
reviews	94.7	79.9	94.5	96.4	96.0	80.5	95.1	**97.8**
wiki	95.4	81.4	92.4	95.4	96.4	81.0	92.8	**96.9**

Table 20: TPR@1 % FPR by domain, all eight factorial cells. Best per row in bold.

Generator	+F n→n	+F n→r	+F r→r	+F r→n	−F n→n	−F n→r	−F r→r	−F r→n
chatgpt	87.5	75.6	96.4	95.1	99.4	83.1	97.6	**99.5**
cohere	75.3	64.3	82.3	84.4	85.5	71.1	84.9	**89.8**
cohere-chat	79.5	67.1	89.7	91.4	90.4	75.6	89.3	**92.5**
gpt2	95.1	83.6	94.3	97.4	93.6	79.1	93.6	**97.6**
gpt3	82.6	70.5	91.5	92.7	90.0	74.9	92.6	**94.6**
gpt4	84.4	73.3	95.1	95.7	**99.1**	82.7	94.8	98.9
llama-chat	88.1	76.3	97.5	97.5	99.4	83.2	97.8	**99.6**
mistral	88.1	77.7	90.0	91.8	90.3	78.1	91.2	**93.9**
mistral-chat	88.2	75.5	96.9	97.6	98.3	82.3	95.9	**98.7**
mpt	90.0	78.3	92.5	93.5	92.0	77.6	93.0	**95.0**
mpt-chat	89.2	75.8	97.2	97.6	97.7	81.6	96.0	**98.4**

Table 21: TPR@1 % FPR by generator, all eight factorial cells. Best per row in bold.

Condition	+F n→n	+F n→r	+F r→r	+F r→n	−F n→n	−F n→r	−F r→r	−F r→n
TPR@1 % FPR, decoding								
greedy	90.8	79.3	96.6	96.8	96.1	81.3	96.6	**98.3**
sampling	84.1	71.9	90.7	92.6	92.9	77.7	91.0	**94.9**
TPR@1 % FPR, repetition penalty								
no	85.1	74.0	92.4	93.4	93.3	78.7	93.2	**96.0**
yes	91.8	78.5	95.9	97.2	96.7	81.0	94.9	**97.6**
TPR@5 % FPR, decoding								
greedy	99.5	94.2	99.5	**99.6**	99.3	87.7	99.3	99.5
sampling	98.2	91.3	98.1	98.5	97.7	84.6	97.5	**98.5**
TPR@5 % FPR, repetition penalty								
no	98.7	93.4	98.6	98.9	98.2	86.3	98.2	**98.9**
yes	99.2	91.6	99.2	**99.3**	99.0	85.8	98.8	99.2

Table 22: TPR by decoding strategy and repetition penalty at 1 % and 5 % FPR, all eight factorial cells. Best per row in bold.

Attack	+F n→n	+F n→r	+F r→r	+F r→n	−F n→n	−F n→r	−F r→r	−F r→n
alternative_spelling	99.0	99.0	**99.2**	99.1	98.7	98.7	99.1	99.0
article_deletion	99.0	99.0	**99.2**	99.2	98.7	98.7	99.2	99.1
homoglyph	99.0	87.9	97.9	**99.1**	98.7	43.0	96.4	99.0
insert_paragraphs	98.9	99.0	**99.2**	99.1	98.7	98.7	99.1	99.0
none	99.0	99.0	**99.2**	99.1	98.7	98.7	99.1	99.0
number	99.0	99.0	**99.2**	99.1	98.7	98.7	99.1	99.1
paraphrase	97.5	97.5	98.2	98.0	96.6	96.6	**98.4**	98.3
perplexity_misspelling	99.0	99.0	**99.2**	99.1	98.7	98.7	99.1	99.0
synonym	98.8	98.8	**99.3**	99.2	98.3	98.3	99.2	99.1
upper_lower	99.1	99.1	**99.3**	99.2	98.9	98.9	99.3	99.3
whitespace	99.0	98.9	**99.1**	99.1	98.7	98.7	99.1	99.0
zero_width_space	99.0	36.8	96.9	**99.1**	98.7	6.0	93.7	99.0

Table 23: TPR@5 % FPR by attack, all eight factorial cells. Best per row in bold.

Domain	+F n→n	+F n→r	+F r→r	+F r→n	−F n→n	−F n→r	−F r→r	−F r→n
abstracts	98.8	97.9	**99.6**	99.4	98.0	88.8	98.7	98.6
books	99.2	88.5	97.9	99.0	98.9	83.6	97.9	**99.5**
news	98.8	90.6	99.5	**99.6**	98.4	86.7	97.6	99.0
poetry	98.5	94.2	97.8	98.4	98.6	85.3	98.4	**98.8**
recipes	**99.9**	97.5	99.9	99.9	99.9	92.6	99.8	99.8
reddit	97.8	91.3	**98.3**	97.9	97.2	83.9	97.6	98.1
reviews	99.2	91.8	98.9	99.2	98.7	84.0	99.0	**99.5**
wiki	98.5	90.3	98.8	**98.9**	98.3	84.1	98.1	98.8

Table 24: TPR@5 % FPR by domain, all eight factorial cells. Best per row in bold.

Generator	+F n→n	+F n→r	+F r→r	+F r→n	−F n→n	−F n→r	−F r→r	−F r→n
chatgpt	99.7	93.1	**99.7**	99.7	99.7	86.0	99.6	99.7
cohere	97.2	91.1	97.0	**97.5**	95.4	82.3	95.5	96.9
cohere-chat	97.8	91.8	97.9	97.9	97.1	84.4	97.3	**98.0**
gpt2	99.5	94.5	98.8	99.4	99.2	88.1	98.3	**99.5**
gpt3	98.1	92.5	**98.4**	98.3	97.2	85.6	98.2	98.3
gpt4	99.8	93.0	99.7	**99.9**	99.7	85.5	99.1	99.8
llama-chat	99.9	93.2	99.9	99.9	99.9	86.3	99.8	**100.0**
mistral	97.6	93.1	97.7	98.2	97.2	87.9	97.2	**98.3**
mistral-chat	99.7	93.0	99.7	**99.8**	99.6	86.3	99.3	99.7
mpt	97.6	92.0	97.9	**98.1**	97.4	85.8	97.4	98.1
mpt-chat	99.6	92.0	99.6	99.6	99.4	85.8	99.4	**99.7**

Table 25: TPR@5 % FPR by generator, all eight factorial cells. Best per row in bold.

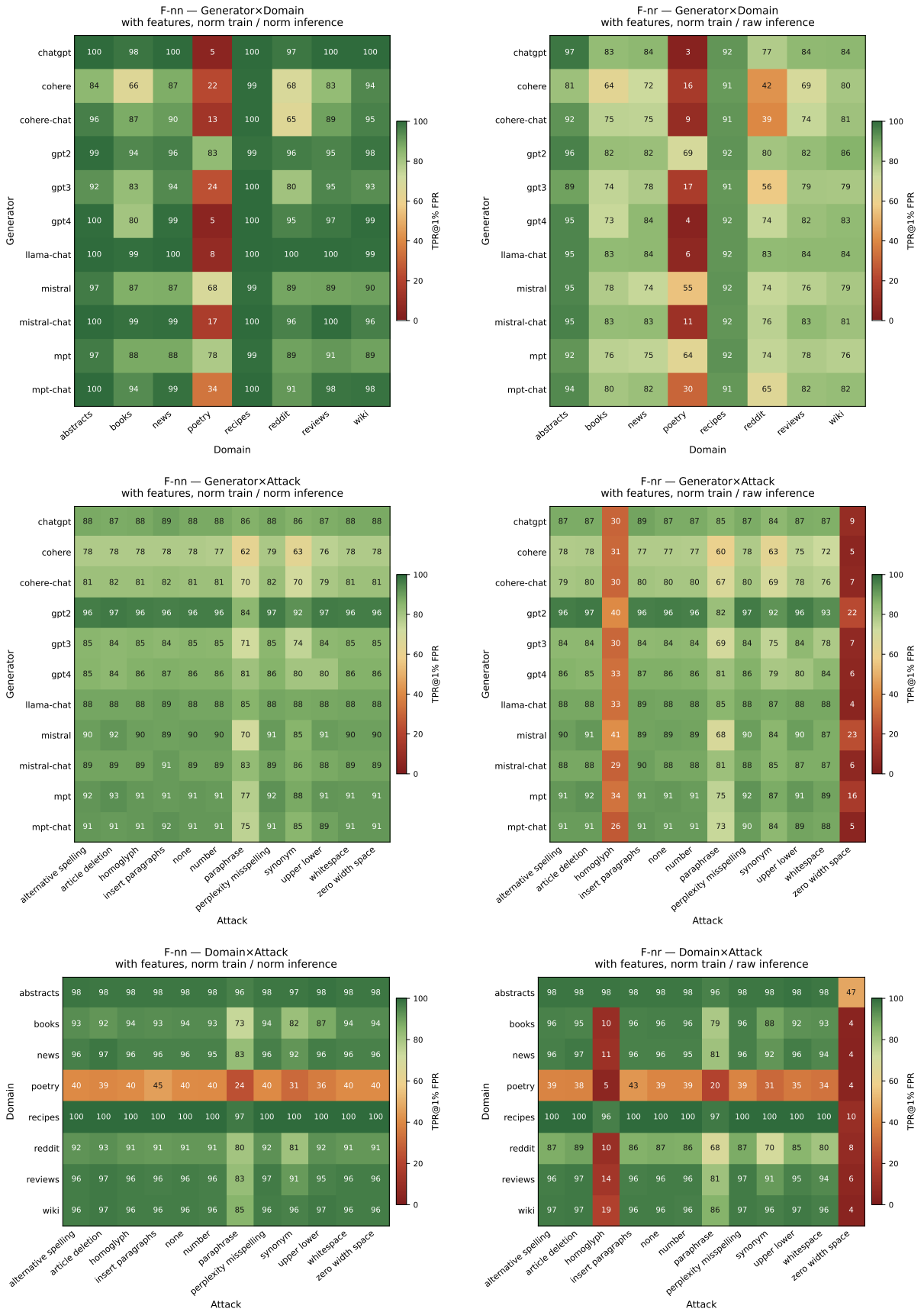


Figure 4: Left: cell +F, norm→norm. Right: cell +F, norm→raw. Rows: generator×domain (top), generator×attack (middle), domain×attack (bottom). TPR@1 % FPR.

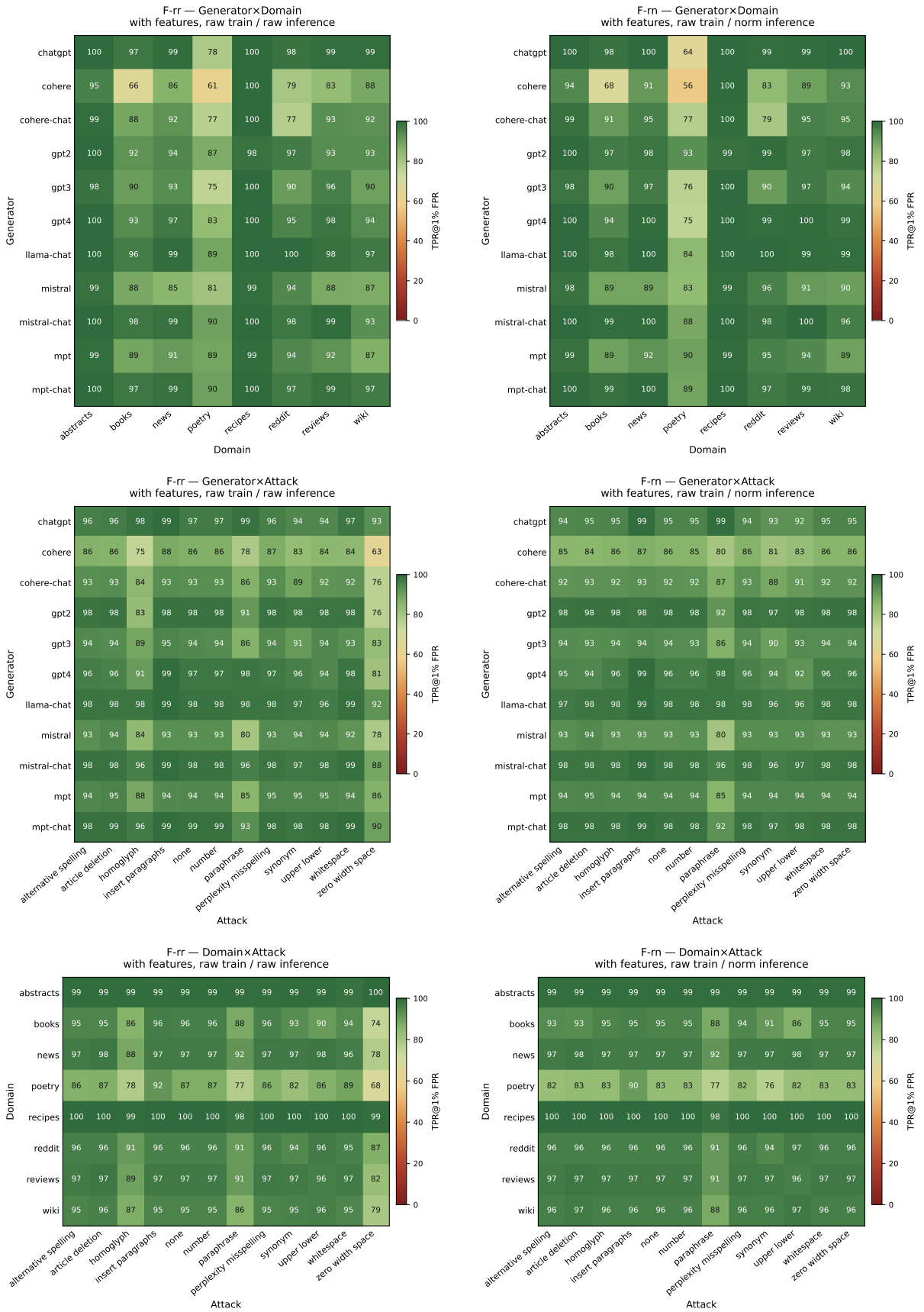


Figure 5: Left: cell +F, raw→raw. Right: cell +F, raw→norm. Rows: generator×domain (top), generator×attack (middle), domain×attack (bottom). TPR@1 % FPR.

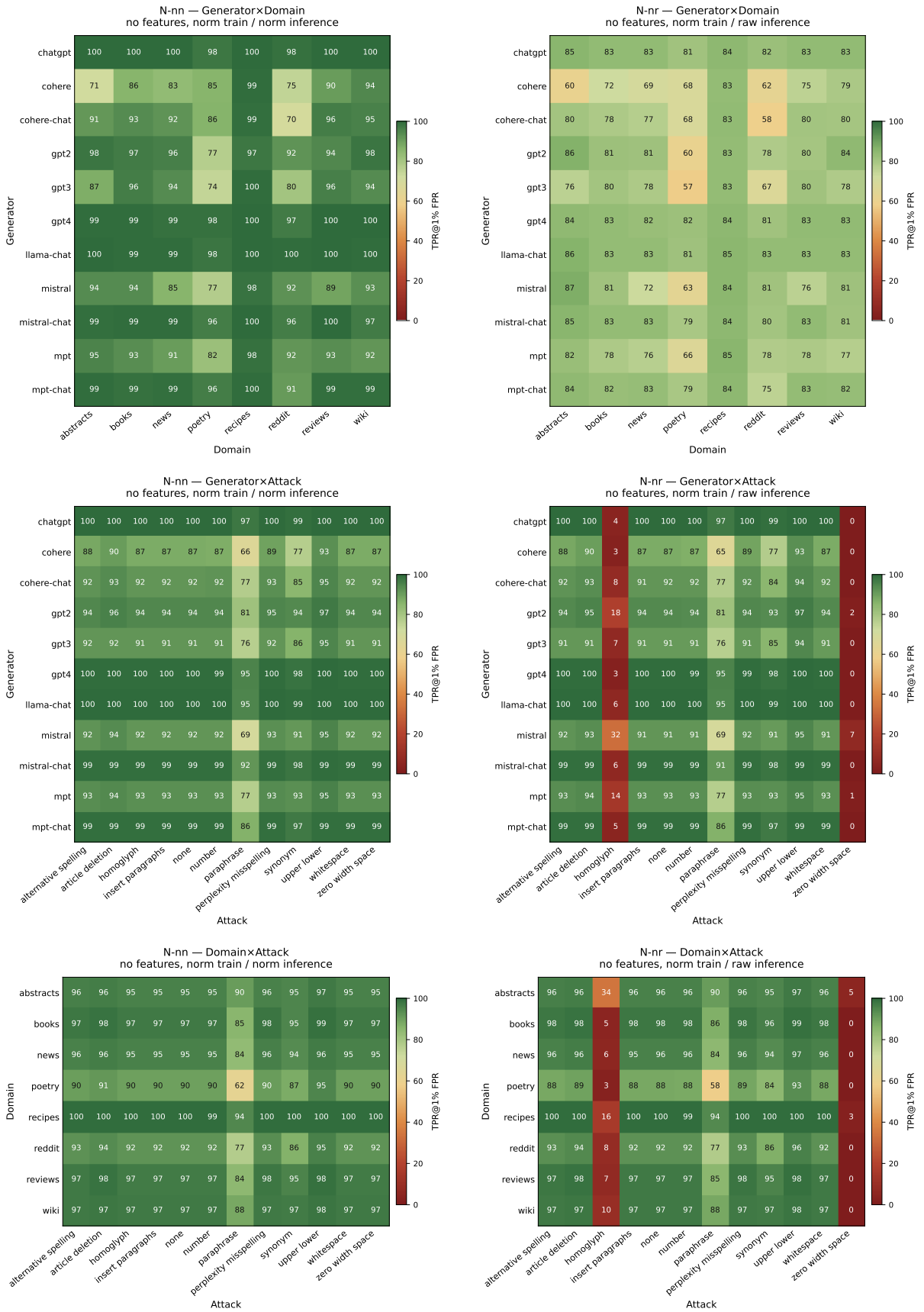


Figure 6: Left: cell −F, norm→norm. Right: cell −F, norm→raw. Rows: generator×domain (top), generator×attack (middle), domain×attack (bottom). TPR@1 % FPR.

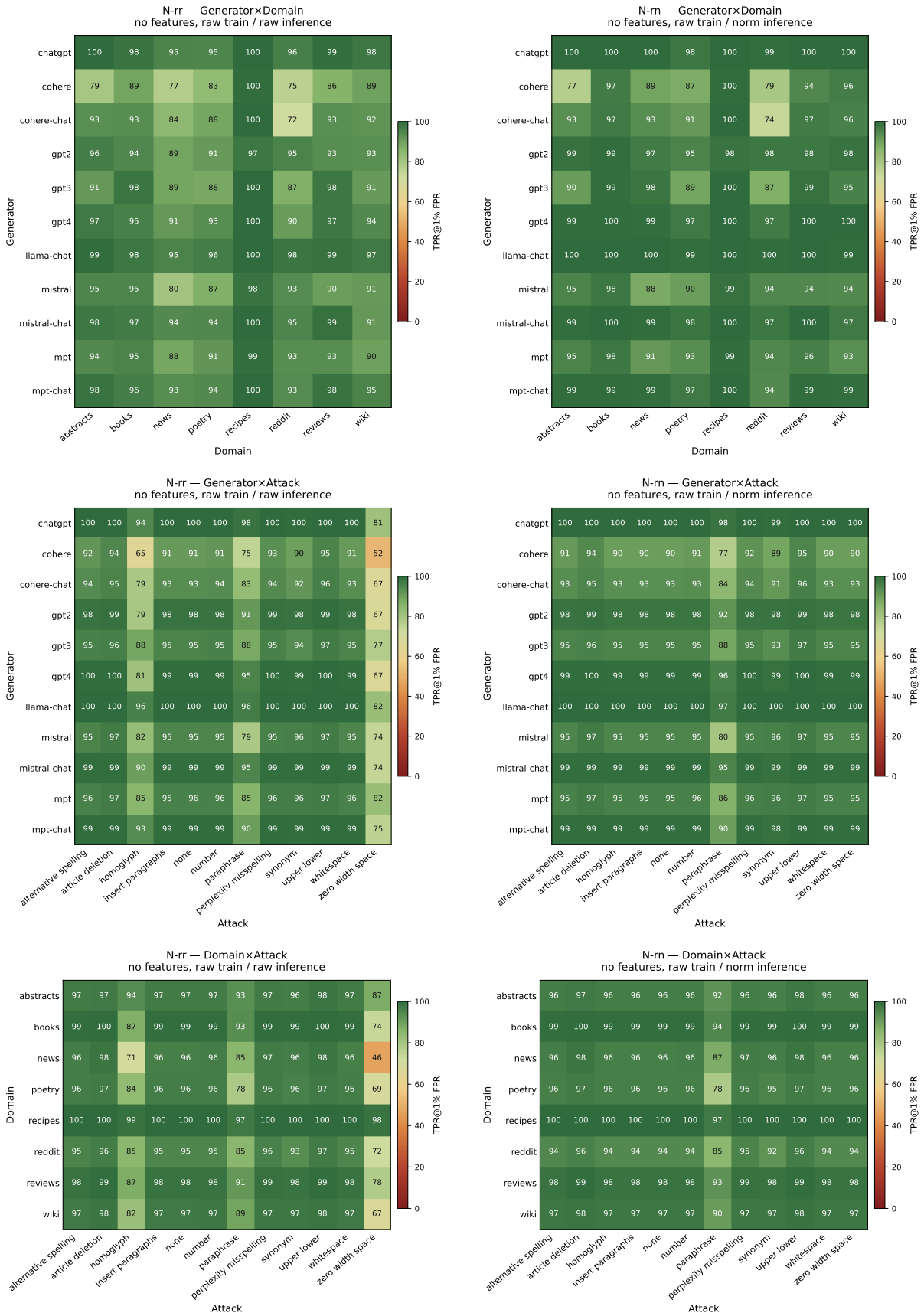


Figure 7: Left: cell −F, raw→raw. Right: cell −F, raw→norm. Rows: generator×domain (top), generator×attack (middle), domain×attack (bottom). TPR@1 % FPR.

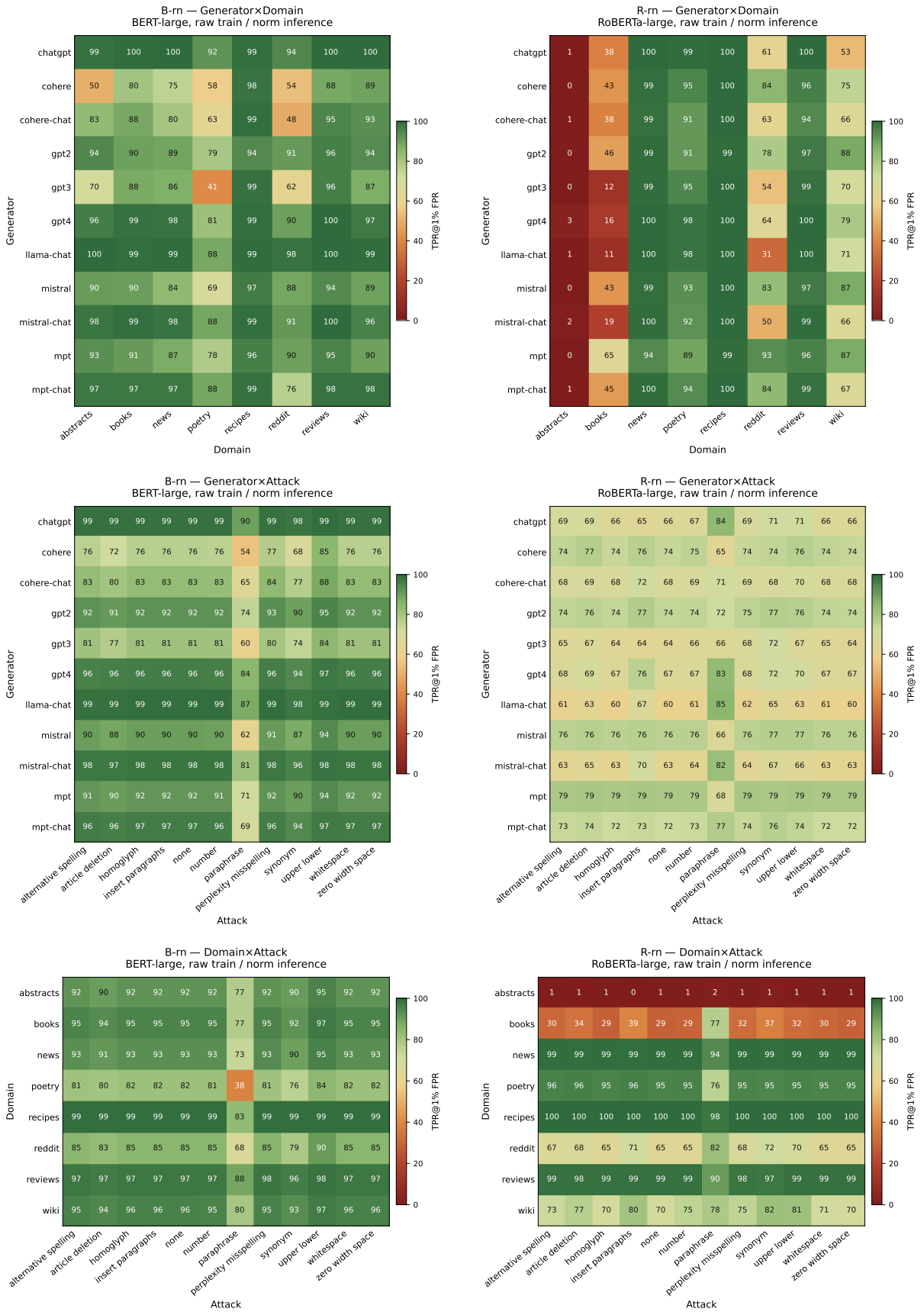


Figure 8: Left: cell BERT, raw→norm. Right: cell RoBERTa, raw→norm. Rows: generator×domain (top), generator×attack (middle), domain×attack (bottom). TPR@1 % FPR.

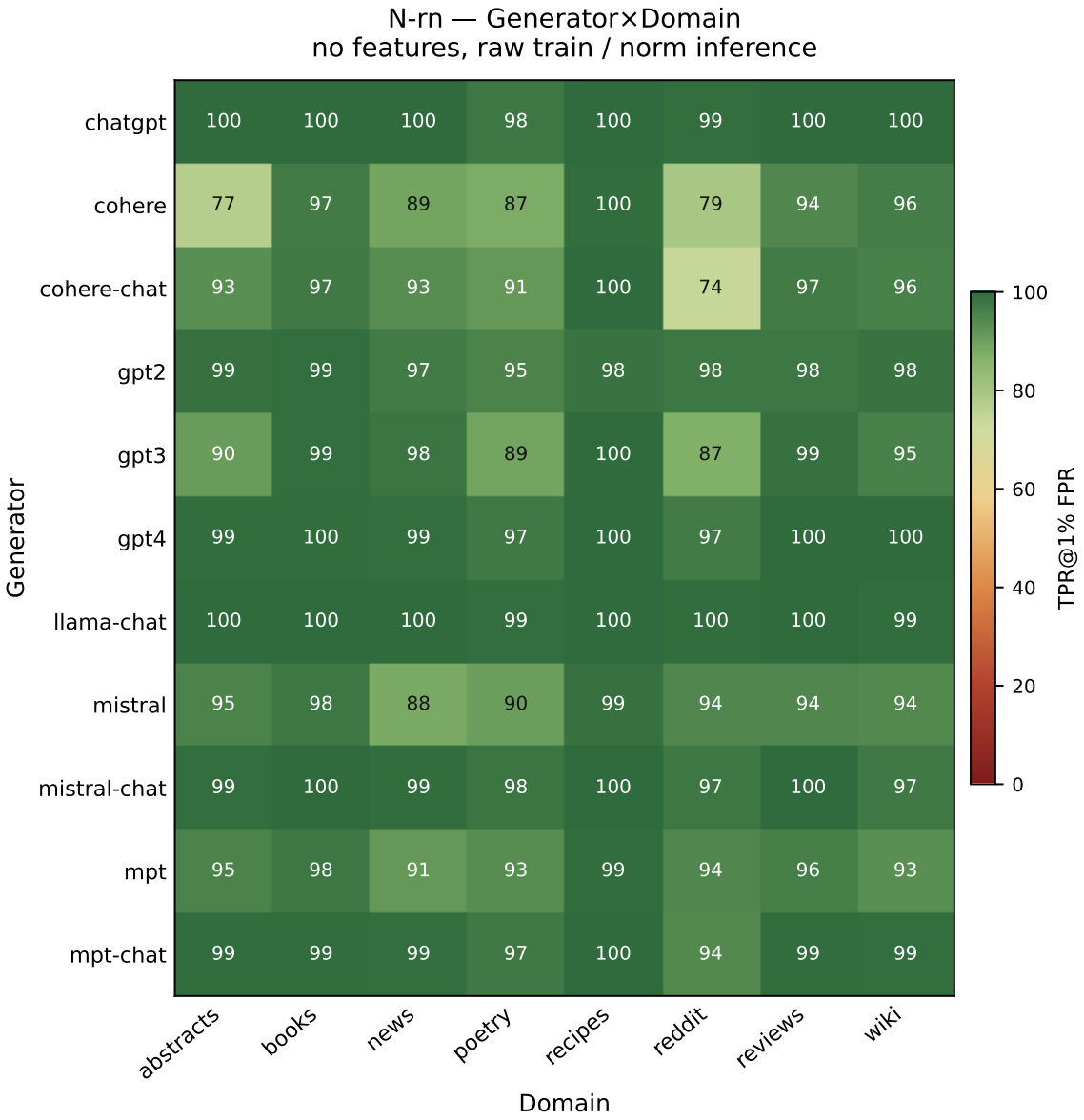


Figure 9: The configuration we report (−F, raw→norm) at full size: generator×domain, TPR@1 % FPR.

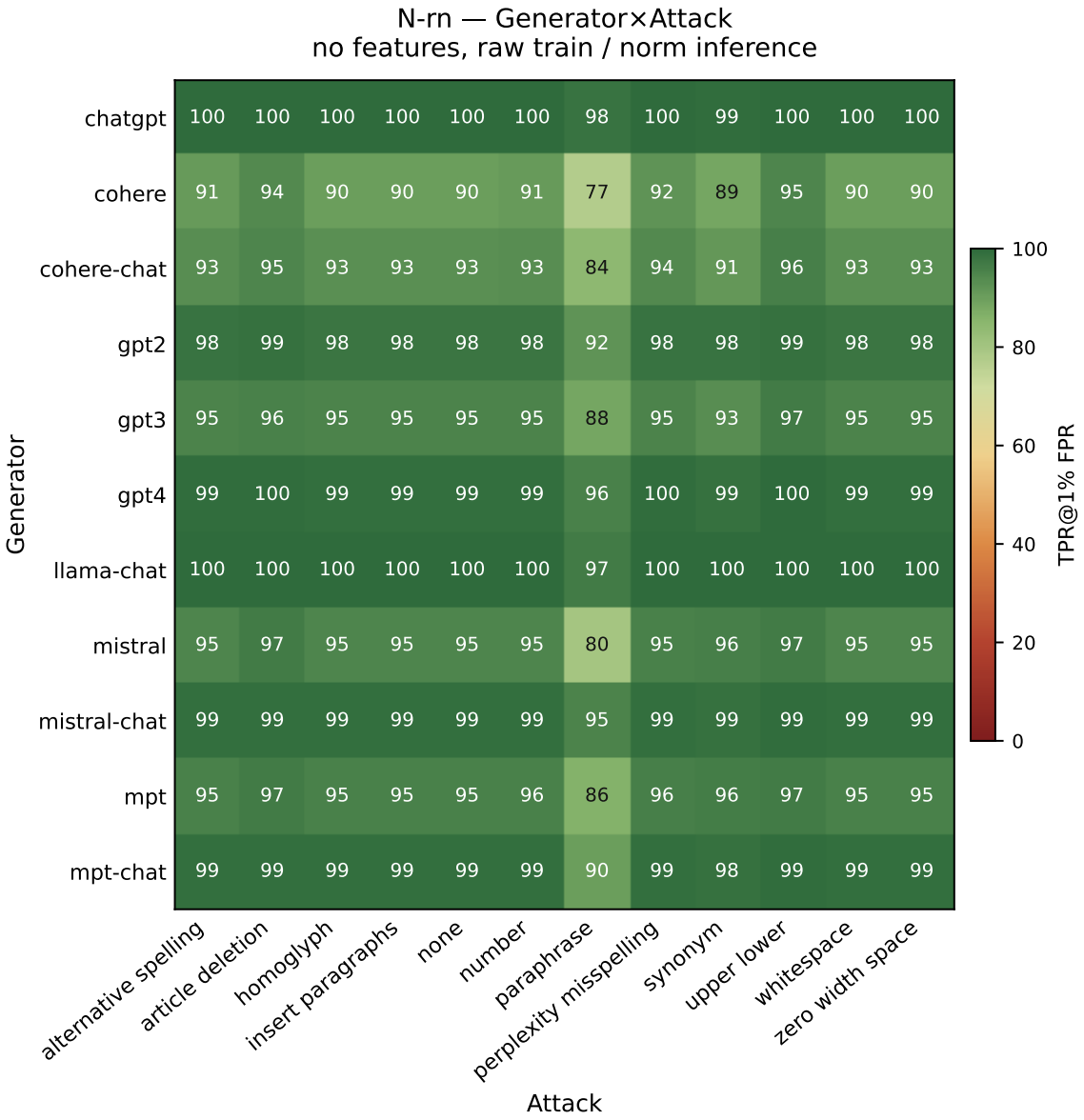


Figure 10: The configuration we report (−F, raw→norm) at full size: generator×attack, TPR@1 % FPR.

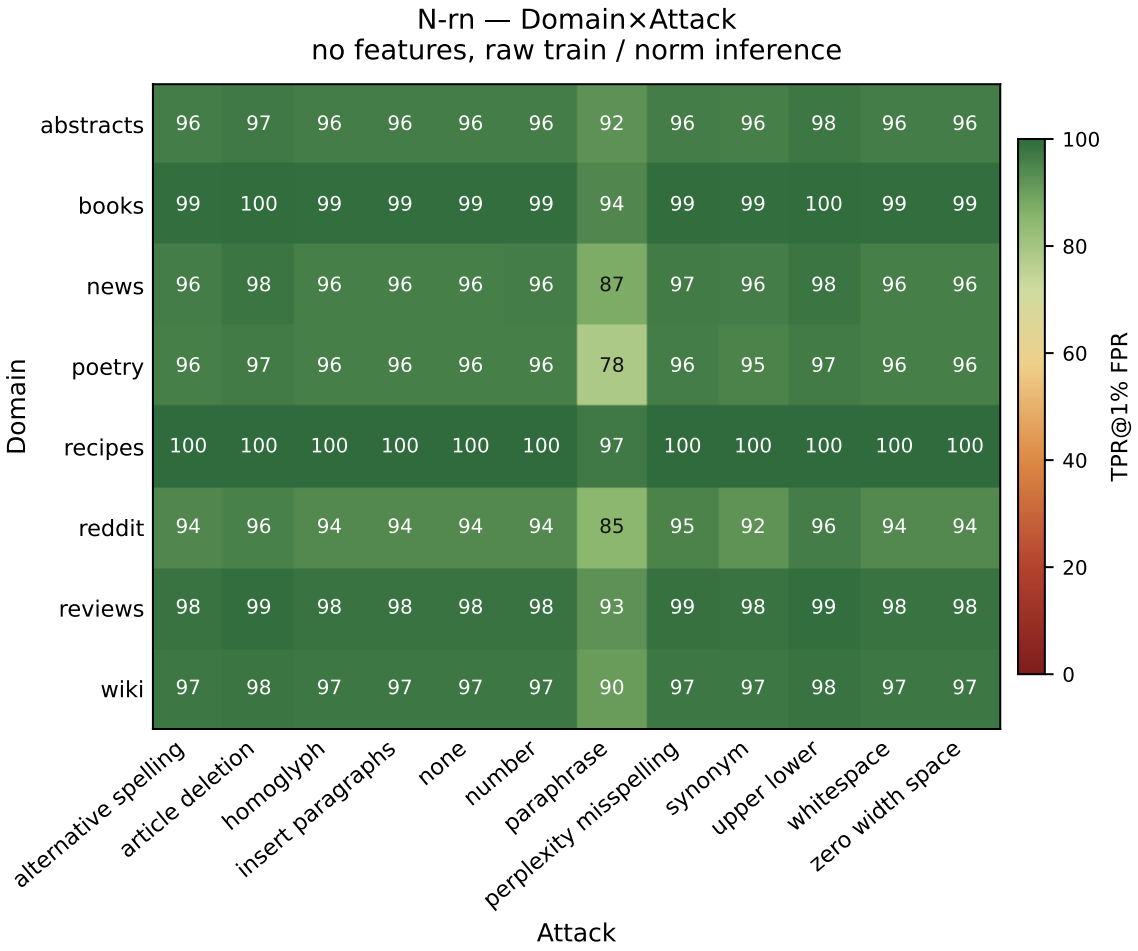


Figure 11: The configuration we report (−F, raw→norm) at full size: domain×attack, TPR@1 % FPR.