Pangram 4 Technical Report

Ben Glickenhaus* Katherine Thai* Jenna Russell Elyas Masrour Yue Han
Max Spero Bradley Emi

Pangram Labs *University of Maryland*

Abstract

We present **Pangram 4**, the latest deep-learning-based AI-text classification model from Pangram Labs. **We achieve an AUROC of 0.9916 with a false positive rate of 0.0041% and a false negative rate of 0.3396%**. In addition to its increased overall accuracy compared with Pangram 3, **Pangram 4** exhibits superior out-of-distribution generalization and robustness to adversarial attacks. Another novel contribution of **Pangram 4** is its improved ability to distinguish fine-grained edits and mixed AI-human co-authored text. We demonstrate improvements to both boundary detection tasks and the detection of interleaved AI assistance. Finally, we report metrics on standard AI detection benchmarks showing that **Pangram 4** achieves state-of-the-art performance on the AI text detection task across a wide variety of settings and domains.

1 Introduction

The world is being inundated with an unprecedented volume of AI-generated text. Over a third of new internet material (Dolezal et al., 2026), 26% of long-form social media posts (Pangram Labs, 2026), 9% of news articles (Russell et al., 2026a), and 21% of machine-learning conference reviews (Pangram Labs, 2025) are largely or fully AI-generated. "AI slop," a term for unwanted AI content, is used to describe text that is obviously AI-generated (Shaib et al., 2026) or that offloads cognition onto the recipient.[1] AI is a tool with the power to automate many long-horizon tasks, but with that power comes increased potential for societal harm.

While it has been shown that individuals who regularly use AI can detect AI text by eye (Russell et al., 2025), this is a laborious task that requires both time and experience to do well. This creates an effort asymmetry between readers and writers: AI can produce well-formed text that could plausibly have come from an expert, nearly for free, but readers still incur the burden of understanding whether the content they are ingesting is meaningful, grounded, and well-researched. This strains existing systems that lack a way to reliably discriminate between human and AI text.

Additionally, people are finding ways to incorporate AI into their writing workflows, producing documents of mixed authorship. These uses, while legitimate, do not diminish the societal harms caused by LLM-powered bots and undisclosed, fully AI-generated content polluting information

*Equal contribution.

[1] https://ampblog.substack.com/p/what-makes-slop-slop

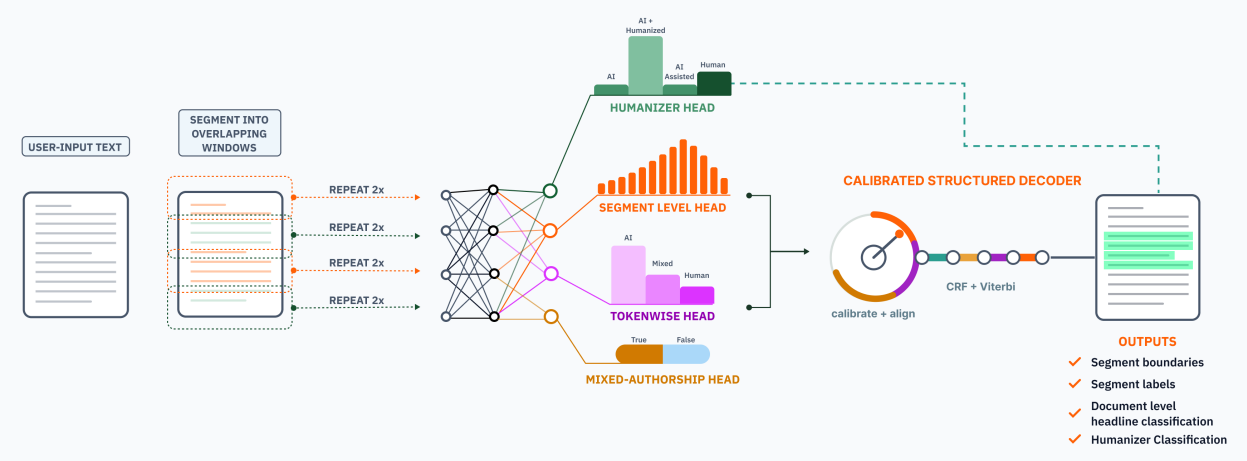

Figure 1. Overview of the **Pangram 4** architecture.

ecosystems (Raj et al., 2026; Drayson et al., 2025; Yu et al., 2026). To effectively mitigate these harms, it is imperative to understand the degree of human and AI involvement in written documents. This task motivates the creation of an AI-text detection model with high accuracy and granularity on AI, human, and AI-assisted text.

We introduce **Pangram 4**, the most capable AI text detection model developed by Pangram Labs. Designed to detect frontier models such as Claude Fable 5 (Anthropic, 2026a) and GPT-5.6 Sol (OpenAI, 2026c), **Pangram 4** achieves the highest accuracy on the AI-text detection task while maintaining a false positive rate of 0.0041% (roughly 1 in 24,000). We build on Pangram 3, an AI detection model based on the EditLens architecture (Thai et al., 2026), with data and architectural improvements. **Pangram 4** represents a new frontier in AI text detection technology, with state-of-the-art precision and recall, performance on mixed-authorship text, fine-grained span predictions, and robustness to humanizer attacks.

Contents

2 Data

In this section, we report on the composition of our training set. We train on a wide distribution of domain sources, languages, and generator models. The human-written sources in the training dataset undergo various transformations, including synthetic mirroring, AI-editing, and translation, to produce the AI-generated and AI-edited texts that are present in the dataset.

2.1 Human-written Datasets

Pangram is trained on a wide variety of datasets consisting of human writing from several sources, domains, and languages, similar to corpora used to train large language models. All datasets used to train Pangram are either openly licensed for commercial use or their rights are owned by the company. **Pangram 4 is not trained on user-submitted data, customer data from API users, or data obtained by unauthorized Internet crawls**. A rough distribution of the categories of text used to train the model is presented in Figure 2.

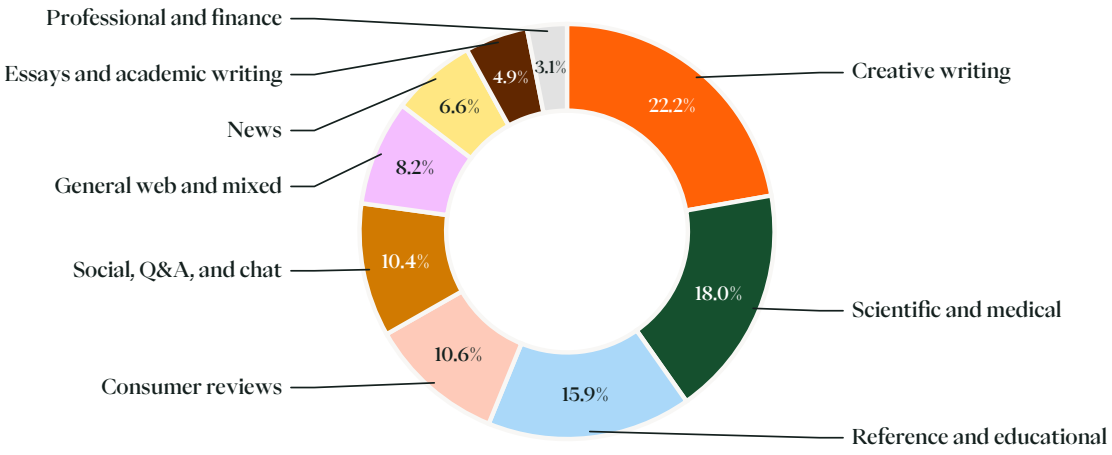

Figure 2. Approximate breakdown of human source text by category.

2.2 AI-Generated Datasets & Synthetic Mirroring

As in previous versions of Pangram, the AI examples in our dataset are generated entirely in-house. We generate data from a distribution of the most widely used LLMs. Our synthetic mirroring algorithm is described in the original Pangram technical report (Emi and Spero, 2024). This method is used to produce a diverse dataset of AI text and ensure topic invariance. We want the model to learn "How was this text written?" as opposed to "What is this text about?" An example of a synthetic mirror is as follows:

Synthetic mirror example. The synthetic mirror is produced in two steps:

1. **Prompt:** "What is the topic of this article? ⟨*Original Document*⟩"
   LLM Response: topic X

2. **Prompt:** "Write an article about X"
   LLM Response: the *synthetic mirror* of the original document

Synthetic mirrors are often grounded by additional metadata when available, such as article titles or keywords. For question-and-answer datasets, the question alone may suffice as a synthetic mirror prompt. To prevent synthetic mirrors from repeating human text, a final check compares the generated text against its original source and discards examples where a significant portion of the original document is repeated verbatim.

2.3 AI-Assisted Text

Several recent works (Zhang et al., 2024; Saha and Feizi, 2025; Artemova et al., 2025; Saha et al., 2026; Quaremba et al., 2026; Dycke et al., 2026; Bsharat et al., 2026) benchmark AI detectors' ability to detect AI-polished, mixed, or otherwise co-authored writing that does not fall neatly into either the human-written or AI-generated categories. Our dataset consists of AI-assisted text in addition to human-written text and fully AI-generated text. We generate the AI-assisted text according to the method described in the EditLens paper (Thai et al., 2026).

3 Task Formulation

3.1 What is AI-generated Text?

In order to define a tractable problem, we must first scope our definition of AI-generated text. We cannot define this as "any text that was produced by any large language model." A large language model can simply memorize and reproduce human text, either from its training set or from its context window. Additionally, a large language model can also produce purely factual information, such as the answers to "What is the capital of France?" or "What is Martin Luther King Jr.'s birthday?" Answers to both of these questions would be ambiguous, and thus must be considered out of scope.

Pangram defines AI-generated text as **original, substantial natural-language prose generated by an LLM in response to an open-ended writing task or question**. Pangram is designed to detect responses to open-ended prompts and questions, rather than responses to questions that have a single correct answer.

In addition to the open-generation criterion, we also require that, for AI text, the number of output tokens is greater than the number of input tokens. While AI summarization is a popular editing task, we must consider the output to be human if the AI does not add any original tokens of its own. Finally, we require that text be at least 50 words long to be considered for analysis. This is a necessary condition for the model to have enough signal to make a confident prediction. This excludes short, conversational replies from Pangram's scope.

3.2 Heterogeneous vs. Homogeneous Mixed Text

To motivate our labeling and architecture choices, we first introduce the concept of heterogeneous vs. homogeneous mixed text, which was originally introduced in the EditLens paper (Thai et al., 2026). In heterogeneous mixed text, each token has a clear authorship label, and each segment can be directly attributed to a single author, either human or AI. An example is a situation where a human writes a paragraph and then asks AI to continue it.

In homogeneous mixed text, authorship is entangled by the editing process. An example of this kind of mixed text is a human who writes a paragraph and then goes back and forth with an AI to

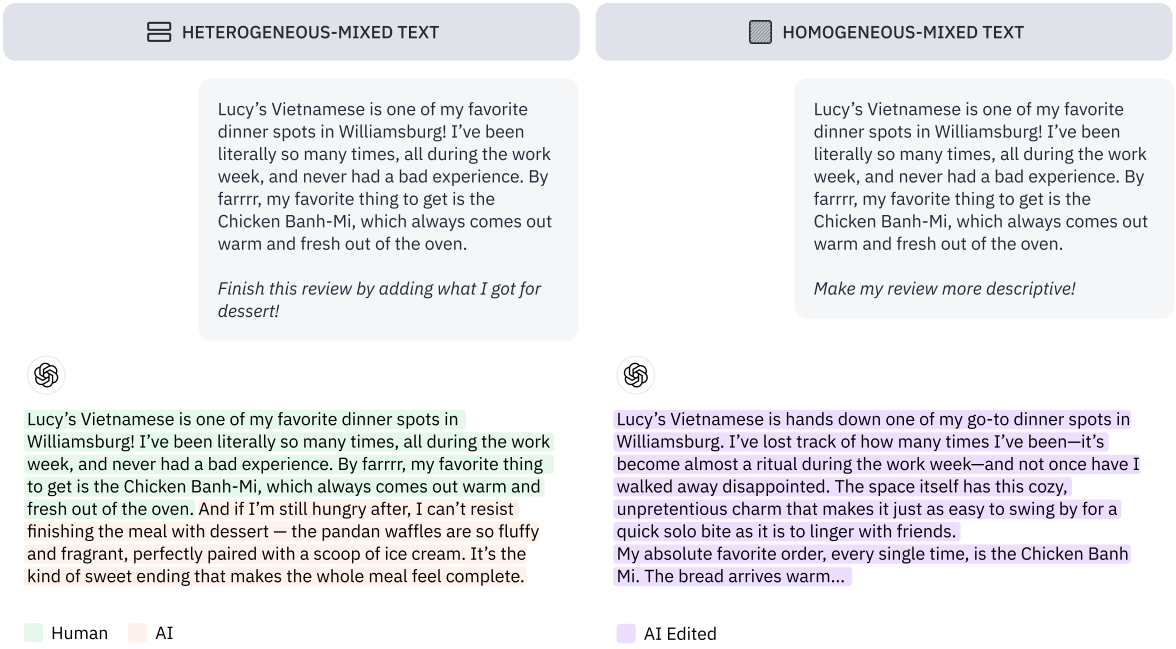

Figure 3. The difference between heterogeneous and homogeneous mixed text. In heterogeneous mixed text, each token has a well-defined author. In homogeneous mixed text, authorship attribution is mixed and ambiguous.

edit it. In this situation, the paragraph no longer has a distinct author; there are contributions from both.

Additionally, the provenance of the surface style of a text can be distinct from the provenance of its underlying ideas (Russell et al., 2026b). A human can have a very well-formed, concrete idea that is then worded and rephrased by an AI assistant: this is also an example of co-authorship. We use this to motivate our labeling schema, wherein we aim to differentiate fully AI-generated text from text where a human came up with the original idea, and then the AI was responsible for the final lexical choices that appear in the writing.

3.3 Task Definition

To our knowledge, **Pangram 4** is the first AI detection model that is able to identify both homogeneous and heterogeneous mixed writing *simultaneously in a single pass*. **Pangram 4** takes an input sequence of N tokens and produces an output of logits $Z_{\text{tok}} \in \mathbb{R}^{N\times 3}$ over the classes $\{$HUMAN, AI-ASSISTED, AI-GENERATED$\}$ for each token.

1. HUMAN means that the text was solely authored by humans, or that the amount of AI contribution is so low as to be negligible (e.g., light copyediting, literal translation, spelling and grammar fixes).
2. AI-ASSISTED means that the text is co-authored **and** homogeneously mixed. Although this label is nebulous for a single token, a segment of consecutive tokens with the AI-ASSISTED label indicates genuine co-authorship between humans and AI.
3. AI-GENERATED means that the text is solely authored by AI, or that the text is significantly more AI-authored than human-authored—in this case, we would define text predicted as AI-

GENERATED to be novel text from an LLM consistent with an open-ended prompt that allowed for original content solely from the LLM.

Because these three labels are predicted on a per-token basis, a prediction of heterogeneous mixed text is expressed as a combination of these three labels across the sequence. We also note that, as a corollary of this formulation, text can be simultaneously heterogeneous and homogeneous mixed: there may be some sections that are co-authored, and some sections that have a single author.

3.4 Humanization as an Auxiliary Task

As Pangram becomes a more capable AI detector, adversarial actors have greater incentive to evade it. We define *humanization* as the deliberate transformation or obfuscation of AI-generated text to evade AI-text detection (Masrour et al., 2025). The auxiliary output estimates whether a document exhibits characteristics consistent with humanization.

We formulate humanization detection as a four-way, document-level classification task over HUMAN, AI-GENERATED, HUMANIZED-AI, MIXED-AUTHORSHIP. This supplemental classifier is applied when **Pangram 4** detects evidence of AI-generated content. It identifies evasion techniques such as deliberate typo injection, casing changes, synonym substitution, homoglyph attacks, and transformations produced by services that advertise humanization or AI-detector evasion. Incidental artifacts, such as PDF-extraction errors, copy-and-paste corruption, OCR errors, and encoding or Unicode-normalization problems, are not classified as humanization.

3.5 Soft N-Grams Labeling

We do not have ground-truth tokenwise labels for mixed text *a priori*. Therefore, we compute tokenwise labels for AI-assisted text using the method described in Algorithm 1, which we call Soft N-Grams Labeling. Following the assumptions in EditLens, we synthetically generate examples of mixed human- and AI-authored text by starting with confirmed human documents and applying AI edits to these texts. The input to the soft n-gram labeler is a human document S and its corresponding AI-edited document T, which we refer to as the *source* and the *target*, respectively.

We diverge from EditLens, however, in the *resolution* of the labeler. While EditLens (and Pangram 3) use a scalar distance metric in the embedding space of an LLM to compute the similarity between S and T, in **Pangram 4**, we instead use a clause-level[2] soft N-grams approach loosely inspired by the evaluation metric proposed in ROUGE (Ng and Abrecht, 2015).

Clause-level n-grams. The goal of the soft n-grams labeler is to extract granular provenance of each clause in the target T. We select the clause as the unit of the labeler after observing that the clause is the smallest atomic unit corresponding to a single expressible "idea" whose provenance can be traced. We begin by splitting the target T into clauses using Claude Haiku 4.5 (Anthropic, 2025). We also tried classical linguistic approaches to clause splitting, but found Claude Haiku more accurate and less brittle on unusual edge cases. For each clause in T, we ask the following questions:

1. For this clause y, is there an exact or near-exact lexical match to this clause in S? If so, label this clause HUMAN.
2. If no lexical match is found, for this clause y, is there a clause in S that semantically matches the idea in y? If so, label this clause AI-ASSISTED.

[2]A clause is defined as a group of words that contains a subject and a verb.

Algorithm 1 Pangram soft n-gram labeling

Require: Source text S, edited text T
Ensure: Labels for the edited text and AI fraction f_{AI}

Split S and T into clauses
for all clauses y in T **do**
    Find the most similar span x in S
    Compute lexical similarity $L(x, y)$ using token and character n-grams
    Compute semantic similarity $E(x, y)$ using text embeddings
    if $L(x, y)$ is high **then**
        Label y as HUMAN
    else if $L(x, y)$ or $E(x, y)$ indicates a substantial rewrite **then**
        Label y as AI-ASSISTED
    else
        Label y as AI-GENERATED
    end if
end for
Let C_{H}, C_{AA}, and C_{AG} be the character counts for each label
$D \leftarrow C_{\text{H}} + C_{\text{AA}} + C_{\text{AG}}$
$f_{\text{AI}} \leftarrow \dfrac{0.5C_{\text{AA}} + C_{\text{AG}}}{D}$
return clause labels and f_{AI}

3. Finally, if there is neither a lexical nor a semantic match to the clause y, then we deem the clause an original open-generation output and label this clause AI-GENERATED.

This gives us tokenwise labels, where each token inherits the label from its parent clause. To convert the tokenwise labels into document labels, we define the weighted AI fraction f_{AI}, which is the fraction of characters labeled AI-generated, plus half the fraction of characters labeled AI-assisted. We then supervise this quantity as a bucketed regression target following the methodology in EditLens.

Deletion and rearrangement invariance. We note that the Soft N-Grams Labeler is invariant to both deletions in the target T and rearrangements of clauses between S and T. We view this as a feature, not a bug: because at test time, our model only has access to T, it is ill-posed for the model to predict deletions from S, because it has no access to S. In informal language, it's impossible for the model to predict deletions because it can't see what was originally there. We also consider rearrangements to be non-edits from the perspective of the Soft N-Grams labeler. If all the sentences in a human-written document are merely reordered and not rewritten, we believe it is correct for our model to classify the text as human-written.

4 Training

In this section, we describe the training procedure for **Pangram 4**. We first describe the model architecture, ablations run to pick the backbone, and individual task heads. We then describe the training process, active learning, and calibration to generate confidence estimates.

Candidate	LoRA targets	Test FPR (%) ↓	Test FNR (%) ↓
A	Attention	0.480	1.277
A	Attention + dense	0.416	**0.930**
B	Attention	0.486	1.181
B	Attention + dense	0.495	1.410
B	Attention + routed experts	1.339	85.033
C	Attention	0.488	4.317
C	Attention + routed experts	0.506	5.037
D	Attention	0.458	5.336

Table 1. Iso-compute backbone ablations. Backbone identities are anonymized.

4.1 Architecture

Here we describe the architecture of the model used to train token- and segment-level predictions, how calibration was conducted, and how we address humanized text. The model architecture is depicted in Figure 1.

Backbone model. The model architecture of **Pangram 4** is based on a popular open-weight MoE model. We attach a single dense layer as a custom classification head for each task that uses hidden states computed by the shared backbone at the final sequence position as inputs. For an input sequence of length S, the shared backbone produces a hidden representation $\mathbf{h}_i \in \mathbb{R}^D$ at every position i. We attach an independent linear classification head for each task. We train on three window-level objectives: segment-level edits, mixed-authorship binary classification, and humanizer detection. These tasks use the representation at the final supervised sequence position, $\mathbf{h}_S$, as input. We use this approach (e.g., instead of mean pooling) because under causal attention, $\mathbf{h}_S$ has access to the complete input window. We also train a tokenwise provenance head that instead applies its linear projection to every $\mathbf{h}_i$, producing a separate prediction at each token position.

Base model ablations. We ran small-scale, iso-compute ablations against a variety of dense and sparse open-weight models to select the most promising candidates (performance is shown in Table 1). For each candidate, we additionally swept attention, attention + dense, and attention + dense + routed expert (for MoEs) LoRA module targeting. Each candidate is evaluated by its FNR at a fixed FPR on our holdout dataset. All further experimentation is done on this base model.

Segment predictions. Following EditLens (Thai et al., 2026), we model the AI authorship attribution problem as a continuous spectrum rather than a binary classification task. Each labeled document is chunked into segments of length $S = 512$ tokens. Each segment is then assigned a weighted AI fraction

$$f_{AI} = \frac{0.5C_{AA} + C_{AG}}{C_H + C_{AA} + C_{AG}},$$

where C_{H}, C_{AA}, and C_{AG} are character counts attributed to human, AI-assisted, and AI-generated text, respectively. We discretize $f_{AI} \in [0, 1]$ into 15 ordered buckets and train a segment-level classification head on the hidden state at the final supervised sequence position.

Tokenwise predictions. **Pangram 4** significantly increases prediction granularity. We are able to achieve this via a tokenwise prediction head. Let $H \in \mathbb{R}^{S \times D}$ denote the final-layer hidden states for

a window of S tokens with hidden dimension D. A shared linear token-classification head,

$$Z_{\text{tok}} = HW_{\text{tok}}^{\top}, \qquad W_{\text{tok}} \in \mathbb{R}^{3\times D},$$

produces logits $Z_{\text{tok}} \in \mathbb{R}^{S\times 3}$ over the classes {HUMAN, AI-ASSISTED, AI-GENERATED} for each token.

Repeat2. Tokenwise prediction introduces a new challenge with a causal-LM backbone. A segment-level classifier can use last-position logits to ensure the classification head attends to the entire sequence. Naively, this is inherently not possible for a tokenwise head—the hidden state for token x_i can ordinarily attend only to the prefix $x_{\leq i}$. This limitation manifests as significantly worse prediction accuracy on tokens at the start of a sequence. To fix this, we repeat the input sequence twice, as suggested by Repeat2 (Leviathan et al., 2025), such that

$$\widetilde{x} = (x_1, \ldots, x_S, x_1, \ldots, x_S).$$

This Repeat2 construction gives every supervised token access to the complete document context. We mask the token-classification loss over the entire first copy and supervise only the corresponding tokens in the second copy:

$$\widetilde{y} = (\underbrace{\texttt{ignore}, \ldots, \texttt{ignore}}_{S}, y_1, \ldots, y_S).$$

During training, Repeat2 is applied to each sampled tokenwise window. At inference time, long documents are divided into overlapping windows of at most $S = 512$ tokens with a stride of 256 tokens, giving adjacent full-length windows a 50% overlap. Repeat2 is applied to each window, and the resulting predictions are aligned using their source-token offsets and assembled before structured decoding. This allows us to aggregate logits for the same token across all windows in which it appears, such that the output is a document-length sequence with a three-class vector for each token in its original position.

Mixed-authorship head. Stage 2 additionally trains a binary, window-level mixed-authorship head. Its training target is positive when more than 15% of the supervised tokens in a window belong to provenance classes other than the dominant class. We use this head to predict whether a window contains a meaningful mix of token labels, which is useful for downstream decoding as we will discuss in Section 4.3.

Humanizer Head. Pangram 4 introduces a new supplemental task: humanization detection. We train a probe on the shared last-position logits that learns to differentiate regular AI text from humanized AI text. We train this classification head as a probe to reduce the risk of humanizer-specific learned representations increasing our false positive rate on the main AI detection task. In controlled ablations, letting gradients flow through the base model from the humanizer head kept top-level FPR equivalent, but concentrated the failures in specific registers: academic writing and ESL essays. We opt to train the humanizer head as a probe to prevent the adverse impact of false positives.

4.2 Training Process

Pangram 4 is trained using a two-stage process: first, we train a checkpoint supervised on just the segment and humanizer objectives. We then merge the LoRA adapters into the base model, initialize

Stage	Initialization	Trained heads	Input and schedule
1	Base backbone with a new LoRA adapter	Segment-level edit head (15-way); humanizer head (4-way, loss weight 0.25, stop-gradient)	Single-copy inputs of at most 512 tokens
2	Stage-1 adapter merged into the backbone; new LoRA adapter	Segment-level edit head (15-way); tokenwise provenance head (3-way); mixed-authorship head (2-way); humanizer head (4-way, loss weight 0.25, stop-gradient)	512-token source windows using Repeat2

Table 2. Pangram 4 training stages. Stop-gradient prevents the humanizer loss from updating the shared backbone in both stages.

a fresh LoRA adapter, and continue training with the addition of the tokenwise objective. We use this two-stage approach as an efficiency optimization. The Repeat2 approach used in tokenwise training results in roughly half the throughput since we process twice as many tokens. Stage 1 training lets us bootstrap useful representations in a more compute-efficient manner, allowing us to train on the tokenwise objective for fewer total steps. Table 2 summarizes the two stages.

Active learning and hard negatives. After training the **Pangram 4** model candidate, we do a round of offline active learning. This process involves running inference on a large pool of reserved data to find hard negatives: human text that the current checkpoint labels as AI. These samples are incorporated into our data pipeline, synthetically mirrored, and added to the training dataset. **Pangram 4** is then retrained from scratch on the resulting combined dataset.

4.3 Tokenwise Postprocessing

Compared to Pangram 3, **Pangram 4** uses a much more robust postprocessing algorithm that combines both token-level and segment-level signals and simultaneously handles calibration and smoothing. The new postprocessing algorithm is therefore much more precise in boundary detection and better calibrated on long documents.

4.3.1 Motivation

Long documents. The model's context length is 512 tokens. In Pangram 3, postprocessing handles longer documents by repeatedly scoring candidate text spans, but this procedure is imprecise and computationally inefficient. Our new algorithm instead aggregates predictions over a single sliding-window pass and combines all observations in a CRF decoding problem.

Fine-grained Detection. Boundary detection is particularly difficult for heterogeneous documents that interleave human-written, AI-assisted, and AI-generated passages. A tokenwise decoder increases the model's expressiveness by assigning one of these three authorship states to every token before converting contiguous labels into character-aligned segments.

Global Calibration. In Pangram 3, we calibrated individual segments and subsequently aggregated them with adaptive boundaries. Consequently, the effect of aggregation on the false positive and false negative tradeoff was difficult to control. The tokenwise algorithm unifies calibration and

postprocessing by producing a calibrated unary score for every token and decoding all token labels jointly.

4.3.2 Inputs and Outputs

For a document containing N source tokens, we divide the document into overlapping windows of at most 512 source tokens with a stride of 256; the final window is anchored to the end of the document. In Repeat2, the source tokens in each window are concatenated twice before model inference. Let $\mathcal{W}(i)$ denote the windows that contain source token i. Each window w produces three relevant observations:

1. a 15-bucket segment-head distribution indicating the overall estimate of the fraction of AI content in the segment: $\mathbf{p}_w^s \in \mathbb{R}^{15}$;
2. a ternary token-head logit vector $\mathbf{z}_{w,i}^t \in \mathbb{R}^3$ for each token in the window; and
3. a binary mixed-document distribution, whose positive-class probability is denoted q_w. We supervise this as an auxiliary task in which the model is trained to produce TRUE if the text has any mix of AI and human content, and FALSE if the text has homogeneous authorship.

For a document, the post-processor returns a sequence of segments, each of which has a hard prediction of either HUMAN, AI-ASSISTED, or AI-GENERATED.

4.3.3 Algorithm

Sliding-window inference. We run a forward pass of the model for every overlapping window. This creates multiple model observations for tokens that appear in more than one window. We then collapse these observations into one ternary unary potential per source token and one transition score per adjacent token pair.

Segment-head projection. The 15-bucket head estimates the weighted AI fraction as defined in subsection 3.5. Each bucket is projected onto three class anchors using triangular basis functions:

$$\phi_{\text{human}}(f) = \max(0, 1 - 2f), \tag{1}$$
$$\phi_{\text{assisted}}(f) = \max(0, 1 - 2|f - 0.5|), \tag{2}$$
$$\phi_{\text{AI}}(f) = \max(0, 2f - 1). \tag{3}$$

The ternary document prior for window w is a weighted average of the basis functions weighted by the predicted bucket probability vector.

$$\pi_w(c) = \frac{\sum_{b=0}^{14} p_w^s(b)\phi_c(f_b)}{\sum_{c'} \sum_{b=0}^{14} p_w^s(b)\phi_{c'}(f_b)}.$$

This prior peaks at human for bucket 0, AI-assisted near the middle buckets, and AI-generated near bucket 14.

Calibrated unary potentials. Raw token and segment logits are averaged across all windows containing token i, reducing the multiple observations of each token from sliding-window inference into a single 6-vector.

For each token, the calibration feature vector is

$$\mathbf{x}_i = \left[\ell_i^t(\text{human}) \quad \ell_i^t(\text{assisted}) \quad \ell_i^t(\text{AI}) \quad \ell_i^s(\text{human}) \quad \ell_i^s(\text{assisted}) \quad \ell_i^s(\text{AI})\right]^\top.$$

A multinomial logistic calibration model produces the CRF unary potential

$$u_i(c) = \alpha_c + \mathbf{w}_c^\top \mathbf{x}_i + \delta_c,$$

where α_c is a fitted intercept, $\mathbf{w}_c$ contains the fitted feature weights, and δ_c is an optional class-prior adjustment. These parameters are fitted on held-out calibration data and may vary between model releases.

Linear-chain CRF. The calibrated unary scores define a three-state linear-chain conditional random field. For adjacent labels a and b, the Potts transition potential at boundary i is

$$\tau_i(a, b) = \begin{cases} 0, & a = b, \\ \min(0, -\lambda + \gamma m_i), & a \neq b, \end{cases}$$

where m_i is the center-weighted mixed log-odds at the boundary, $\lambda \geq 0$ is the smoothness penalty, and γ controls how strongly mixed evidence can relax that penalty. Both values are selected using calibration data.

The maximum a posteriori label sequence is

$$\widehat{\mathbf{T}} = \underset{\mathbf{T} \in \{0,1,2\}^N}{\arg\max} \left[\sum_{i=1}^{N} u_i(T_i) + \sum_{i=1}^{N-1} \tau_i(T_i, T_{i+1}) \right].$$

Decoding and marginals. Multiclass Viterbi decoding computes $\widehat{\mathbf{T}}$. A numerically stable forward–backward pass separately computes the token marginals

$$P(T_i = c \mid \text{all model observations}).$$

Viterbi labels determine segment boundaries, while the marginals are retained as token-level class probabilities.

Sentence majority voting. Product output is constrained to sentence-level resolution. Sentences end at terminal punctuation (including a trailing quote or bracket), a blank line, or the end of the document. Tokens are assigned to sentences by character midpoint. Every token in a sentence is replaced by the sentence's majority Viterbi label. Deterministic tie-breaking is used to make the operation fully reproducible.

Minimum segment length. After sentence voting, the decoder repeatedly finds the first contiguous label run shorter than a configured minimum $L_{\min}$ and merges it into a neighbor:

1. if only one neighbor exists, that neighbor's label is used;
2. if both neighbors have the same label, that label is used;
3. otherwise, the longer neighboring run supplies the label; and
4. a deterministic rule resolves equal-length ties.

Merging continues until every remaining run has at least $L_{\min}$ tokens or the document contains only one run. The value of $L_{\min}$ is a configurable postprocessing choice rather than a fixed property of the model.

4.3.4 Segments and Document-Level Prediction

Adjacent tokens with the same final constrained label are coalesced and mapped back to a complete partition of the original character sequence. Document fractions are computed by character length:

$$f_c = \frac{\text{number of source characters assigned to class } c}{\text{number of source characters in the document}}.$$

4.3.5 Confidence

Token class marginals are computed from the CRF before sentence-majority voting and minimum-run merging. Token confidence is the largest marginal probability, and segment confidence is the mean token confidence within the segment. Therefore, confidence measures the peakedness of the unconstrained CRF posterior; it is not a recalibrated probability that the final constrained label is correct.

5 Evaluations

We evaluate **Pangram 4** on human, AI-generated, and AI-assisted data over a range of in-domain and challenging benchmarks. The goal is to evaluate performance across representative real-world use cases across varying domains and languages.

5.1 Evaluation Metrics

Our final prediction $Z_{\text{tok}} \in \mathbb{R}^{S\times 3}$ where S is the number of segments as described in Section 4 must be decoded into a document-level prediction for evaluation purposes.

For purposes of evaluation, we decode the final tokenwise output into a documentwise output using the following rule-based method:

$$\widehat{y} = \begin{cases} \text{Human}, & \text{if } f_{\text{human}} \geq 0.90, \\ \text{AI}, & \text{if } f_{\text{AI}} \geq 0.80, \\ \text{Mixed}, & \text{otherwise.} \end{cases}$$

When we evaluate our model on fully human and fully AI text, we use standard binary classification metrics. We define a **False Positive** to be a prediction of MIXED or AI when the ground truth is HUMAN, and a **False Negative** to be a prediction of HUMAN or MIXED when the ground truth is AI. Note that a prediction of MIXED counts as a failure for both false positive rate and false negative rate measurements, making our metrics stricter than threshold-based calibration on a binary logit distribution as is typical for other methods. For AUROC, TPR @ X% FPR and other rank-based metrics, we directly use the post-processed weighted AI fraction to rank the texts in the corpus.

5.2 Overall False Negative Rate

We generate 520,000 AI outputs and run Pangram on them to identify false negatives. At the production operating point, **Pangram 4** achieves an **overall false negative rate of 0.3396%**. This is an improvement over Pangram 3.3.2, which reports a FNR of 1.9942% on the same dataset.

Company	Model family	Generator	Release date	Pangram 3.3.2	Pangram 4
Anthropic	Claude	Claude Fable 5 (Anthropic, 2026a)	Jun. 9, 2026	1.030%	0.255%
		Claude Opus 5 (Anthropic, 2026c)	Jul. 24, 2026	0.585%	0.195%
		Claude Sonnet 5 (Anthropic, 2026e)	Jun. 30, 2026	0.750%	0.140%
		Claude Opus 4.8 (Anthropic, 2026b)	May 28, 2026	0.885%	0.235%
		Claude Sonnet 4.6 (Anthropic, 2026d)	Feb. 17, 2026	1.955%	0.215%
		Claude Haiku 4.5 (Anthropic, 2025)	Oct. 15, 2025	0.805%	0.130%
		Family aggregate	–	**1.002%**	**0.195%**
DeepSeek	V4	DeepSeek V4 Flash (DeepSeek-AI, 2026)	Apr. 24, 2026	1.625%	0.455%
		DeepSeek V4 Pro (DeepSeek-AI, 2026)	Apr. 24, 2026	2.050%	0.320%
		Family aggregate	–	**1.838%**	**0.388%**
Google	Gemini	Gemini 3.1 Pro Preview (Google, 2026a)	Feb. 19, 2026	6.380%	0.555%
		Gemini 3.5 Flash (Google, 2026b)	May 19, 2026	3.895%	0.440%
		Family aggregate	–	**5.138%**	**0.498%**
Google	Gemma	Gemma 4 31B IT (Google, 2026c)	Apr. 2, 2026	2.910%	0.430%
Meta	Llama	Llama 3.3 70B Instruct (Meta, 2024)	Dec. 6, 2024	4.405%	0.550%
Mistral AI	Mistral	Mistral Medium 3.5 (Mistral AI, 2026)	Apr. 29, 2026	0.835%	0.400%
Moonshot AI	Kimi	Kimi K2.6 (Moonshot AI, 2026)	Apr. 20, 2026	2.560%	0.395%
NVIDIA	Nemotron	Nemotron 3 Ultra 550B-A55B (NVIDIA, 2026)	Jun. 4, 2026	1.320%	0.310%
OpenAI	GPT	GPT-5.6 Sol (OpenAI, 2026c)	Jul. 9, 2026	3.240%	0.450%
		GPT-5.6 Terra (OpenAI, 2026c)	Jul. 9, 2026	2.040%	0.345%
		GPT-5.5 (OpenAI, 2026b)	Apr. 23, 2026	1.645%	0.345%
		GPT-5.4 (OpenAI, 2026a)	Mar. 5, 2026	1.145%	0.270%
		GPT-5.4 Mini (OpenAI, 2026a)	Mar. 17, 2026	0.900%	0.330%
		GPT-OSS 120B (OpenAI, 2025)	Aug. 5, 2025	0.395%	0.155%
		Family aggregate	–	**1.561%**	**0.316%**
Alibaba Cloud	Qwen	Qwen 3.7 Max (Qwen, 2026)	May 19, 2026	2.545%	0.315%
Tencent	Hunyuan	HY3 Preview (Tencent, 2026)	Apr. 23, 2026	1.830%	0.330%
Thinking Machines	Inkling	Inkling (Thinking Machines Lab, 2026)	Jul. 15, 2026	0.300%	0.190%
xAI	Grok	Grok 4.3 (xAI, 2026)	Apr. 17, 2026	2.645%	0.605%
Z.ai	GLM	GLM 5.2 (Z.ai, 2026)	Jun. 13, 2026	3.175%	0.470%
Overall				**1.9942%**	**0.3396%**

Table 3. False negative rate by generator model and family.

Synthetic benchmark construction. We evaluate **Pangram 4** on a comprehensive synthetic benchmark of 520,000 examples generated by 26 frontier open-source and closed-weight models. First, we use `Mistral-Small-24B-Instruct-2501` (Mistral AI, 2025) to filter prompts from the Chatbot Arena Conversations Dataset (Zheng et al., 2023). We chose the Chatbot Arena Conversations dataset so that the prompts in the benchmark reflect the real-world distribution of user prompts to LLMs. We filtered for prompts that request an open-ended generation or pose an open-ended question. Multiple-choice questions, math, and coding-related inquiries were considered out of scope for this benchmark. After identifying 20,000 in-scope prompts, we generate a response for each prompt using each of the 26 LLMs for a total of 520,000 attempted samples, of which 519,993 were successfully scored. We found 1,766 false negatives, yielding an FNR of 0.3396%, with a 95% Wilson confidence interval of (0.3242%, 0.3558%).

FNR by generation model. Table 3 displays results by generator model. Anthropic models have the lowest FNR, at 0.195%, and OpenAI models have an FNR of 0.316%. We find that there is no significant correlation between release date and FNR.

5.3 Overall False Positive Rate

We evaluate **Pangram 4** on over a million human-written English texts and find that we have a false positive rate of 0.0041%, with a 95% confidence interval of (0.0032%, 0.0050%). To put this in perspective, this is one false positive for every ∼24,300 requests. This is also an improvement to Pangram 3.3.2, which has a false positive rate of 0.0539% on the same dataset.

5.4 Performance by Length

AI detection is often harder on shorter texts, due to less text signal to compute predictions on. We evaluate performance by length, finding that **Pangram 4** outperforms Pangram 3.3.2 at every length bucket, as displayed in Figure 4.

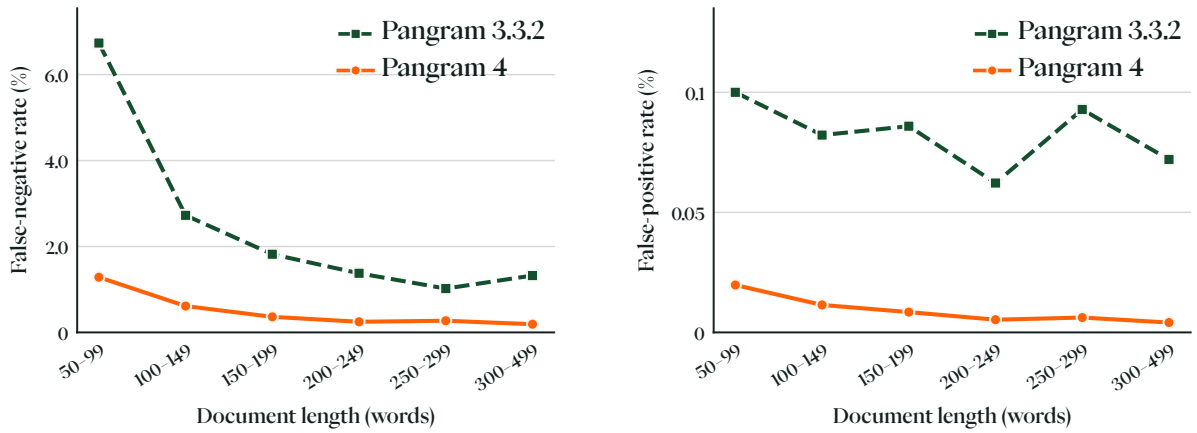

Figure 4. Error rates by document length for Pangram 3.3.2 and **Pangram 4**.

5.5 Performance on Mixed Authorship Text

5.5.1 Homogeneous Mixed Eval

FPR on AI-Polished Text. We define **AI-Assisted False Positive Rate (FPR)** as the rate at which a model incorrectly flags minor edits made by AI as AI. To evaluate the performance of our model at distinguishing lightly AI-edited texts from fully AI-generated texts, we create a dataset of AI-proofread and AI-polished text. We use consumer products (Grammarly[3], Apple Intelligence, and Gemini in Google Docs) and frontier LLMs (Opus 4.8, Gemini 3.1, and GPT-5.5) to apply light edits to 11,363 pieces of human-written student writing from ELLIPSE (Crossley et al., 2023), PELIC (Juffs et al., 2020), and ICNALE (Ishikawa, 2019). For example, we used the Writing Tools feature embedded in Apple's Pages word editor to "Proofread" some texts or prompted Opus 4.8 to "Correct minor academic style issues while keeping the draft recognizably human." For the full list of editing tools and prompts, see Table 16. In table 4, we show that Pangram 3.3.2 classifies AI-polished text as AI 0.18% of the time. **Pangram 4** only flags AI-polished text as AI 0.01% of the time–an 18x reduction in AI-Assisted FPR.

[3] https://app.grammarly.com/

Model	Human	Mixed	AI (FPR)
Pangram 3.3.2	11,329 (99.70%)	14 (0.12%)	20 (0.18%)
Pangram 4	11,301 (99.45%)	61 (0.54%)	**1 (0.01%)**

Table 4. Prediction distribution and AI-Assisted FPR on our AI polish evaluation set. Cells report count (percentage). $n = 11{,}363$ texts.

Recall on AI-Edited Texts. We define **AI-Assisted Recall** as the rate at which a model classifies an AI-assisted text as MIXED. Because there is no single definition for when a text should be considered AI-edited, we create two datasets with different standards for AI assistance.

We create the first dataset in a similar fashion to our AI Polish evaluation dataset. We use the same consumer products and frontier LLMs to apply editing instructions or prompts that demonstrate a clear intent to substantially modify the source text. For the full list of editing tools and prompts, see Table 17. As in our AI Polish dataset, these edits are applied to 14,990 human-written student essays. In Table 5, we show that Pangram 3.3.2 has an AI-Assisted Recall of only 5.54%, predicting HUMAN the majority (79.23%) of the time. **Pangram 4** has a much-improved AI-Assisted Recall of 55.01% on the same edited texts, predicting MIXED 10 times as often as Pangram 3.3.2.

Model	Human	Mixed (Recall)	AI
Pangram 3.3.2	11,877 (79.23%)	831 (5.54%)	2,282 (15.22%)
Pangram 4	6,201 (41.37%)	**8,246 (55.01%)**	543 (3.62%)

Table 5. Prediction distribution and AI-Assisted Recall on our AI Editing Prompts evaluation set. Cells report count (percentage). $n = 14{,}990$ texts.

We also create a second dataset where AI-assistance is defined by a quantitative metric: the Soft N-grams Labeling approach from subsection 3.5. First, we mine the WildChat dataset (Zhao et al., 2024) using GPT-5.5 nano for over 5,000 unique editing prompts submitted by users that are neither domain-specific (i.e., related to a field or discipline) nor source-specific (i.e., mentioning specific words, phrases, or arguments that should be modified). Then we randomly apply these edits using one of Opus 4.8, Gemini 3.1, and GPT-5.5 to human-authored Reddit posts, student essays, and creative writing. Finally, we score all resulting edited texts using Soft N-Grams Labeling and filter our dataset to 4,826 examples with an AI fraction of between 0.25 and 0.75. We deem texts with this score to be AI-assisted by any reasonable standard, as a score of 0.25 means that half of the words have been modified by AI or 25% of the words have been newly generated by AI (or some combination of the two). Conversely, a score of 0.75 indicates that the LLM generated 75% of the words in the human-written source. In table 6, we show that Pangram 3.3.2 only classifies 12.6% of these AI-assisted texts as MIXED. Additionally, Pangram 3.3.2 displays polarized behavior, preferring to classify these AI-assisted texts as HUMAN or AI rather than MIXED. On the other hand, **Pangram 4** classifies the majority of these texts as MIXED, achieving an AI-Assisted Recall of 65.7%, five times that of Pangram 3.3.2.

5.5.2 Heterogeneous Mixed Eval

Dataset Construction. For heterogeneous mixed eval, we aim to evaluate the resolution of our detector. We create datasets with interleaved human and AI text. These datasets contain

Model	Human	Mixed (Recall)	AI
Pangram 3.3.2	3,150 (65.27%)	609 (12.62%)	1,067 (22.11%)
Pangram 4	1,344 (27.85%)	**3,145 (65.17%)**	337 (6.98%)

Table 6. Prediction distribution and AI-Assisted Recall on our WildChat Edits evaluation set. Cells report count (percentage). $n = 4{,}826$ texts.

documents that alternate N fully human sentences with N fully AI-generated sentences, for $N \in \{1, 4, 8, 12, 16, 20\}$. We create these documents by taking long human-written documents from our test set, and replacing alternating N sentence chunks with AI sentences from a random choice of GPT-5.4, Claude Sonnet 4.6, and Gemini 2.5 Flash. We sample the original human documents from our creative writing and formal academic writing splits.

Results. In table 7, we report the tokenwise binary classification metrics stratified by N on this heterogeneous mixed dataset, as well as the document mean average error (MAE), which is the average difference between the predicted fraction of AI tokens and the ground-truth fraction of AI tokens (lower is better). **Pangram 4** is significantly more accurate than Pangram 3.3.2 at all resolutions on these challenging interleaved texts.

Qualitatively, **Pangram 4** is able to predict AI text at a much finer-grained resolution than Pangram 3.3.2. We display a representative example below in Figure 5.

Resolution	Model	Tokenwise Acc. ↑	Tokenwise Prec. ↑	Tokenwise Recall ↑	Doc MAE ↓
1	Pangram 3.3.2	46.12%	63.25%	22.43%	45.01%
	Pangram 4	**75.02%**	**92.82%**	**62.86%**	**18.27%**
4	Pangram 3.3.2	51.59%	56.88%	19.27%	40.60%
	Pangram 4	**92.41%**	**99.56%**	**85.43%**	**6.80%**
8	Pangram 3.3.2	55.08%	56.21%	49.42%	31.80%
	Pangram 4	**95.94%**	**99.61%**	**92.31%**	**3.51%**
12	Pangram 3.3.2	68.31%	65.67%	71.30%	25.39%
	Pangram 4	**97.12%**	**99.66%**	**94.33%**	**2.58%**
16	Pangram 3.3.2	81.89%	76.35%	85.85%	16.68%
	Pangram 4	**97.50%**	**99.63%**	**94.73%**	**2.20%**
20	Pangram 3.3.2	88.17%	82.02%	93.37%	11.76%
	Pangram 4	**98.18%**	**99.51%**	**96.31%**	**1.52%**
Overall	Pangram 3.3.2	64.18%	68.33%	52.61%	28.60%
	Pangram 4	**92.02%**	**98.35%**	**85.45%**	**5.87%**

Table 7. Comparison of Pangram 3.3.2 with adaptive boundaries and **Pangram 4**. Bold values indicate the better result. Precision and recall treat AI text as the positive class.

5.6 Performance on Languages Other than English

Table 8 reports the FPR and FNR for supported languages. FPRs are measured on pre-2022 per-language subsets of FineWeb2 (Penedo et al., 2025). While not reported, we test all 104 languages reported in FineWeb2, finding only 14 false positives among 996,273 examples, for an **overall multilingual FPR of 0.0014%** with a 95% Wilson confidence interval of (0.0008%,

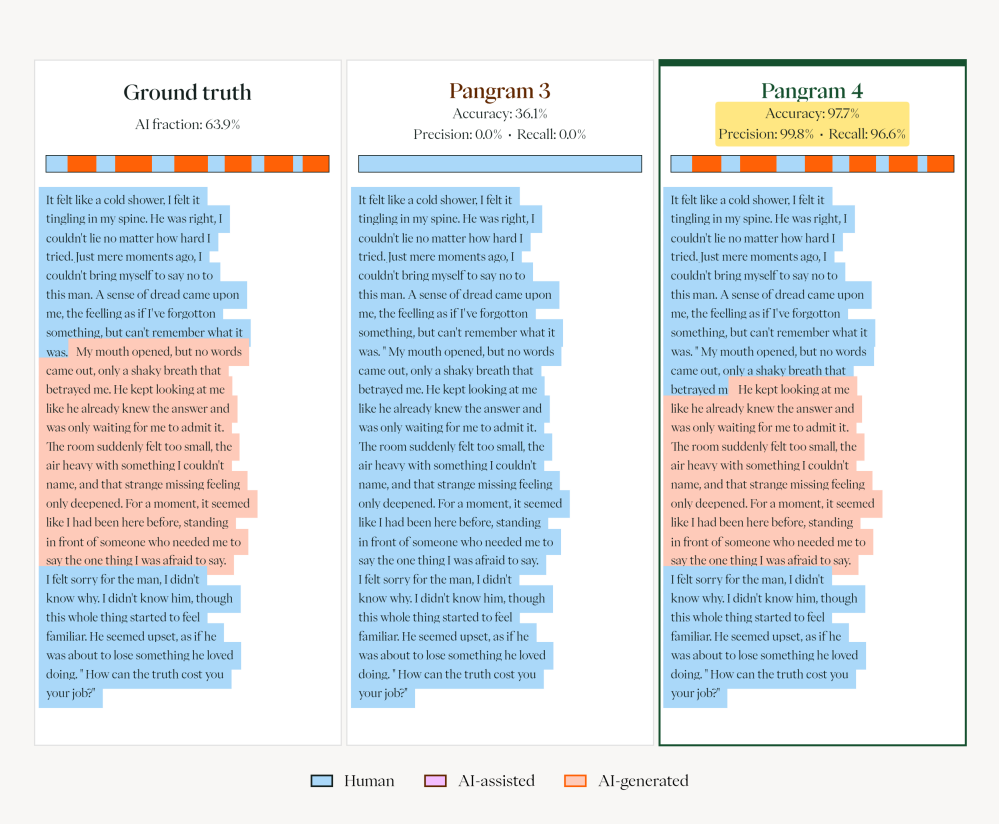

Figure 5. Pangram 3.3.2 vs. 4 on an excerpt from a heterogeneous mixed document. **Pangram 4** is able to detect AI at a much finer-grained resolution.

0.0024%). FNRs are measured across 18 supported languages using a test set of 190,149 multilingual AI generations, which are synthetic mirrors of English data, including how-to articles, news articles, and reviews. **Pangram 4**'s **overall multilingual FNR is 1.24%** with a 95% Wilson confidence interval of (1.19%, 1.29%). We hypothesize that higher false negative rates in languages such as Urdu (5.31%) and Persian (3.17%) may originate from differences in the base model's tokenization; however, the effects of tokenization on multilingual AI-text detection remain unknown.

5.7 Performance on Non-native English

Datasets. There has been speculation and anxiety that AI detectors are biased against non-native English speakers (Liang et al., 2023). To address this concern, we evaluate **Pangram 4**'s false positive rate on several public corpora of non-native English text: ELLIPSE (Crossley et al., 2023), ICNALE (Ishikawa, 2019) (a dialogue dataset), PELIC (Juffs et al., 2020), and the TOEFL dataset by Liang et al. (2023). In Table 9, we show that **Pangram 4** has a single raw false positive out of 24,586 samples, resulting in a 0.0041% FPR that is the same as our overall FPR reported in Section 5.3.

Language	FPR	FNR	Language	FPR	FNR
Arabic	0.0000%	0.9764%	Persian	0.0000%	3.1715%
Chinese	0.0000%	0.7133%	Polish	0.0000%	—
Czech	0.0000%	0.2760%	Portuguese	0.0078%	0.9946%
Dutch	0.0026%	0.8397%	Romanian	0.0000%	—
French	0.0026%	1.7839%	Russian	0.0052%	0.4850%
German	0.0026%	1.3258%	Spanish	0.0026%	0.8867%
Greek	0.0000%	—	Swedish	0.0000%	—
Hindi	0.0000%	1.3682%	Turkish	0.0000%	1.1066%
Hungarian	0.0000%	—	Ukrainian	0.0361%	1.5214%
Italian	0.0000%	0.3436%	Urdu	0.0000%	5.3169%
Japanese	0.0000%	0.9305%	Vietnamese	0.0026%	1.9653%
Korean	0.0000%	0.5805%			

Table 8. Multilingual false positive and false negative rates by language.

		Pangram 4		Pangram 3.3.2	
Dataset	N	FP	FPR (%)	FP	FPR (%)
ELLIPSE	3,899	0	0.000%	0	0.000%
ICNALE	5,593	0	0.000%	0	0.000%
PELIC	15,005	1	0.0067%	2	0.0133%
Liang TOEFL	89	0	0.000%	0	0.000%
Overall	**24,586**	**1**	**0.0041%**	**2**	**0.0081%**

Table 9. False positive rates on human-authored English learner writing and speech transcripts.

5.8 Public Benchmarks

Benchmark	AUROC / AUC	TPR at fixed FPR	FPR	FNR	Mean accuracy	F1
Binary						
UChicago (Jabarian and Imas, 2025)	✓	✓	✓			
VUB (Van Vlasselaer et al., 2026)						
GEDE (Gehring and Paaßen, 2025)	✓	✓				
Perkins et al. (Perkins et al., 2024)					✓	
Epoch AI Style Imitation (Lee, 2026)			✓	✓		
DetectRL (Wu et al., 2024)	✓					✓
MELD-eval (Li et al., 2026)	✓	✓				
Sem-Detect (Duarte et al., 2026)	✓	✓				
Mixed						
Saha Peer Review (Saha et al., 2026)		✓	✓			
OpAI-Bench (Bsharat et al., 2026)						

Table 10. Metric coverage across the binary and mixed-authorship evaluations. A checkmark means that Section 5 reports that metric for **Pangram 4**.

Summary of results. Table 12 compares **Pangram 4**, Pangram 3, and the strongest non-Pangram comparator on every split reported in the detailed benchmark tables. Comparisons between benchmarks are challenging, as there is no unified metric of success; some choose FNR/FPR, while others look at AUROC or other locally defined metrics. A summary of what metrics are presented in each benchmark is shown in Table 10.

Benchmark	Evaluation split	Pangram 4 result	Pangram 3 result	Next-best result
Binary				
UChicago	Standard, full length	TPR@1%FPR: 100.00%	99.86%	GPTZero: 98.64%
	Standard, under 50 words	TPR@1%FPR: 99.70%	88.87%	GPTZero: 0.00%
	Humanizer, full length	TPR@1%FPR: 98.93%	98.08%	GPTZero: 44.32%
	Humanizer, under 50 words	TPR@1%FPR: 73.32%	61.74%	GPTZero: 0.00%
	Human controls	FPR: 0.00%	0.00%	Originality: 0.11%
VUB	Fully AI-generated	Recall: 100%	97.4%	GPTZero: 0%
GEDE	Overall	TPR@1%FPR: 100.0%	98.9%	GPTZero: 77.6%
	Fully generated	TPR@1%FPR: 100.0%	100.0%	GPTZero: 93.2%
	AI-improved human	TPR@1%FPR: 100.0%	97.4%	GPTZero: 54.7%
	Humanized	TPR@1%FPR: 100.0%	100.0%	GPTZero: 92.6%
Perkins et al.	Baseline AI	Mean accuracy: 100.0%	—	Copyleaks: 73.9%
	Manipulated AI	Mean accuracy: 94.1%	—	Copyleaks: 58.7%
Epoch AI Style Imitation	Overall	FNR / FPR: 2.86% / 0.00%	5.05% / 0.00%	GPTZero: 5.72% / 0.00%
DetectRL	Multi-domain	F1: 96.93	98.05	Rob-Base: 99.75
	Multi-LLM	F1: 97.25	98.39	Rob-Base: 99.58
	Multi-attack	F1: 96.00	97.53	Rob-Large: 99.03
	Domain generalization	F1: 96.95	98.05	Rob-Base: 83.00
	LLM generalization	F1: 97.25	98.39	X-Rob-Base: 92.73
	Attack generalization	F1: 95.99	97.53	Rob-Base: 92.37
	Time, test	F1: 91.30	93.73	Rob-Large: 85.23
	Human writing	F1: 90.77	91.95	Rob-Base: 97.34 / Rob-Large: 94.63
	Overall	Average F1: 95.30	96.70	Rob-Base: 93.02
MELD-eval	GPT-5.4-Mini	TPR@1%FPR: 99.989%	99.962%	MELD: 100.0%
	Gemini-3-Flash	TPR@1%FPR: 99.991%	99.931%	MELD: 99.7%
	Claude-Haiku-4.5	TPR@1%FPR: 99.989%	99.949%	MELD: 100.0%
	Qwen-3.6-Plus	TPR@1%FPR: 99.993%	99.985%	MELD: 99.9%
	Generator overall	TPR@1%FPR: 99.990%	99.957%	MELD: 99.9%
	No attack	TPR@1%FPR: 100.00%	99.99%	—
	Homoglyph	TPR@1%FPR: 99.97%	99.99%	—
	Number perturbation	TPR@1%FPR: 100.00%	99.98%	—
	Synonym	TPR@1%FPR: 99.96%	99.72%	—
	Upper–lower flip	TPR@1%FPR: 99.99%	99.99%	—
	Whitespace	TPR@1%FPR:99.99%	99.99%	—
	Zero-width space	TPR@1%FPR: 100.00%	99.99%	—
	Attack overall	TPR@1%FPR: 99.99%	99.95%	—
Sem-Detect	Binary detection	TPR@0.1%FPR : 95.5%	100.0%	Sem-Detect: 76.0%
Mixed				
Saha Peer Review	Easy, AI-BP	TPR: 98.2%	100.0%	Fast-DetectGPT: 100.0%
	Easy, AI-EP	TPR: 100.0%	100.0%	Fast-DetectGPT: 100.0%
	Easy, AI-HI	TPR: 96.3%	100.0%	Fast-DetectGPT + Context: 99.3%
	Easy, H-AI	FPR: 1.4%	14.9%	Binoculars: 0.0%
	Hard, AI-BP	TPR: 100.0%	100.0%	GPTZero: 96.0%
	Hard, AI-EP	TPR: 100.0%	100.0%	GPTZero: 95.8%
	Hard, AI-HI	TPR: 98.9%	100.0%	GPTZero: 89.4%

Continued on next page

Benchmark	Evaluation split	Pangram 4 result	Pangram 3 result	Next-best result
	Hard, H-AI	FPR: 2.5%	4.5%	LogLikelihood: 0.0%
	Human	FPR: 0.0%	0.0%	LogLikelihood: 0.0%
OpAI-Bench	$v0$, none (target: 0.00%)	AI+Assisted mean: 0.00%	0.26%	—
	$v1$, polish (target: 20.75%)	AI+Assisted mean: 0.01%	0.41%	—
	$v2$, paraphrase (target: 30.95%)	AI+Assisted mean: 3.18%	1.84%	—
	$v3$, style rewrite (target: 46.47%)	AI+Assisted mean: 13.59%	4.60%	—
	$v4$, compress (target: 49.61%)	AI+Assisted mean: 5.48%	3.70%	—
	$v5$, expand (target: 68.58%)	AI+Assisted mean: 32.57%	12.38%	—
	$v6$, style rewrite (target: 81.62%)	AI+Assisted mean: 56.17%	20.73%	—
	$v7$, paraphrase (target: 92.57%)	AI+Assisted mean: 67.94%	22.91%	—
	$v8$, polish (target: 99.35%)	AI+Assisted mean: 69.50%	25.07%	—

Table 12. Results on the native metrics reported for each public benchmark. The next-best entry is the strongest non-Pangram result available for each metric. Full comparisons appear in §C.

UChicago. Jabarian and Imas (2025) pairs authentic human writing from several genres with outputs from multiple LLMs across a range of document lengths. It also includes dedicated evaluations of ultra-short passages and texts processed by commercial humanizers. Full detector comparisons and default-threshold false positive rates are reported in Table 18 and Table 19.

VUB. Van Vlasselaer et al. (2026) focuses on long-form academic papers and evaluates fully human-written, fully AI-generated, hybrid, and humanized AI-generated documents. Because the synthetic documents have known ground-truth AI proportions, the benchmark can test whether detector scores reflect the extent of AI involvement rather than merely provide binary classifications. Our evaluation uses the public fully AI-generated subset, whose score distribution is reported in Table 20.

GEDE. Gehring and Paaßen (2025) is an education-focused benchmark containing more than 900 student-written essays and over 12,500 LLM-generated essays. Its defining feature is a spectrum of student-contribution levels, ranging from fully human writing through light LLM improvement and full generation to adversarial humanization. Results for every evaluated split are reported in Table 21.

Perkins et al.. Perkins et al. (2024) evaluates 15 original AI-generated samples and 89 variants created using six detector-evasion techniques. It emphasizes adversarial robustness and the risks that unreliable detection poses to fairness and inclusivity in academic assessment. Baseline and manipulated-text results are reported in Table 22.

Epoch AI Style Imitation. Lee (2026) contains 495 human passages and 594 AI passages: 297 generated from basic prompts and 297 prompted to imitate the styles of sampled authors. We

report document-level false negative rates across both AI conditions, span-level false negative rates for the Pangram models, and false positive rates on the human controls in Table 23.

DetectRL. Wu et al. (2024) evaluates detector robustness and generalization across domains, generators, attack strategies, text lengths, and real-world factors in human writing. Task 1 measures in-domain performance across domains, generators, and attack types. Task 4 applies paraphrasing, perturbation, and data mixing to human-written texts to test how these transformations affect detector behavior. The full zero-shot and supervised leaderboard is reported in Table 24.

MELD-eval. Li et al. (2026) introduces a controlled generator-shift benchmark built from four current-generation chat models and eight English domains. It contains paired human texts, clean AI outputs, and attack-augmented AI outputs, and it tests zero-shot transfer to held-out generators at low false positive rates. Generator-level and attack-level results are reported in Table 25 and Table 26, respectively.

Sem-Detect. Duarte et al. (2026) introduces a dataset of more than 20,000 human-written, AI-generated, and LLM-refined peer reviews from ICLR and NeurIPS, together with a detector that combines textual features with claim-level semantic analysis. The benchmark tests whether detectors can distinguish fully AI-generated reviews from authentic or LLM-refined human reviews whose original judgments are preserved. Binary detection results are reported in Table 27.

Saha Peer Review. Saha et al. (2026) simulates several levels of human–AI collaboration, including fully AI-generated reviews, AI-generated reviews conditioned on human-provided key points, AI-polished human reviews, and unmodified human reviews. Its easy and hard subsets vary the diversity of generating models and prompts. The full cohort-level comparison is reported in Table 28.

OpAI-Bench. Bsharat et al. (2026) constructs nine successive versions of each human-written document using five AI editing operations and increasing predefined levels of AI coverage. Because it provides character-level authorship provenance, we compare the predicted AI+Assisted fraction directly with the ground-truth AI character fraction throughout the editing trajectory. Stage-level results are reported in Table 29.

5.9 Adversarial Robustness

We evaluate the robustness of **Pangram 4** to both humanizers and broader red-teaming attacks.

5.9.1 Robustness to Humanizer Attacks

We focus our evaluation of humanization on three domains in particular. These domains are a manually collected set of commercial humanizers chosen by top web traffic in early 2026; a popular GitHub repository containing instructions for AI agents to humanize their text; and an academic benchmark of AI texts attacked in specific manners.

How well does Pangram do at detecting humanized text?. We find Pangram is robust to humanization attempts. We detect humanized text as AI-generated 97.67% of the time and as either Mixed or AI-generated 98.83% of the time.

How well does Pangram distinguish between AI-generated and humanized text?. We evaluate the humanizer classifier on paired unmodified and humanized AI-generated texts. For this evaluation, humanized AI is the positive class and the corresponding unmodified AI text is the negative class. The humanizer classifier is only run if AI use is detected. Table 13 reports aggregate performance. For this separate AI-versus-humanized-AI evaluation, the auxiliary head's per-system TPR ranges from 91.52% to 99.39% across commercial humanizers, as detailed alongside the primary-detector results in Table 14.

Accuracy	F1	Precision	Recall	FPR	FNR
96.82%	0.9678	98.02%	95.57%	1.93%	4.43%

Table 13. Auxiliary humanizer-head performance on paired unmodified and humanized AI-generated texts. A false negative is a humanized text predicted as AI-generated. A false positive is AI-generated text predicted as humanized.

BLADER. BLADER[4] is a popular GitHub repository that contains instructions for LLMs on how to "remove signs of AI-generated writing from text." As of publication, BLADER has nearly 32,000 stars on GitHub. Here, we report our false negative rate on text humanized using three frontier models prompted with BLADER's instructions.

5.9.2 Red-Teaming

Manual Red-Teaming. One week before launching **Pangram 4**, we gave all Pangram employees and a small group of early testers access to Pangram to stress-test it for false positives and false negatives. We used the feedback from the early access group to refine our calibration and postprocessing procedure and make small final adjustments to the model.

Automated Red-Teaming with AI Agents. We also gave two Codex agents with GPT-5.6 Sol access to the **Pangram 4** API for 24 hours and put them on "Goal" mode with Full Access to attempt to fool **Pangram 4**. One agent was tasked with finding false positive gaps in our evaluation: kinds of text that are not explicitly present in our evaluation sets that could systematically have an elevated false positive rate. The false positive search covered a wide variety of historical and contemporary literature, official and legal records, patents and filings, public minutes, engineering and incident records, medical reports, field notes, learner writing, and interviews. Zero new false positives were found by the false positive red-teaming agent in 24 hours.

The second agent was tasked with finding systematic bypasses that could create false negatives repeatedly. This agent tested several potential bypass hypotheses, such as imitating specific human voices and styles, using different operational and professional registers, imitating a notetaker, using much terser vocabulary, conditioning its output on detailed source writing, aggregating multiple AI documents together, reformatting data as PDFs, putting the text through multiple rounds of translation, and more. The only bypass explicitly found by the agent was acting as a surgeon/pathologist dictating notes aloud. We examined these false negatives by eye and deemed them to be out of scope: more context was given to the LLM than the output, and the outputs more resembled extremely terse factual bullets rather than open-ended prose.

From these two experiments, we deemed the model resilient to agent-based adversarial attacks and sufficiently robust to release to the public.

[4] `blader/humanizer` on GitHub

	Pangram 4 on humanized AI text		Auxiliary head
Humanizer	AI recall	AI or Mixed recall	TPR
Commercial A	98.49%	99.40%	95.47%
Commercial B	97.57%	99.09%	91.79%
Commercial C	98.78%	98.78%	98.78%
Commercial D	98.48%	99.39%	94.83%
Commercial E	97.84%	98.77%	97.84%
Commercial F	99.70%	100.00%	91.52%
Commercial G	99.09%	99.39%	99.39%
Commercial H	92.78%	95.28%	92.22%
Commercial I	96.96%	98.78%	96.66%
Commercial J	97.32%	98.66%	95.30%
Commercial K	98.78%	99.39%	98.48%
Commercial L	99.33%	100.00%	92.00%
Commercial M	95.87%	98.73%	96.51%
Overall commercial	**97.69%**	**98.83%**	**95.57%**

Table 14. Performance of **Pangram 4** against different commercial evasion systems. The first two columns report TPRs for humanized text from the Pangram AI-use detector. The rightmost column is the auxiliary humanizer head's true-positive rate.

Humanization model	Documents	False negatives, n (%)
GPT-5.5	3,404	17 (0.499%)
Opus 4.8	3,412	14 (0.410%)
Sonnet 4.6	3,407	13 (0.382%)
Overall	**10,223**	**44 (0.430%)**

Table 15. False negative rates on humanized AI-generated text, grouped by the model used for humanization.

6 Conclusion

We introduce our most powerful AI detector model, **Pangram 4**, with the purpose of detecting more fine-grained AI usage and increasing both recall and specificity over Pangram 3.3.2. **Pangram 4** is the first AI-detection model that can predict not only document-level AI usage, but also homogeneous and heterogeneous mixed text. We intend for **Pangram 4** to increase visibility into how an author crafted a text, while still managing to keep our overall FPR at 0.0041%.

Limitations. While **Pangram 4** is a state-of-the-art AI detection model, its estimates are statistical, and both false positives and false negatives, although rare, do occur. Its predictions are also to some degree a black box, and the reasons it makes particular predictions are difficult to understand. Predictions for the same text in different contexts may also be inconsistent. For example, a section of a paper run in isolation may receive a different prediction than it does within the full context of the paper. In such cases, the additional context may give the model more information with which to make a holistic decision. We are working to reduce these inconsistencies over time. Finally, **Pangram 4** does not account for people who intentionally write like LLMs or who have absorbed elements of LLM style into their writing. This form of data drift is a major focus of our ongoing research.

Looking forward, we hope to work on answering questions related to data drift, consistency, and interpretability, and to work toward the overall goal of understanding the provenance of AI-generated text and how it influences the world around us.

AI Disclosure

We use AI agents extensively for coming up with research ideas, processing data, writing code, running experiments, and evaluating results. We use some AI assistance in the writing of the technical report, for proofreading and review, but the majority of the writing is ours. We claim full responsibility for the factual accuracy of the claims we make in this report. For full disclosure, we provide a link to the **Pangram 4** result on this technical report, and agree with its assessment of our authorship.[5]

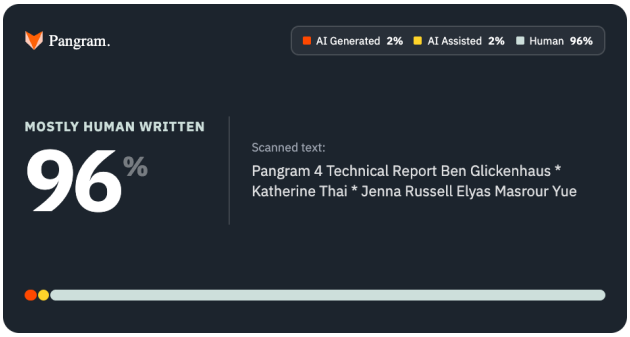

Figure 6. Pangram AI result for this technical report.

Acknowledgements

We would like to thank the following people for their contributions to **Pangram 4**. We thank Artem Frenk for early discussions regarding tokenwise postprocessing and the HMM/CRF formulation. We thank Fanyi Pan, Lu Lyu, Anna Lin, and Tianxu Zhou for **Pangram 4** product support, design, UI/UX engineering, and testing to bring **Pangram 4** to our users. We thank Jocelyn James for graphic design contributions. We thank Annamieka Aerts and James Freedman for helping with the technical writing and proofreading. We thank Levi Goldstein and Lauryl Salami for data expertise and contributions. We also thank our early testers for providing invaluable pre-release feedback on the model.

References

Anthropic. Claude haiku 4.5, October 2025. URL https://www.anthropic.com/news/claude-haiku-4-5. Released October 15, 2025.

Anthropic. Claude fable 5 and mythos 5, June 2026a. URL https://www.anthropic.com/news/claude-fable-5-mythos-5. Released June 9, 2026.

[5] https://www.pangram.com/history/6c43abfc-803e-4dd5-96d1-8eba2768ba39

Anthropic. Claude opus 4.8, May 2026b. URL https://www.anthropic.com/news/claude-opus-4-8. Released May 28, 2026.

Anthropic. Claude opus 5. Large language model, July 2026c. URL https://www.anthropic.com/news/claude-opus-5.

Anthropic. Claude sonnet 4.6, February 2026d. URL https://www.anthropic.com/news/claude-sonnet-4-6. Released February 17, 2026.

Anthropic. Claude sonnet 5, June 2026e. URL https://www.anthropic.com/news/claude-sonnet-5. Released June 30, 2026.

Ekaterina Artemova, Jason S Lucas, Saranya Venkatraman, Jooyoung Lee, Sergei Tilga, Adaku Uchendu, and Vladislav Mikhailov. Beemo: Benchmark of expert-edited machine-generated outputs. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6992–7018, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.357. URL https://aclanthology.org/2025.naacl-long.357/.

Sondos Mahmoud Bsharat, Jiacheng Liu, Xiaohan Zhao, Tianjun Yao, Xinyi Shang, Yi Tang, Jiacheng Cui, Ahmed Elhagry, Salwa K. Al Khatib, Hao Li, Salman Khan, and Zhiqiang Shen. Operation-guided progressive human-to-ai text transformation benchmark for multi-granularity ai-text detection, 2026. URL https://arxiv.org/abs/2606.06481.

Scott Crossley, Yu Tian, Perpetual Baffour, Alex Franklin, Youngmeen Kim, Wesley Morris, Meg Benner, Aigner Picou, and Ulrich Boser. The English Language Learner Insight, Proficiency and Skills Evaluation (ELLIPSE) Corpus. *International Journal of Learner Corpus Research*, 9(2): 248–269, 2023. doi: 10.1075/ijlcr.22026.cro. Pre-print available at https://doi.org/10.5281/zenodo.11217937.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Jonas Dolezal, Sawood Alam, Mark Graham, and Maty Bohacek. The impact of ai-generated text on the internet, 2026. URL https://arxiv.org/abs/2604.26965.

George Drayson, Emine Yilmaz, and Vasileios Lampos. Machine-generated text detection prevents language model collapse. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29657–29673, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1506. URL https://aclanthology.org/2025.emnlp-main.1506/.

Andre Vicente Duarte, Brian Tufts, Aditya Oke, Fei Fang, Arlindo L. Oliveira, and Lei Li. Sem-detect: Semantic level detection of AI generated peer-reviews. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=3S9eVockjL.

Nils Dycke, Marina Sakharova, Nico Daheim, and Iryna Gurevych. 'your ai text is not mine': Redefining and evaluating ai-generated text detection under realistic assumptions, 2026. URL https://arxiv.org/abs/2606.04906.

Bradley Emi and Max Spero. Technical report on the pangram ai-generated text classifier, 2024. URL https://arxiv.org/abs/2402.14873.

Lukas Gehring and Benjamin Paaßen. Assessing LLM text detection in educational contexts: Does human contribution affect detection?, 2025. URL https://arxiv.org/abs/2508.08096.

Google. Gemini 3.1 pro: A smarter model for your most complex tasks, February 2026a. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-1-pro/. Released in preview February 19, 2026.

Google. Gemini 3.5: Frontier intelligence with action, May 2026b. URL https://blog.google/innovation-and-ai/models-and-research/gemini-models/gemini-3-5/. Gemini 3.5 Flash released May 19, 2026.

Google. Gemma 4: Our most capable open models to date, April 2026c. URL https://blog.google/innovation-and-ai/technology/developers-tools/gemma-4/. Released April 2, 2026.

Shin'ichiro Ishikawa. The ICNALE Spoken Dialogue: A new dataset for the study of Asian learners' performance in L2 English interviews. *English Teaching*, 74(4):153–177, 2019. doi: 10.15858/engtea.74.4.201912.153. URL https://files.eric.ed.gov/fulltext/EJ1284816.pdf.

Brian Jabarian and Alex Imas. Artificial writing and automated detection. NBER Working Paper 34223, National Bureau of Economic Research, September 2025. URL https://www.nber.org/papers/w34223.

Alan Juffs, Na-Rae Han, and Ben Naismith. The University of Pittsburgh English Language Institute Corpus (PELIC). [Data set], 2020. URL https://github.com/ELI-Data-Mining-Group/PELIC-dataset.

Jaeho Lee. Ai detectors rarely flag human writing, but sometimes miss ai text imitating real authors, 2026. URL https://epoch.ai/data-insights/ai-detectors-false-negatives. Accessed: 2026-07-28.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Prompt repetition improves non-reasoning llms, 2025. URL https://arxiv.org/abs/2512.14982.

Chenjun Li, Cheng Wan, and Johannes C. Paetzold. Meld: Multi-task equilibrated learning detector for ai-generated text, 2026. URL https://arxiv.org/abs/2605.06903.

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

Elyas Masrour, Bradley Emi, and Max Spero. Damage: Detecting adversarially modified ai generated text, 2025. URL https://arxiv.org/abs/2501.03437.

Meta. Llama 3.3 70B Instruct model card, December 2024. URL https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct. Released December 6, 2024.

Mistral AI. Mistral small 3. https://mistral.ai/news/mistral-small-3/, 2025.

Mistral AI. Mistral medium 3.5 model card, April 2026. URL https://docs.mistral.ai/models/model-cards/mistral-medium-3-5-26-04. Released April 28, 2026.

Moonshot AI. Kimi research, April 2026. URL https://www.kimi.com/blog/. Kimi K2.6 released April 20, 2026.

Jun-Ping Ng and Viktoria Abrecht. Better summarization evaluation with word embeddings for ROUGE. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1925–1930, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1222. URL https://aclanthology.org/D15-1222/.

NVIDIA. NVIDIA Nemotron 3 Ultra powers faster, more efficient reasoning for long-running agents, June 2026. URL https://developer.nvidia.com/blog/nvidia-nemotron-3-ultra-powers-faster-more-efficient-reasoning-for-long-running-agents/. Released June 4, 2026.

OpenAI. Introducing gpt-oss, August 2025. URL https://openai.com/index/introducing-gpt-oss/. Released August 5, 2025.

OpenAI. Introducing GPT-5.4, March 2026a. URL https://openai.com/index/introducing-gpt-5-4/. Released March 5, 2026.

OpenAI. Introducing GPT-5.5, April 2026b. URL https://openai.com/index/introducing-gpt-5-5/. Released April 23, 2026.

OpenAI. Introducing GPT-5.6, July 2026c. URL https://openai.com/index/gpt-5-6/. Generally available July 9, 2026.

Pangram Labs. Pangram predicts 21% of ICLR reviews are ai-generated, November 2025. URL https://www.pangram.com/blog/pangram-predicts-21-of-iclr-reviews-are-ai-generated.

Pangram Labs. Ai content is everywhere on social media, especially linkedin, July 2026. URL https://www.pangram.com/blog/ai-in-your-feed.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. Fineweb2: One pipeline to scale them all — adapting pre-training data processing to every language. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=jnRBe6zatP.

Mike Perkins, Jasper Roe, Binh H. Vu, Darius Postma, Don Hickerson, James McGaughran, and Huy Q. Khuat. Simple techniques to bypass GenAI text detectors: Implications for inclusive education. *International Journal of Educational Technology in Higher Education*, 21(1):53, 2024. doi: 10.1186/s41239-024-00487-w. URL https://doi.org/10.1186/s41239-024-00487-w.

Gerrit Quaremba, Elizabeth Black, Denny Vrandečić, and Elena Simperl. TSM-Bench: Detecting LLM-generated text in real-world wikipedia editing practices, 2026. URL https://arxiv.org/abs/2605.31113.

Qwen. Model releases, May 2026. URL https://docs.qwencloud.com/changelog/models. Qwen 3.7 Max released May 21, 2026.

Manav Raj, Justin M. Berg, and Rob Seamans. The artificial intelligence disclosure penalty: Humans persistently devalue AI-generated creative writing. *Journal of Experimental Psychology: General*, 155(4):896–915, April 2026. doi: 10.1037/xge0001889. URL https://doi.org/10.1037/xge0001889. Epub January 8, 2026. PMID: 41505277.

Jenna Russell, Marzena Karpinska, and Mohit Iyyer. People who frequently use ChatGPT for writing tasks are accurate and robust detectors of AI-generated text. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5342–5373, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.267. URL https://aclanthology.org/2025.acl-long.267/.

Jenna Russell, Marzena Karpinska, Destiny Akinode, James Zhou, Katherine Thai, Bradley Emi, Max Spero, and Mohit Iyyer. AI use in American newspapers is widespread, uneven, and rarely disclosed. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14554–14580, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.663. URL https://aclanthology.org/2026.acl-long.663/.

Jenna Russell, Rishanth Rajendhran, Chau Minh Pham, Mohit Iyyer, and John Wieting. Storyscope: Investigating idiosyncrasies in ai fiction, 2026b. URL https://arxiv.org/abs/2604.03136.

Rounak Saha, Gurusha Juneja, Dayita Chaudhuri, Naveeja Sajeevan, Nihar B Shah, and Danish Pruthi. Policies permitting LLM use for polishing peer reviews are currently not enforceable. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=znFIR7WFGt.

Shoumik Saha and Soheil Feizi. Almost AI, almost human: The challenge of detecting AI-polished writing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25414–25431. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1303. URL https://aclanthology.org/2025.findings-acl.1303/.

Chantal Shaib, Tuhin Chakrabarty, Diego Garcia-Olano, and Byron C. Wallace. Measuring ai "slop" in text, 2026. URL https://arxiv.org/abs/2509.19163.

Tencent. Tencent hunyuan officially releases Hy3, advancing agent capabilities and deeper product integration, July 2026. URL https://www.tencent.com/tencent-hunyuan-officially-releases-hy3-advancing-agent-capabilities-and-deeper-product-inte Reports the Hy3 Preview launch on April 23, 2026.

Katherine Thai, Bradley Emi, Elyas Masrour, and Mohit Iyyer. Editlens: Quantifying the extent of AI editing in text. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=gOkitaPCfZ.

Thinking Machines Lab. Inkling: Our open-weights model, July 2026. URL https://thinkingmachines.ai/news/introducing-inkling/. Released July 15, 2026.

Marijke Van Vlasselaer, Filip Van Droogenbroeck, and Bram Spruyt. Who wrote this? evaluating the reliability of AI detection tools in higher education. *International Journal for Educational Integrity*, 22:16, 2026. doi: 10.1007/s40979-026-00226-w. URL https://doi.org/10.1007/s40979-026-00226-w.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. DetectRL: Benchmarking LLM-generated text detection in real-world scenarios. In *The*

Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2024. URL https://openreview.net/forum?id=ZGMkOikEyv.

xAI. Inference API: Models, April 2026. URL https://docs.x.ai/developers/rest-api-reference/inference/models. The Grok 4.3 endpoint reports creation time 1776556800, corresponding to April 19, 2026.

Hongyeon Yu, Dongchan Kim, and Young-Bum Kim. Retrieval collapses when ai pollutes the web. In *Proceedings of the ACM Web Conference 2026*, WWW '26, page 8745–8748, New York, NY, USA, 2026. Association for Computing Machinery. ISBN 9798400723070. doi: 10.1145/3774904.3792955. URL https://doi.org/10.1145/3774904.3792955.

Z.ai. GLM-5.2: Built for long-horizon tasks, June 2026. URL https://z.ai/blog/glm-5.2. Released June 16, 2026.

Qihui Zhang, Chujie Gao, Dongping Chen, Yue Huang, Yixin Huang, Zhenyang Sun, Shilin Zhang, Weiye Li, Zhengyan Fu, Yao Wan, and Lichao Sun. LLM-as-a-coauthor: Can mixed human-written and machine-generated text be detected? In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 409–436, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.29. URL https://aclanthology.org/2024.findings-naacl.29/.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m ChatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Bl8u7ZRlbM.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

A Citation reference

Citation

Please use the following BibTeX entry when citing this work:

```
@techreport{pangram2026pangram4,
  title={Pangram 4 Technical Report},
  author={Ben Glickenhaus and Katherine Thai and
    Jenna Russell and Elyas Masrour and Yue Han and
    Max Spero and Bradley Emi},
  institution={Pangram Labs},
  year={2026}
}
```

B Mixed Authorship Text Dataset

In tables 16 and 17, we provide the prompts and instructions used to create our AI Polish and AI Editing Prompts evaluation datasets.

Consumer product / LLM	**Editing instruction / prompt**
Apple Intelligence in Pages	Proofread
Grammarly	Fix any mistakes
Claude Opus 4.8	Fix spelling, punctuation, and clear grammar errors only.
Gemini 3.1 Pro Preview	Smooth awkward phrasing while preserving the author's sentence structure.
GPT-5.5	Make the writing slightly clearer without changing tone or vocabulary level.
	Correct minor academic style issues while keeping the draft recognizably human.
	Lightly polish for readability.
	Make minimal copyediting changes suitable for a final proofreading pass.
	Change approximately one tenth of the text.

Table 16. Editing instructions and prompts used to create our AI Polish evaluation set.

C Detailed Benchmark Results

This section expands upon the results in Table 12.

Consumer product / LLM	Editing instruction / prompt
Apple Intelligence	Make concise Rewrite
Gemini in Google Docs	Rephrase
Grammarly	Change writing style: Professional Change writing style: Academic Change writing style: Creative Sound fluent Improve it Simplify it Paraphrase it
Claude Opus 4.8 Gemini 3.1 Pro Preview GPT-5.5	Substantially rewrite for polished academic style while preserving meaning. Restructure sentences and paragraphs for stronger argument flow. Substantially transform the draft into publication-ready academic prose. Heavily revise wording, organization, and transitions without changing facts. Improve logic, emphasis, and readability through extensive rewriting. Produce a highly polished version that remains faithful to the source. Change approximately half of the text. Change approximately three quarters of the text.

Table 17. Editing instructions and prompts used to create our AI Editing Prompts evaluation set.

C.1 Binary

Split	Detector	N	AUROC	TPR @ 1% FPR
Standard, full length	**Pangram 4**	15,936	**0.999997**	**100.00%**
	Pangram 3	15,928	0.999884	99.86%
	Originality	14,613	0.998038	94.15%
	GPTZero	15,934	0.989643	98.64%
Standard, under 50 words	**Pangram 4**	1,312	**0.999888**	**99.70%**
	Pangram 3	1,311	0.995829	88.87%
	GPTZero	1,312	0.940549	0.00%
Humanizer, full length	**Pangram 4**	15,932	**0.999579**	**98.93%**
	Pangram 3	15,928	0.998566	98.08%
	Originality	10,812	0.962496	28.49%
	GPTZero	11,689	0.718367	44.32%
Humanizer, under 50 words	**Pangram 4**	1,312	0.981019	**73.32%**
	Pangram 3	1,312	**0.983775**	61.74%
	GPTZero	1,312	0.644036	0.00%

Table 18. Performance on the UChicago DetectionAI benchmark. TPR is measured at a 1% false positive rate.

Note: Originality is excluded from both under-50-word splits because it does not support texts of that length. N is the number of observations for which each detector returned a usable score. GPTZero and Originality sometimes rejected the requests.

Detector	Human N	False positives	FPR
Pangram 4	7,968	**0**	**0.00%**
Pangram 3	7,964	0	0.00%
Originality	7,308	8	0.11%
GPTZero	7,968	57	0.72%

Table 19. Overall false positive rates on the UChicago DetectionAI human controls at each detector's default 50% decision threshold.

Note: A document is classified as AI-generated when its detector score is at least 50%.

Detector	N	80–100%	60–80%	40–60%	20–40%	0–20%
Pangram 4	39	**39**	**0**	**0**	**0**	**0**
Pangram 3	39	25	13	0	1	0
GPTZero	39	0	0	0	12	27
Copyleaks	39	0	0	0	9	30
Turnitin	39	0	0	0	0	39

Table 20. Score distribution on the public fully AI-generated subset of the VUB benchmark.

Split	Detector	N	AUROC	TPR @ 1% FPR
Overall	**Pangram 4**	569	**1.000**	**100.0%**
	Pangram 3	569	0.999	98.9%
	GPTZero	569	0.979	77.6%
	RoBERTa	569	0.937	28.9%
	DetectGPT	542	0.917	33.7%
	Fast-DetectGPT	569	0.911	55.5%
	Ghostbuster	569	0.834	28.9%
	Intrinsic-Dim	569	0.554	0.0%
Fully generated	**Pangram 4**	285	**1.000**	**100.0%**
	Pangram 3	285	**1.000**	**100.0%**
	GPTZero	285	0.997	93.2%
	RoBERTa	285	0.994	54.7%
	Fast-DetectGPT	285	0.985	82.1%
	DetectGPT	269	0.984	69.1%
	Ghostbuster	285	0.960	52.1%
	Intrinsic-Dim	285	0.541	0.0%
AI-improved human	**Pangram 4**	285	**1.000**	**100.0%**
	Pangram 3	285	0.999	97.4%
	GPTZero	285	0.953	54.7%
	RoBERTa	285	0.888	11.1%
	DetectGPT	274	0.847	6.6%
	Fast-DetectGPT	285	0.797	10.5%
	Ghostbuster	285	0.680	7.4%
	Intrinsic-Dim	285	0.509	0.0%
Humanized	**Pangram 4**	189	**1.000**	**100.0%**
	Pangram 3	189	**1.000**	**100.0%**
	GPTZero	189	0.995	92.6%
	Fast-DetectGPT	189	0.993	92.6%
	DetectGPT	181	0.928	18.9%
	RoBERTa	189	0.918	12.8%
	Ghostbuster	189	0.893	25.5%
	Intrinsic-Dim	189	0.671	0.0%

Table 21. Performance on the GEDE public benchmark. TPR is measured at a maximum empirical false positive rate of 1%.

Detector	Baseline AI		Manipulated AI	
	N	Mean accuracy	N	Mean accuracy
Pangram 4	15	**100.0%**	89	**94.1%**
Copyleaks	15	73.9%	90	58.7%
GPTZero	15	26.4%	90	16.7%
Turnitin	15	50.0%	90	7.9%

Table 22. Performance on the Perkins benchmark using the paper's three-method mean accuracy.

Detector	Document-level FNR	Span-level FNR	FPR
Pangram 4	**2.86%** (17/594)	**1.44%**	0.00% (0/495)
Pangram 3.3.2	5.05% (30/594)	4.28%	0.00% (0/495)
GPTZero	5.72% (34/594)	—	0.00% (0/495)
Originality.ai	9.09% (54/594)	—	3.84% (19/495)

Table 23. Epoch AI Style Imitation benchmark.

Strict verdicts: For document-level results, only a clean "AI" verdict counts as a detection; "Mixed" and "AI-Assisted" verdicts count as non-detections (and as false positives on human text). Span-level results count any non-AI span as a false negative in proportion to its length.

		Multi-Domain		Multi-LLM		Multi-Attack		Generalization			Time		Human Writing		Avg.
Detector	**Evaluation Type**	**AUROC**	**F1**	**AUROC**	**F1**	**AUROC**	**F1**	**Domain F1**	**LLM F1**	**Attack F1**	**Train F1**	**Test F1**	**AUROC**	**F1**	**F1**
Pangram 4	Zero-shot	97.02	96.93	97.32	97.25	97.22	96.00	96.95	97.25	95.99	N/A	91.30	93.80	90.77	95.30†
Pangram 3.3.2	Zero-shot	99.75	98.05	99.85	98.39	99.72	97.53	**98.05**	**98.39**	**97.53**	N/A	**93.73**	95.67	91.95	**96.70**†
Open Pangram	Zero-shot	99.48	94.20	99.48	94.30	99.26	94.14	94.20	94.30	94.14	N/A	88.10	96.44	91.60	93.12†
Rob-Base	Supervised	**99.98**	**99.75**	**99.93**	**99.58**	99.56	97.66	83.00	91.81	92.37	79.99	74.00	**97.34**	94.31	93.02
Rob-Large	Supervised	99.78	98.87	95.16	90.03	**99.87**	**99.03**	77.20	82.85	83.96	**86.08**	85.23	96.68	**94.63**	91.49
X-Rob-Base	Supervised	99.92	99.34	99.14	98.17	98.49	96.07	75.97	92.73	90.58	84.25	73.83	93.43	90.29	91.71
X-Rob-Large	Supervised	99.01	97.44	97.40	93.47	99.31	97.75	76.14	85.89	73.42	86.35	79.83	97.21	94.43	90.59
Binoculars	Zero-shot	83.95	78.25	83.30	74.83	85.05	78.53	77.47	74.10	74.70	73.82	74.34	90.68	85.98	79.61
Revise-Detect.	Zero-shot	67.24	60.82	66.36	53.72	70.89	57.24	54.50	53.28	50.63	65.71	67.96	83.29	82.16	64.13
Log-Rank	Zero-shot	64.43	57.53	63.75	54.18	68.52	55.15	55.10	52.78	51.28	57.44	59.74	88.46	83.85	62.48
LRR	Zero-shot	65.47	55.45	64.93	53.01	68.53	57.99	54.61	52.73	57.41	57.09	58.15	85.99	80.56	62.46
Log-Likelihood	Zero-shot	63.71	56.36	62.97	53.13	67.97	54.38	53.37	51.77	50.73	57.92	59.28	88.48	83.75	61.83
DNA-GPT	Zero-shot	64.92	55.83	64.36	51.09	68.36	53.36	51.51	47.09	41.98	57.63	62.43	87.80	82.77	60.70
Fast-DetectGPT	Zero-shot	58.52	48.07	59.58	46.55	60.70	50.63	48.35	36.56	49.47	61.31	55.08	76.03	68.47	55.33
Rank	Zero-shot	51.34	44.97	50.33	42.06	57.08	48.83	42.61	41.49	38.84	41.67	46.65	83.86	80.00	51.52
NPR	Zero-shot	48.37	41.41	47.27	40.04	53.49	45.22	38.58	38.83	36.10	37.60	42.17	80.03	75.98	48.08
DetectGPT	Zero-shot	34.43	21.52	34.93	14.80	36.19	19.15	11.54	13.11	11.84	35.78	34.69	60.86	48.76	29.05
Entropy	Zero-shot	46.02	27.40	46.97	34.25	43.75	24.69	25.06	31.07	16.53	13.38	15.99	22.39	16.60	28.01

Table 24. Performance on the DetectRL benchmark. All values are reported as percentages.

Note: Baseline results are taken from the original DetectRL leaderboard. **Pangram 4**, Pangram 3.3.2, and Open Pangram are evaluated in a **zero-shot** setting without benchmark-specific training and remain competitive with or outperform the supervised methods. The average F1 is computed over the test datasets.

Detector	GPT-5.4-Mini	Gemini-3-Flash	Claude-Haiku-4.5	Qwen-3.6-Plus	Overall
Zero-shot detectors					
Pangram 4	99.989 ± 0.008	**99.991 ± 0.008**	99.989 ± 0.009	**99.993 ± 0.006**	**99.990 ± 0.004**
Pangram 3.3.2	99.962 ± 0.015	99.931 ± 0.022	99.949 ± 0.019	99.985 ± 0.010	99.957 ± 0.009
Open Pangram	57.2 ± 1.1	59.4 ± 1.0	65.1 ± 1.1	65.0 ± 1.2	61.7 ± 1.0
GLTR	1.2 ± 0.3	0.5 ± 0.3	1.3 ± 0.4	0.3 ± 0.1	0.8 ± 0.3
Fast-DetectGPT	9.2 ± 1.7	19.8 ± 2.1	14.3 ± 2.0	24.7 ± 2.3	17.0 ± 2.0
Binoculars	0.2 ± 0.1	0.6 ± 0.1	1.1 ± 0.2	0.7 ± 0.2	0.6 ± 0.1
Supervised detector baselines					
RoBERTa-Large (OpenAI)	1.1 ± 0.1	1.1 ± 0.1	1.8 ± 0.1	1.8 ± 0.1	1.4 ± 0.0
RoBERTa-ChatGPT	0.5 ± 0.1	1.2 ± 0.2	0.5 ± 0.2	0.9 ± 0.2	0.8 ± 0.2
RADAR	0.4 ± 0.2	1.4 ± 0.5	2.2 ± 0.8	1.2 ± 0.5	1.3 ± 0.5
Human-as-Outlier	2.7 ± 1.2	1.6 ± 0.9	0.8 ± 0.4	1.1 ± 0.8	1.6 ± 0.8
ModernBERT-Detect	97.2 ± 1.0	93.0 ± 2.2	98.2 ± 0.9	93.7 ± 2.0	95.5 ± 1.5
RepreGuard	23.0 ± 1.3	27.2 ± 1.4	41.2 ± 1.8	45.4 ± 2.0	34.2 ± 1.6
MELD	**100.0 ± 0.0**	99.7 ± 0.1	**100.0 ± 0.0**	99.9 ± 0.0	99.9 ± 0.0

Table 25. TPR@1%FPR (×100) on MELD-eval by generator.

Note: Results for all detectors other than Pangram 4, Pangram 3.3.2, and Open Pangram are taken from Table 4 of the MELD paper. Pangram 4 uses the derived AI likelihood $\text{fraction}_{\text{AI}} + 0.5\,(\text{fraction}_{\text{AI-assisted}})$. Confidence intervals for both Pangram models are 95% percentile-bootstrap intervals based on 5,000 resamples. The best and second-best results in each column are shown in bold and underlined, respectively.

Table 26. Pangram 4 and Pangram 3.3.2 performance on MELD-eval by attack type. AUROC and TPR values are reported as percentages.

Attack Type	Detector	AI N	AUROC	TPR@1%FPR	Detected AI	False Negatives
None (clean)	Pangram 3.3.2	31,448	99.99%	99.99%	31,447 / 31,448	1
None (clean)	Pangram 4	31,448	100.00%	100.00%	31,448 / 31,448	0
Homoglyph	Pangram 3.3.2	31,448	99.99%	99.99%	31,446 / 31,448	2
Homoglyph	Pangram 4	31,448	99.98%	99.97%	31,440 / 31,448	8
Number perturbation	Pangram 3.3.2	31,448	99.99%	99.98%	31,444 / 31,448	4
Number perturbation	Pangram 4	31,448	100.00%	100.00%	31,448 / 31,448	0
Synonym	Pangram 3.3.2	31,448	99.98%	99.72%	31,363 / 31,448	85
Synonym	Pangram 4	31,448	99.98%	99.96%	31,438 / 31,448	10
Upper–lower flip	Pangram 3.3.2	31,448	99.99%	99.99%	31,447 / 31,448	1
Upper–lower flip	Pangram 4	31,448	99.99%	99.99%	31,447 / 31,448	1
Whitespace	Pangram 3.3.2	31,448	99.99%	99.99%	31,447 / 31,448	1
Whitespace	Pangram 4	31,448	99.99%	99.99%	31,446 / 31,448	2
Zero-width space	Pangram 3.3.2	31,448	99.99%	99.99%	31,447 / 31,448	1
Zero-width space	Pangram 4	31,448	100.00%	100.00%	31,448 / 31,448	0
Overall	Pangram 3.3.2	220,136	99.99%	99.95%	220,041 / 220,136	95
Overall	Pangram 4	**220,136**	**99.99%**	**99.99%**	**220,115 / 220,136**	**21**

Note: Each attack is evaluated against the same 7,862 clean human controls. Thresholds are calibrated independently at a target FPR of 1%. Attacks are applied only to AI texts.

Detector	AUC (%) ↑	TPR@0.1%FPR (%) ↑	TPR@1%FPR (%) ↑
Pangram 4	97.8 ± 1.0	95.5 ± 1.0	95.5 ± 1.0
Pangram 3.3.2	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$
LogRank	57.6 ± 1.0	0.0 ± 0.0	0.1 ± 0.0
MAGE	69.9 ± 1.0	0.0 ± 0.0	0.8 ± 1.0
Fast-DetectGPT	69.9 ± 1.0	2.1 ± 1.0	6.2 ± 1.0
Binoculars	75.1 ± 1.0	0.8 ± 1.0	6.2 ± 2.0
TF Model†	92.6 ± 1.0	36.9 ± 6.0	55.3 ± 4.0
RADAR	96.5 ± 0.0	15.3 ± 5.0	37.1 ± 6.0
Anchor†	97.9 ± 0.0	54.1 ± 4.0	71.3 ± 7.0
EditLens†	$\underline{99.8 \pm 0.0}$	60.6 ± 16.0	$\underline{95.6 \pm 2.0}$
Sem-Detect†	99.9 ± 0.0	$\underline{76.0 \pm 11.0}$	97.3 ± 3.0

Table 27. Performance on the Sem-Detect peer-review benchmark. AUC and TPR values are reported as percentages.

Note: Baseline results are taken from Table 1 of the original paper. **Pangram 4** uses the derived AI likelihood, calculated as $\text{fraction}_{\text{AI}} + 0.5\,(\text{fraction}_{\text{AI-assisted}})$. Uncertainty is estimated using 1,000 bootstrap resamples. All uncertainty terms are reported in percentage points. Best and second-best results are shown in bold and underlined, respectively. Detectors marked with † are domain-specific models trained or tuned on peer-review data.

C.2 Mixed

Detector	Easy Subset				Hard Subset				Human
	TPR			FPR	TPR			FPR	FPR
	AI-BP	AI-EP	AI-HI	H-AI	AI-BP	AI-EP	AI-HI	H-AI	H
Pangram 4	98.2	**100.0**	96.3	1.4	**100.0**	**100.0**	98.9	2.5	**0.0**
Pangram 3.3.2	**100.0**	**100.0**	**100.0**	14.9	**100.0**	**100.0**	**100.0**	4.5	**0.0**
Pangram 3.0	**100.0**	**100.0**	**100.0**	3.0	97.0	99.3	92.6	3.1	**0.0**
GPTZero	96.7	93.3	91.0	3.0	96.0	95.8	89.4	3.4	1.0
LogLikelihood	97.8	96.4	72.4	0.4	46.0	40.9	30.5	**0.0**	**0.0**
LogLikelihood + Context	98.9	99.6	91.6	1.6	59.8	50.5	43.7	4.2	0.4
Fast-DetectGPT	**100.0**	**100.0**	97.5	3.5	72.1	68.2	63.1	4.6	0.2
Fast-DetectGPT + Context	**100.0**	**100.0**	99.3	9.0	75.3	73.9	68.2	9.5	0.9
Binoculars	51.1	50.7	39.6	**0.0**	21.6	22.3	14.1	0.1	**0.0**
Binoculars + Context	50.7	53.3	33.7	**0.0**	20.5	20.0	14.3	0.2	**0.0**

Table 28. Performance on the Saha peer-review benchmark. All values are percentages.

Note: Baseline results are taken from Table 2 of the original paper. For Pangram 3.3.2 and **Pangram 4**, AI-BP, AI-EP, and AI-HI are successfully detected when the prediction is either AI or Mixed. For H-AI, only an AI prediction is counted as a false positive; Mixed and Human predictions are accepted. For fully human reviews (H), both AI and Mixed predictions are counted as false positives. Pangram results are evaluated on the test partition of each subset.

Version	Operation	Actual AI character fraction (mean)	Pangram 4 Prediction (mean)	Pangram 4 Prediction (median)	Pangram 3.3.2 edit score (mean)	Pangram 3.3.2 edit score (median)
$v0$	None	0.0000	0.0000	0.0000	0.0026	0.0021
$v1$	Polish	0.2075	0.0001	0.0000	0.0041	0.0022
$v2$	Paraphrase	0.3095	0.0318	0.0000	0.0184	0.0030
$v3$	Style rewrite	0.4647	0.1359	0.0000	0.0460	0.0139
$v4$	Compress	0.4961	0.0548	0.0000	0.0370	0.0089
$v5$	Expand	0.6858	0.3257	0.2210	0.1238	0.0603
$v6$	Style rewrite	0.8162	0.5617	0.6751	0.2073	0.1241
$v7$	Paraphrase	0.9257	0.6794	0.8415	0.2291	0.1359
$v8$	Polish	0.9935	0.6950	0.8853	0.2507	0.1444

Table 29. Pangram 4 and Pangram 3.3.2 predictions across progressive human–AI editing stages in the OpAI-Bench test set.

Note: Each version contains 4,754 documents and represents a successive stage in a cumulative human–AI editing trajectory. The actual AI character fraction is provided by the character-level ground-truth annotations in OpAI-Bench. For **Pangram 4**, the predicted AI+Assisted fraction is calculated as $\text{fraction}_{\text{AI}} + \text{fraction}_{\text{AI-assisted}}$. The mean and median are calculated independently across the 4,754 documents at each stage.