DETECTION OF MACHINE-GENERATED TEXT USING FINE-TUNED TRANSFORMERS AND A COMPACT CLASSIFIER BASED ON TOKEN-LEVEL PROBABILISTIC FEATURES

Anonymous authors
Paper under double-blind review

ABSTRACT

The widespread adoption of large language models (LLMs) has intensified the need for reliable detection of machine-generated content from human-written text. This work proposes and evaluates two complementary detection strategies: (1) fine-tuned pretrained transformers (RoBERTa, Llama 3.2 3B, and Qwen2.5-1.5B) for text classification, and (2) a compact sequence classifier based on token-level probabilistic features. Both approaches are trained on an extensive corpus combining the M4 and CUDRT datasets, which contain human-written and machine-generated texts. We evaluate the models on a diverse set of challenging benchmarks, including outputs from multiple LLMs and texts edited by humans or LLMs. Across five held-out benchmarks, detector performance was strongly dataset-dependent, and no single approach dominated across all evaluation settings. The Feature Model achieved the strongest performance among our individual models on selected benchmarks, while fine-tuned transformers performed better on others. A hard-voting ensemble combining the Feature Model with two fine-tuned transformers achieved the strongest result on BUST among the evaluated detectors. Ablation experiments show how feature types and feature-generating models affect detection performance. Using the same probabilistic inputs, the Feature Model achieves higher Macro-F1 than Logistic Regression and XGBoost baselines on all five held-out benchmarks. The Feature Model itself contains approximately 100k trainable parameters, although the computational cost of end-to-end inference is dominated by the four LLM feature generators. Overall, the results demonstrate the complementary strengths of fine-tuned and probabilistic-feature detectors rather than the universal superiority of either approach.

1 INTRODUCTION

In recent years, the artificial intelligence (AI) field has witnessed remarkable progress, particularly in the area of natural language processing (NLP). One of the major advances has been the development of large language models (LLMs), such as GPT-4o (OpenAI, 2024a). These models are specifically designed for analysing, processing, and generating human-like text. Leveraging vast datasets and billions of parameters, LLMs demonstrate an impressive ability to generate coherent responses.

Although LLM capabilities offer tremendous potential across various domains, they also introduce new and significant challenges. The text generated by these models is often indistinguishable from that written by humans, raising concerns about potential misuse. Threats arising from improper use include the spread of disinformation (Vykopal et al., 2024), plagiarism (Wahle et al., 2022), and cheating on academic exams or assignments (Cotton et al., 2023).

As these risks grow, the ability to distinguish between human-written and machine-generated content becomes critically important. Furthermore, as the accessibility and quality of machine-generated texts continue to improve, developing reliable detection methods becomes increasingly essential.

The goal of this work is to investigate and develop approaches for identifying texts generated by large language models and distinguishing them from human-written texts. Rather than seeking a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

single universally dominant detector, we propose two complementary detection strategies: (1) fine-tuned pretrained transformers (RoBERTa, Llama 3.2 3B, and Qwen2.5-1.5B) for text classification and (2) a compact sequence classifier based on token-level probabilistic features. While token-level probabilistic features are established in machine-generated text detection, our focus is a systematic study of whether their sequential, multi-model structure provides complementary information under cross-dataset distribution shift. In addition to these two strategies, we have conducted experiments using ensemble methods that combine models from both approaches. Our contributions are threefold. First, we provide a controlled evaluation of the proposed approaches on held-out benchmarks containing multiple generators and editing processes. Second, we analyse the contributions of individual probabilistic feature types and feature-generating models, and compare the proposed sequence classifier with simpler baselines using the same features. Third, we analyse model errors on humanised and edited texts and discuss how benchmark labelling affects their interpretation.

2 BACKGROUND

Text detection aims to distinguish between human-written and machine-generated content produced by large language models (LLMs). Formally, given a candidate text generated by an LLM in response to a prompt, a detector must classify it as either human or machine-written. Detection strategies differ based on access to the source model output logits. In **black-box** detection, the source model is unknown or inaccessible, and the content may have been generated by models such as GPT-4o (OpenAI, 2024a), Llama 3 (Grattafiori et al., 2024), or other undisclosed systems. On the other hand, **white-box** detection assumes that the detector has full access to the source model or, at the very least, partial access to its output logits. Detection methods fall into three main categories mentioned in survey Yang et al. (2024), applicable in both black-box and white-box. Watermarking embeds detectable signals in generated text (Yang et al., 2022; 2023; Kirchenbauer et al., 2023), zero-shot methods identify statistical patterns without labelled training data (Mitchell et al., 2023; Hans et al., 2024; Bao et al., 2024), and training-based methods learn from labelled human-written and machine-generated examples (Li et al., 2023; Guo et al., 2023; Hu et al., 2023; Wu et al., 2023; Verma et al., 2024).

3 METHODOLOGY

3.1 DATASETS

Table 1: Statistics of train, dev, and test datasets.

Dataset	Train	Dev	Test
M4 (Wang et al., 2024b) + CUDRT (Tao et al., 2024)	180 551	16 600	-
GRiD (Qazi et al., 2024)	-	-	6 513
MAGE (Li et al., 2024)	-	-	60 743
Mixset (Zhang et al., 2024)	-	-	600
Beemo (Artemova et al., 2025)	-	-	21 870
BUST (Cornelius et al., 2024)	-	-	30 064

This work uses seven English-language datasets selected for their diversity in domain, structure, and source models. **M4** (Wang et al., 2024b) contains human-written and machine-generated texts from multiple domains and language models and forms part of the training and validation corpus. **CUDRT** (Tao et al., 2024) adds texts covering several generation and editing operations to the same corpus. **GRiD** (Qazi et al., 2024) pairs human Reddit responses with answers generated by GPT-3.5-turbo (Brown et al., 2020). **MAGE** (Li et al., 2024) covers seven writing tasks and outputs from 27 models across several model families (Brown et al., 2020; Touvron et al., 2023a; Chung et al., 2024; Zhang et al., 2022; Workshop, 2024; Black et al., 2022). **Mixset** (Zhang et al., 2024) contains texts revised by humans or LLMs, with humanised model outputs labelled as human-written and the remaining transformed texts as machine-generated (OpenAI, 2024b; Touvron et al., 2023b). **Beemo** (Artemova et al., 2025) contains human and generated texts revised by professional editors or LLMs and retains the machine-generated label for edited model outputs (Grattafiori et al., 2024; OpenAI, 2024a). **BUST** (Cornelius et al., 2024) contains human and machine-generated responses

to instructions grouped into ten categories, with generated responses produced by several model families (Almazrouei et al., 2023; Touvron et al., 2023b; Dettmers et al., 2023; Brown et al., 2020; OpenAI, 2024b). Table 1 summarises their use in our pipeline. Although the table uses conventional *Train*, *Dev*, and *Test* labels, it is important to clarify that only the merged M4 + CUDRT dataset was used for training and validation. All other datasets were exclusively reserved for out-of-distribution testing. This evaluation setup allows us to test the generalisation capabilities of our models on unseen datasets and distributions, although some generator families may occur across multiple benchmarks.

3.2 Approaches

We evaluate two primary detection approaches. The first fine-tunes pre-trained transformers for binary text classification. The second trains a compact classifier on token-level probabilistic features extracted from pre-trained language models. The Hard-voting ensembles then combine the selected classifiers from these two groups.

3.2.1 Fine-tuned Transformer Detectors

All of the transformer-based detectors were fine-tuned using a parameter-efficient fine-tuning (**PEFT**) technique called LoRA (Low-Rank Adaptation) (Hu et al., 2022) and its quantised variant QLoRA (Dettmers et al., 2023). This method injects small, trainable, low-rank matrices into each Transformer layer while keeping the original pre-trained weights frozen. It dramatically reduces the number of trainable parameters, lowers the GPU memory requirements, and shortens the training time. Upon this groundwork, we fine-tuned three architectures:

- RoBERTa$_{\text{BASE}}$ (Liu et al., 2019) with LoRA,
- Llama 3.2 3B (Grattafiori et al., 2024) with QLoRA,
- Qwen2.5-1.5B (Qwen, 2025) with QLoRA.

For both QLoRA models, the pre-trained weights were loaded in 4-bit precision using the NormalFloat4 (NF4) data type and double quantisation. Computations used half precision (float16), while the 4-bit NF4 weights were packed into uint8 storage. LoRA adapters were applied to the query, key, value, and output projection modules (`q_proj`, `k_proj`, `v_proj`, and `o_proj`).

These models were selected to represent a range of different model sizes. RoBERTa was selected as the baseline approach for this problem, as it is commonly used in machine-generated text detection tasks, including the SemEval 2024 competition (Wang et al., 2024a). Beyond RoBERTa, we aimed to fine-tune more recent and larger transformers. Consequently, Llama 3.2 3B and Qwen2.5-1.5B were chosen. Both possess a greater number of parameters than RoBERTa and represent contemporary LLM architectures.

Each model was trained for 5 epochs with a batch size of 32 on sequences of up to 512 tokens. Training was performed using AdamW optimiser and cross-entropy loss. A linearly decaying learning rate schedule, including an initial warm-up phase, was used. To avoid overfitting, dropout was applied to the LoRA adapters. The exact hyperparameter settings are detailed in Table 2.

Table 2: Hyperparameters used to fine-tune models.

Model	Learning rate	Warmup ratio	Rank	Alpha	Dropout rate
RoBERTa$_{\text{BASE}}$	2×10^{-4}	0.05	32	64	0.1
Qwen2.5-1.5B	2×10^{-5}	0.03	64	48	0.05
Llama 3.2 3B	2×10^{-5}	0.03	32	32	0.1

3.2.2 Probabilistic-Feature Classifier

The second approach is based on the assumption that token-level probabilistic features obtained from a specific group of LLMs can effectively distinguish between human-written texts and machine-generated texts (Przybyla et al., 2023; Sarvazyan et al., 2024). Based on this assumption, the following predictability features (Przybyla et al., 2023) were created.

- The logarithm of the probability of the observed token indicates how likely a token is, given the preceding tokens. It is expressed by Equation (1):

$$\alpha_i = \ln P_{\mathrm{M}}(t_i \mid t_{0:i-1}) \tag{1}$$

- The logarithm of the probability of the most probable token indicates the confidence of the model in its best prediction by selecting the token with the highest probability. It is expressed by Equation (2):

$$\beta_i = \max_{w_j \in D} \ln P_{\mathrm{M}}(w_j \mid t_{0:i-1}) \tag{2}$$

- The entropy of the token probability distribution reflects the uncertainty of the model when selecting the next token. It is expressed by Equation (3):

$$\gamma_i = -\sum_{w_j \in D} P_{\mathrm{M}}(w_j \mid t_{0:i-1}) \log_2 P_{\mathrm{M}}(w_j \mid t_{0:i-1}) \tag{3}$$

The above equations use the following notation and parameters:

- t_i is the observed token at position i, and $t_{0:i-1}$ is its preceding token sequence,
- $P_{\mathrm{M}}(w_j \mid t_{0:i-1})$ is the conditional probability that language model M assigns to vocabulary token w_j at position i,
- D is the vocabulary of language model M, and $w_j \in D$ denotes a candidate token.

To extract these features, four models based on the same tokenization algorithm were used: Llama 3.2 1B, Llama 3.2 1B Instruct, Llama 3.2 3B, and Llama 3.2 3B Instruct (Grattafiori et al., 2024). The features are generated for sequences of up to 128 tokens. Because the first token has no preceding context, its feature values are excluded. To account for sequences of varying lengths, a binary mask is also stored to indicate which elements of the sequence contain data and which are padding. In the complete configuration, each token is represented by 12 values, because each language model provides three features. Before training, each of the 12 feature dimensions is standardised independently using the corresponding mean and standard deviation computed from the training features.

The prepared data was used to train a model based on the transformer architecture. The model begins with a linear projection layer that maps the 12-dimensional input features into a 64-dimensional embedding space. This is followed by a transformer encoder consisting of two stacked encoder layers. Each encoder layer uses 4 attention heads, a feedforward dimension of 256 and GELU activation. The transformer encoder models long-range temporal dependencies and contextual interactions across token-level probabilistic feature representations.

After processing the input through the transformer layers, mean pooling is applied to aggregate information across the sequence. This step ensures that the aggregation excludes padded time steps, as indicated by a binary mask. Finally, the pooled representation is passed through a fully connected classification head that outputs logits for the two target classes.

For training, the AdamW optimiser is used with a learning rate of 1×10^{-4}. The model was trained for 10 epochs using the cross-entropy loss function, with a batch size of 128. This classifier contains approximately 100k trainable parameters, excluding the pretrained LLMs used to extract its input features.

3.2.3 Hard-Voting Ensembles

To examine whether the developed models provide complementary predictions, we evaluate an ensemble approach based on hard voting. The feature-inclusive ensemble combines QLoRA Llama 3.2 3B, QLoRA Qwen2.5-1.5B, and the Feature Model. The transformer-only ensemble combines QLoRA Llama 3.2 3B, QLoRA Qwen2.5-1.5B, and LoRA RoBERTa.

4 Results & Analysis

4.1 Experimental Protocol

The developed classifiers were trained and validated only on the merged M4 and CUDRT corpus. GRiD, MAGE, Mixset, Beemo, and BUST were reserved for evaluation. The main comparison

reports Macro-F1 scores for the developed classifiers, the two hard-voting ensembles, and selected existing detectors. Macro-F1 is used as the primary cross-benchmark metric to account for class imbalance and the practical importance of errors on human-written text, with additional metrics reported in Appendix A. The IMGTB framework (Spiegel & Macko, 2023) was used to evaluate the comparison methods consistently across the test datasets, while BUST results were obtained by submitting predictions to its official evaluation website[1], as the ground-truth labels are not publicly available. All experiments were conducted using one NVIDIA Tesla V100-SXM2 GPU with 32 GB of memory.

We also evaluated computational efficiency using parameter count, MACs, FLOPs, and mean latency. Latency was measured after three warm-up passes using ten samples, each averaged over three synchronised forward passes. We measured the Feature Model both with stored features and as a complete pipeline with four parallel feature generators. This experiment does not include ensembles because their computational requirements arise from executing their constituent detectors.

Finally, we conducted a Feature Model ablation on the same evaluation datasets. This analysis reports Macro-F1 while varying the probabilistic feature types and subsets of the four Llama 3.2 feature generators. In addition to that, we analysed Feature Model classification errors.

4.2 Performance across Benchmarks

Table 3: Macro-F1 scores (%) of various models on each test dataset, with implemented models denoted by an asterisk (*). The highest score for each dataset is shown in bold.

Model	GriD	MAGE	Mixset	Beemo	BUST
LoRA RoBERTa*	68.062	49.755	61.465	57.566	75.020
QLoRA Llama 3.2 3B*	90.240	61.596	46.956	63.495	80.765
QLoRA Qwen2.5-1.5B*	**94.874**	63.949	50.000	64.005	81.354
Feature Model*	79.349	**64.968**	51.571	**73.024**	79.376
QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & Feature Model*	94.121	62.627	47.659	66.912	**82.942**
QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & LoRA RoBERTa*	90.389	60.199	51.746	63.417	81.338
Binoculars Hans et al. (2024)	89.424	63.852	40.451	66.628	80.289
ChatGPT-QA Guo et al. (2023)	11.148	42.675	45.246	42.725	31.184
LLMDet Wu et al. (2023)	38.711	43.290	45.545	56.249	64.240
LoRA E5-small Dugan et al. (2024)	41.702	29.914	15.047	16.811	11.249
OpenAI-detector Solaiman et al. (2019)	72.354	63.184	34.145	42.517	56.201
RADAR Hu et al. (2023)	51.502	60.750	**67.501**	60.734	56.569

As shown in Table 3, no detector performed best across all benchmarks. Among the fine-tuned transformers, QLoRA Qwen2.5-1.5B was strongest on GRiD, MAGE, Beemo, and BUST, while LoRA RoBERTa performed best on Mixset, but achieved poor results on GRiD. These differences show that model size and architecture alone do not determine performance across data distributions.

The Feature Model achieved the strongest result on MAGE and Beemo and remained competitive on BUST. Its lower performance on GRiD and Mixset indicates that probabilistic features extracted from the selected Llama models do not transfer uniformly to every generator family and domain. Figure 1 further shows generator-dependent behaviour on MAGE. The Feature Model performed better on outputs from GPT-4, GPT-3.5-Turbo, and GLM130B than on outputs from FLAN-T5, T0, GPT-NeoX, and BLOOM.

From the perspective of the existing detectors, Binoculars was the strongest comparison model on GRiD, MAGE, Beemo, and BUST, while RADAR led this group on Mixset. Binoculars showed a different, yet similarly uneven pattern compared to the Feature Model, performing strongly on a limited group of generators and poorly on several others. The remaining comparison models showed larger performance changes between datasets, which indicates limited generalisation across the evaluated domains and generation processes.

The feature-inclusive ensemble (QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & Feature Model) was strongest on BUST and exceeded Binoculars on four of the five benchmarks. Individual fine-

[1] https://bust.nlp.idsia.ch/

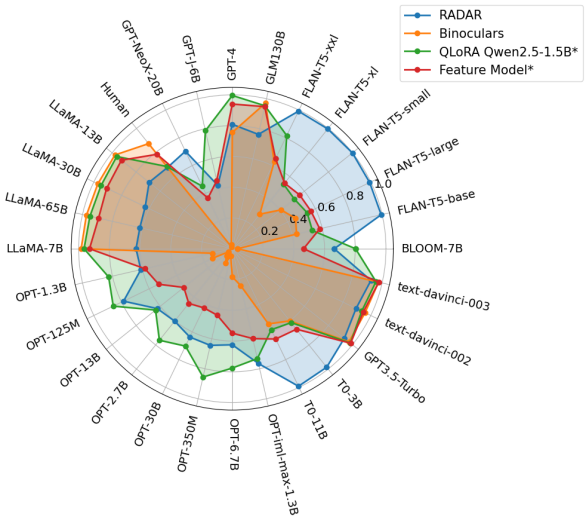

Figure 1: F1 score depending on the text generation models on the MAGE dataset.

tuned transformers remained stronger on GRiD, MAGE, and Mixset. Replacing the Feature Model with LoRA RoBERTa improved slightly the ensemble only on Mixset and reduced its performance on the other benchmarks. The effect of the third member therefore varies across datasets, because, when scores are aggregated, these differences may arise from complementary predictions or from changes in overlapping errors.

4.3 Computational Efficiency of the Detectors

Computational requirements affect the practicality of deploying a detector, particularly when processing time or available resources are constrained. Therefore, we examine parameter count, MACs, FLOPs, and latency across the available sequence lengths, as reported in Table 4.

Table 4: Computational efficiency of the developed detectors at the available sequence lengths. The full pipeline includes parallel feature generation using four Llama models followed by Feature Model inference.

Detector or configuration	Sequence	Total parameters	MACs	FLOPs	Latency (ms)
LoRA RoBERTa	128	127.009M	11.111G	22.222G	16.145
LoRA RoBERTa	256	127.009M	22.221G	44.442G	18.047
LoRA RoBERTa	512	127.009M	44.442G	88.884G	17.400
QLoRA Llama 3.2 3B	128	3.231B	363.127G	726.254G	94.672
QLoRA Llama 3.2 3B	256	3.231B	726.254G	1452.508G	151.687
QLoRA Llama 3.2 3B	512	3.231B	1452.507G	2905.014G	298.893
QLoRA Qwen2.5-1.5B	128	1.561B	169.937G	339.874G	56.715
QLoRA Qwen2.5-1.5B	256	1.561B	339.874G	679.748G	83.100
QLoRA Qwen2.5-1.5B	512	1.561B	679.747G	1359.494G	147.222
Feature Model classifier only	128	0.101M	0.017G	0.034G	1.027
Feature Model full pipeline	128	8.898B	1146.572G	2293.144G	281.395

For the LoRA and QLoRA detectors, arithmetic requirements increase with sequence length. The QLoRA detectors also show increasing latency, while RoBERTa maintains comparatively low latency with minor variation across the evaluated sequence lengths. LoRA RoBERTa is the fastest and uses the fewest parameters among the fine-tuned detectors, but this efficiency is accompanied by more moderate and less consistent detection performance across benchmarks. Qwen remains less computationally demanding and faster than Llama at each shared sequence setting, consistent with its smaller parameter count. The Feature Model classifier itself is substantially more compact than the fine-tuned detectors, while feature extraction accounts for most of the cost of the complete pipeline. At the shared sequence length, the full pipeline requires more MACs, FLOPs, and latency

than each fine-tuned detector. For longer texts, Feature Model keeps its cost unchanged, because it is adapted to sequences of at most 128 tokens, which means that longer inputs are truncated to this limit, so its computational requirements do not increase with text length beyond 128 tokens. At a sequence length of 512 tokens, the full pipeline requires fewer arithmetic operations and lower latency than Llama, while remaining more demanding than Qwen and RoBERTa. The feature-based approach therefore combines a compact classifier with competitive detection performance, although feature generation determines most of the complete pipeline computational requirements.

The feature-based approach is also modular rather than a single end-to-end detector. The four feature generators are independent components and can, therefore, be run separately or in parallel when sufficient hardware resources are available. In contrast, each fine-tuned detector performs prediction as a single model and must be scaled as a complete unit. We emphasise that the compactness of the approximately 100k-parameter Feature Model classifier does not imply that the complete four-generator pipeline is computationally lighter than all fine-tuned baselines.

4.4 Feature-Model Ablation

The ablation analysis assesses the impact of the number and type of language models used for predictability-feature generation, as well as the contribution of individual feature types and their combinations.

Table 5: Macro-F1 scores (%) of the Feature Model under different feature configurations. The full feature set uses α, β, and γ from all four models. Rows labelled with one or two symbols use only the corresponding feature types from all four models. All model labels refer to Llama 3.2 variants, and rows listing multiple variants combine all three feature types generated by each named model.

Configuration	GRiD	MAGE	Mixset	Beemo	BUST
Full feature set	79.349	**64.968**	51.571	**73.024**	79.376
α	70.069	61.269	44.697	67.349	**89.509**
β	68.947	57.326	42.169	64.105	66.050
γ	62.689	57.337	51.850	58.861	68.571
$\alpha + \beta$	70.694	63.517	47.019	70.882	73.964
$\alpha + \gamma$	75.082	64.136	53.157	71.361	73.945
$\beta + \gamma$	67.446	59.745	52.232	66.306	68.136
1B	74.441	59.138	54.233	57.821	63.871
1B Instruct	59.860	55.028	57.684	63.829	65.818
3B	77.444	61.645	48.004	58.797	63.178
3B Instruct	67.675	58.070	53.371	61.831	62.587
1B + 1B Instruct	65.469	59.466	54.588	70.481	73.220
1B + 3B	78.111	59.591	**60.469**	63.285	68.019
1B + 3B Instruct	66.619	57.711	52.909	67.355	87.157
1B Instruct + 3B	61.228	58.944	56.012	67.245	71.435
1B Instruct + 3B Instruct	78.469	60.186	50.012	63.750	67.231
3B + 3B Instruct	63.377	59.282	45.346	65.665	72.735
1B + 1B Instruct + 3B	78.967	63.644	53.134	72.198	76.185
1B + 1B Instruct + 3B Instruct	79.113	59.237	53.029	71.310	77.911
1B + 3B + 3B Instruct	**85.132**	63.206	48.533	72.679	77.236
1B Instruct + 3B + 3B Instruct	83.389	62.868	49.251	70.532	74.913

The ablation results in Table 5 show that α is the most informative individual feature type on most benchmarks and the strongest overall result on BUST. The γ feature is more effective on Mixset, whereas β generally provides the weakest individual representation. Among the reduced configurations, $\alpha + \gamma$ consistently performs better than $\beta + \gamma$. It achieves the strongest results on GRiD, MAGE, Mixset, and Beemo, whereas $\alpha + \beta$ performs comparably on BUST. These results indicate that the observed-token probability (α) provides the most useful individual signal and is complemented by entropy (γ) in most datasets.

The model-specific configurations further show that neither model size nor instruction tuning provides a consistent advantage. Llama 3.2 3B produces the strongest single-model representation on GRiD and MAGE, while Llama 3.2 1B Instruct performs best among the single-model variants on

Mixset, Beemo, and BUST. Combining models improves some results, but the gains are not monotonic with the number of feature generators. The combination of Llama 3.2 1B, Llama 3.2 3B, and Llama 3.2 3B Instruct performs best on GRiD, while the combination of Llama 3.2 1B and Llama 3.2 3B performs best on Mixset. The full feature set is strongest on MAGE and Beemo. Overall, selected subsets of feature-generating models can achieve stronger results on individual datasets, while the full configuration provides strong overall performance across diverse data distributions.

To assess the respective contributions of the probabilistic representation and the transformer architecture, we additionally trained Logistic Regression and XGBoost classifiers on flattened versions of the same features (Appendix C). Both baselines achieve substantial F1 score on several datasets, confirming that the probabilistic features themselves contain a strong detection signal. Nevertheless, the Feature Model achieves higher Macro-F1 than both baselines on all five evaluation datasets, indicating a more balanced recognition of human-written and machine-generated text. These results suggest that the proposed sequence model contributes primarily to class-balanced and cross-dataset generalisation.

4.5 Error Analysis

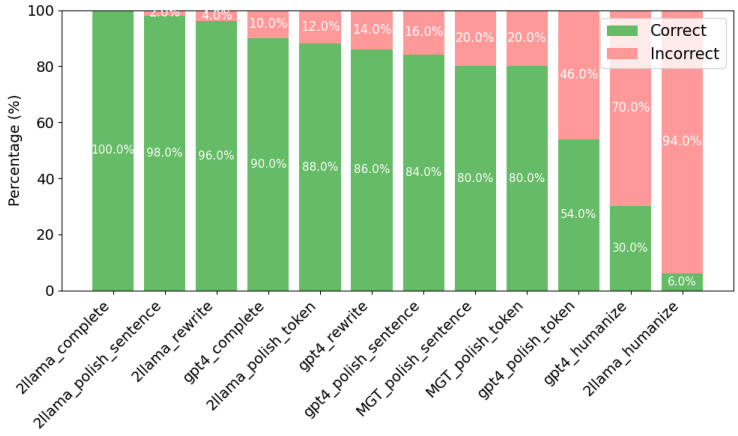

Figure 2: Category-level accuracy of the Feature Model on Mixset for different transformation models and operations. Humanised outputs are evaluated as human-written, while completed, rewritten, and polished outputs retain the machine-generated label, following the Mixset labelling policy.

This section focuses on the error profile of the Feature Model, particularly its behaviour across the text transformations represented in Mixset. The detailed evaluation results show that the Feature Model tends to prioritise the detection of machine-generated text. The same error direction is observed for several fine-tuned detectors and is therefore not unique to the feature-based approach.

As shown in Figure 2, the performance of the Feature Model varies with both the transformation operation and the model used to perform it. Accuracy reached 100% and 90% for completion using Llama and GPT-4, respectively, and 96% and 86% for rewriting. Sentence-level polishing produced accuracies between 80% and 98%, while token-level polishing showed greater variation, including 54% for GPT-4 token polishing. Humanisation was the most difficult operation, with accuracies of 30% for GPT-4 humanisation and 6% for Llama humanisation. In Mixset, humanisation uses an LLM to introduce human-like noise, including typographical and grammatical errors, into machine-generated text. The resulting text of humanisation process is labelled as human-written. Consequently, classifying these humanised texts as machine-generated is counted as a classification error under the benchmark labelling policy. However, this highlights a distinction between final-text authorship and provenance: humanised samples may be labelled as human-written despite originating from an LLM. Thus, the low humanisation accuracy reflects sensitivity to the definition of machine-generated text rather than incorrect benchmark labels.

The stronger recognition of genuinely human-written samples on MAGE (Figure 1) suggests that the humanisation failures should not be interpreted solely as a general inability to recognise human

text. Instead, the contrast between MAGE and Mixset highlights a limitation of binary evaluation when texts undergo mixed transformations. This distinction is particularly relevant when detector outputs are used to assess authorship or provenance.

4.6 Discussion and Limitations

The interpretation of edited texts depends on each benchmark labelling policy. Mixset treats humanised model outputs as human-written, whereas Beemo retains the machine-generated label when a human edits model output. A prediction that counts as an error under one policy could therefore be consistent with a different definition of text origin. Direct language model output is clearly machine-generated, but this becomes less clear when the text is revised by humans or language models. Human editing may substantially change machine-generated content without removing its machine provenance, while LLM editing introduces machine influence into an originally human-written text. Consequently, the binary labels used by a benchmark represent a decision about whether authorship should be determined by the original source, the final editor, or machine involvement. Evaluation results must therefore be interpreted in relation to the adopted labelling policy, particularly for datasets containing edited or humanised texts.

The probabilistic features were extracted exclusively with Llama 3.2 variants. We intentionally used a single model family to maintain a controlled experimental setup and ensure a shared vocabulary and tokenizer across all feature generators. However, this setup leaves open the question of how well the approach transfers when substantially different models are used for feature generation. Future work should therefore examine models from diverse LLM families and determine whether heterogeneous feature generators provide complementary information and improve generalisation.

Although the Feature Model classifier is compact, deploying the complete feature pipeline is more involved than deploying a single fine-tuned detector. Inference requires four Llama feature generators followed by the Feature Model classifier, which means that several components must be configured, coordinated, and maintained.

5 Conclusion

In this work, we addressed the problem of identifying texts generated by large language models (LLMs) and distinguishing them from human-written content. We proposed two complementary detection strategies based on parameter-efficient fine-tuning and probabilistic features.

The first strategy applied LoRA and QLoRA to fine-tune RoBERTa, Llama 3.2 3B, and Qwen2.5-1.5B, achieving strong detection performance with parameter-efficient adaptation. The second strategy used a compact classifier with approximately $\sim$ 100k trainable parameters and token-level probabilistic features generated by multiple LLMs. It performed best among the individual detectors on MAGE and Beemo and remained competitive on BUST. Despite the small classifier size, end-to-end inference still requires the LLM-based feature generators.

To boost robustness, we evaluated hard-voting ensembles combining QLoRA transformers with either the Feature Model or RoBERTa. The feature-inclusive ensemble achieved the top result on BUST and outperformed Binoculars across four of five benchmarks. However, it did not consistently outperform its strongest individual constituent, indicating that the benefits of hard voting are dataset-dependent.

Overall, the results show that fine-tuned transformer detectors and token-level probabilistic representations provide complementary signals whose effectiveness varies across datasets, generator families, and editing processes. This benchmark dependence, together with the observed sensitivity to labelling policy, highlights both the potential and the limitations of current machine-generated text detection systems.

AI use statement

In this work, we used generative AI tools to assist with software code creation and to draft parts of a research paper. All AI-assisted code was manually reviewed and verified before being used to produce the reported results. We have also verified all AI-assisted text and revised it where

necessary. We take responsibility for the final content of this work, including text and artifacts produced with the aid of generative AI.

Reproducibility Statement

The information required to reproduce the main methodology and experimental results is provided throughout the paper. The Datasets (Section 3.1) and Approaches (Section 3.2) subsections describe the data splits, sequence lengths, model architectures, probabilistic features, optimisers, loss functions, training epochs, batch sizes, hyperparameters, and ensemble construction. The Experimental Protocol subsection (Section 4.1) specifies the evaluation procedure, computational-efficiency measurements, and computing resources. The creators of all datasets, pretrained models, comparison methods, and software components are credited in Appendix D. Anonymous source code implementing all proposed detection strategies is provided in the supplementary materials.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models, 2023. URL https://arxiv.org/abs/2311.16867.

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 (eds.), *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)*, pp. 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/.

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

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. GPT-NeoX-20B: An open-source autoregressive language model. In Angela Fan, Suzana Ilic, Thomas Wolf, and Matthias Gallé (eds.), *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pp. 95–136, virtual+Dublin, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.bigscience-1.9. URL https://aclanthology.org/2022.bigscience-1.9/.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. URL http://jmlr.org/papers/v25/23-0870.html.

Joseph Cornelius, Oscar Lithgow-Serrano, Sandra Mitrovic, Ljiljana Dolamic, and Fabio Rinaldi. BUST: Benchmark for the evaluation of detectors of LLM-generated text. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8029–8057, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.444. URL https://aclanthology.org/2024.naacl-long.444.

Debby Cotton, Peter Cotton, and Reuben Shipway. Chatting and cheating: Ensuring academic integrity in the era of chatgpt. *Innovations in Education and Teaching International*, 61, 03 2023. doi: 10.1080/14703297.2023.2190148.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115. Curran Associates, Inc., 2023. doi: 10.52202/075280-0441. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12463–12492, Bangkok, Thailand, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.acl-long.674.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection, 2023. URL https://arxiv.org/abs/2301.07597.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17519–17537. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hans24a.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Radar: Robust ai-text detection via adversarial learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 15077–15095. Curran Associates, Inc., 2023. doi: 10.52202/075280-0662. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/30e15e5941ae0cdab7ef58cc8d59a4ca-Paper-Conference.pdf.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

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

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. MAGE: Machine-generated text detection in the wild. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36–53, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.3. URL https://aclanthology.org/2024.acl-long.3.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24950–24962. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/mitchell23a.html.

OpenAI. Gpt-4o system card, 2024a. URL https://arxiv.org/abs/2410.21276.

OpenAI. Gpt-4 technical report, 2024b. URL https://arxiv.org/abs/2303.08774.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Piotr Przybyla, Nicolau Duran-Silva, and Santiago Egea Gómez. I've seen things you machines wouldn't believe: Measuring content predictability to identify automatically-generated text. In Manuel Montes-y-Gómez, Francisco Rangel, Salud María Jiménez-Zafra, Marco Casavantes, Begoña Altuna, Miguel Ángel Álvarez-Carmona, Gemma Bel-Enguix, Luis Chiruzzo, Iker de la Iglesia, Hugo Jair Escalante, Miguel Ángel García Cumbreras, José Antonio García-Díaz, José Ángel González Barba, Roberto Labadie Tamayo, Salvador Lima, Pablo Moral, Flor Miriam Plaza del Arco, and Rafael Valencia-García (eds.), *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023) co-located with the Conference of the Spanish Society for Natural Language Processing (SEPLN 2023), Jaén, Spain, September 26, 2023*, volume 3496 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2023. URL https://ceur-ws.org/Vol-3496/autextification-paper7.pdf.

Zubair Qazi, William Shiao, and Evangelos E. Papalexakis. Gpt-generated text detection: Benchmark dataset and tensor-based detection method. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 842–846, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701726. doi: 10.1145/3589335.3651513. URL https://doi.org/10.1145/3589335.3651513.

Qwen. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Areg Mikael Sarvazyan, José Ángel González, and Marc Franco-salvador. Genaios at SemEval-2024 task 8: Detecting machine-generated text by mixing language model probabilistic features. In Atul Kr. Ojha, A. Seza Doğruöz, Harish Tayyar Madabushi, Giovanni Da San Martino, Sara Rosenthal, and Aiala Rosá (eds.), *Proceedings of the 18th International Workshop*

on Semantic Evaluation (SemEval-2024), pp. 101–107, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.semeval-1.17. URL https://aclanthology.org/2024.semeval-1.17/.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

Michal Spiegel and Dominik Macko. IMGTB: A framework for machine-generated text detection benchmarking, 2023.

Zhen Tao, Yanfang Chen, Dinghao Xi, Zhiyu Li, and Wei Xu. Cudrt: Benchmarking the detection models of human vs. large language models generated texts, 2024. URL https://arxiv.org/abs/2406.09056.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models, 2023a. URL https://arxiv.org/abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models, 2023b. URL https://arxiv.org/abs/2307.09288.

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

Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Maria Bielikova. Disinformation capabilities of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14830–14847. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.793. URL http://dx.doi.org/10.18653/v1/2024.acl-long.793.

Jan Philip Wahle, Terry Ruas, Frederic Kirstein, and Bela Gipp. How large language models are transforming machine-paraphrase plagiarism. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 952–963, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.62. URL https://aclanthology.org/2022.emnlp-main.62/.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, and Thomas Arnold. SemEval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In Atul Kr. Ojha, A. Seza Doğruöz, Harish Tayyar Madabushi, Giovanni Da San Martino, Sara Rosenthal, and Aiala Rosá (eds.), *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pp. 2057–2079, Mexico City, Mexico, June 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.semeval-1.279. URL https://aclanthology.org/2024.semeval-1.279/.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, et al. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1369–1407, St. Julian's, Malta, March 2024b. Association for Computational Linguistics. URL https://aclanthology.org/2024.eacl-long.83.

BigScience Workshop. Bloom: A 176b-parameter open-access multilingual language model. *Journal of Machine Learning Research*, 25(422):1–74, 2024. URL http://jmlr.org/papers/v25/23-0581.html.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. LLMDet: A third party large language models generated text detection tool. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2113–2133, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.139. URL https://aclanthology.org/2023.findings-emnlp.139/.

Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. Tracing text provenance via context-aware lexical substitution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11613–11621, Jun. 2022. doi: 10.1609/aaai.v36i10.21415. URL https://ojs.aaai.org/index.php/AAAI/article/view/21415.

Xi Yang, Kejiang Chen, Weiming Zhang, Chang Liu, Yuang Qi, Jie Zhang, Han Fang, and Nenghai Yu. Watermarking text generated by black-box language models, 2023. URL https://arxiv.org/abs/2305.08883.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Ruth Petzold, William Yang Wang, and Wei Cheng. A survey on detection of LLMs-generated content. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9786–9805, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.572. URL https://aclanthology.org/2024.findings-emnlp.572/.

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 (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 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/.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

A Detailed Evaluation Results

Table 6 presents accuracy, F1, macro-F1, precision, and recall for all developed detectors, hard-voting ensembles, and comparison models on the five evaluation datasets.

Table 6: Detailed results for the developed classifiers, hard-voting ensembles, and comparison models. Dashes indicate unavailable metrics. Implemented models are denoted by an asterisk (*). The highest score for each dataset is shown in bold.

Dataset	Model	Accuracy	F1	Macro-F1	Precision	Recall
GRiD	LoRA RoBERTa*	0.70689	0.58902	0.68062	0.41745	**1.00000**
	QLoRA Llama 3.2 3B*	0.92753	0.85287	0.90240	0.74348	**1.00000**
	QLoRA Qwen2.5-1.5B*	**0.96392**	**0.92085**	**0.94874**	**0.85384**	0.99927
	Feature Model*	0.82788	0.70921	0.79349	0.54966	0.99927
	QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & Feature Model*	0.95824	0.90957	0.94121	0.83415	**1.00000**
	QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & LoRA RoBERTa*	0.92876	0.85500	0.90389	0.74672	**1.00000**
	Binoculars	0.92108	0.84097	0.89424	0.72908	0.99342
	ChatGPT-QA	0.12452	0.00384	0.11148	0.00253	0.00804
	LLMDet	0.38799	0.36387	0.38711	0.23275	0.83333
	LoRA E5-small	0.71534	0.00000	0.41702	0.00000	0.00000
	OpenAI-detector	0.81514	0.56440	0.72354	0.55874	0.57018
	RADAR	0.51988	0.46647	0.51502	0.30425	0.99927
MAGE	LoRA RoBERTa*	0.54240	0.64767	0.49755	0.52770	0.83824
	QLoRA Llama 3.2 3B*	0.63140	**0.69296**	0.61596	0.59528	0.82899
	QLoRA Qwen2.5-1.5B*	0.64057	0.65919	0.63949	0.62870	0.69279
	Feature Model*	0.64969	0.64815	**0.64968**	0.65332	0.64305
	QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & Feature Model*	0.63028	0.66496	0.62627	0.60971	0.73122
	QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & LoRA RoBERTa*	0.61523	0.67458	0.60199	0.58594	0.79484
	Binoculars	**0.65005**	0.57395	0.63852	0.73747	0.46978
	ChatGPT-QA	0.43612	0.50001	0.42675	0.45038	0.56195
	LLMDet	0.50322	0.63259	0.43290	0.50292	**0.85235**
	LoRA E5-small	0.38711	0.05084	0.29914	0.11401	0.03271
	OpenAI-detector	0.64518	0.56176	0.63184	**0.73860**	0.45324
	RADAR	0.60967	0.63665	0.60750	0.59731	0.68154
Mixset	LoRA RoBERTa*	**0.86333**	**0.92421**	0.61465	0.85911	**1.00000**
	QLoRA Llama 3.2 3B*	0.82333	0.90275	0.46956	0.83390	0.98400
	QLoRA Qwen2.5-1.5B*	0.82000	0.90000	0.50000	0.83793	0.97200
	Feature Model*	0.72667	0.83534	0.51571	0.83871	0.83200
	QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & Feature Model*	0.82000	0.90055	0.47659	0.83447	0.97800
	QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & LoRA RoBERTa*	0.83667	0.90993	0.51746	0.84184	0.99000
	Binoculars	0.43833	0.54643	0.40451	0.83539	0.40600
	ChatGPT-QA	0.56333	0.69885	0.45246	0.82162	0.60800
	LLMDet	0.80833	0.89381	0.45545	0.83019	0.96800
	LoRA E5-small	0.16833	0.02729	0.15047	0.53846	0.01400
	OpenAI-detector	0.34167	0.35352	0.34145	**0.97297**	0.21600
	RADAR	0.74833	0.82938	**0.67501**	0.95325	0.73400
Beemo	LoRA RoBERTa*	0.74947	0.84724	0.57566	0.82706	0.86843
	QLoRA Llama 3.2 3B*	0.80567	0.88459	0.63495	0.84263	0.93096
	QLoRA Qwen2.5-1.5B*	0.75679	0.84504	0.64005	0.86179	0.82893
	Feature Model*	**0.84664**	**0.90744**	**0.73024**	0.87732	**0.93970**
	QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & Feature Model*	0.81317	0.88744	0.66912	0.85657	0.92061
	QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & LoRA RoBERTa*	0.79465	0.87647	0.63417	0.84480	0.91061
	Binoculars	0.75528	0.83862	0.66628	**0.88754**	0.79481
	ChatGPT-QA	0.48244	0.60505	0.42725	0.77667	0.49554
	LLMDet	0.78075	0.87151	0.56249	0.82039	0.92941
	LoRA E5-small	0.19524	0.01786	0.16811	0.37736	0.00914
	OpenAI-detector	0.43919	0.51493	0.42517	0.83579	0.37209
	RADAR	0.79282	0.87721	0.60734	0.83406	0.92507
BUST	LoRA RoBERTa*	–	0.88903	0.75020	–	–
	QLoRA Llama 3.2 3B*	–	0.91025	0.80765	–	–
	QLoRA Qwen2.5-1.5B*	–	0.91044	0.81354	–	–
	Feature Model*	–	0.90967	0.79376	–	–
	QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & Feature Model*	–	**0.92139**	**0.82942**	–	–
	QLoRA Llama 3.2 3B & QLoRA Qwen2.5-1.5B & LoRA RoBERTa*	–	0.91376	0.81338	–	–
	Binoculars	–	0.90945	0.80289	–	–
	ChatGPT-QA	–	0.48696	0.31184	–	–
	LLMDet	–	0.85308	0.64240	–	–
	LoRA E5-small	–	0.05892	0.11249	–	–
	OpenAI-detector	–	0.72498	0.56201	–	–
	RADAR	–	0.82937	0.56569	–	–

B Detailed Feature-Model Ablation Results

Tables 7–11 report detailed results for the Feature Model configurations evaluated in Table 5. All model labels refer to Llama 3.2 variants.

Table 7: Detailed Feature Model ablation results on GRiD.

Configuration	Accuracy	F1	Macro-F1	Precision	Recall
Full feature set	0.82788	0.70921	0.79349	0.54966	0.99927
α	0.72931	0.60814	0.70069	0.43692	**1.00000**
β	0.72179	0.58930	0.68947	0.42707	0.95029
γ	0.64732	0.53959	0.62689	0.37172	0.98392
$\alpha + \beta$	0.73637	0.61407	0.70694	0.44336	0.99854
$\alpha + \gamma$	0.78412	0.65973	0.75082	0.49313	0.99635
$\beta + \gamma$	0.70213	0.57954	0.67446	0.41189	0.97734
1B	0.78474	0.64289	0.74441	0.49335	0.92251
1B Instruct	0.61462	0.51842	0.59860	0.35146	0.98757
3B	0.81545	0.67827	0.77444	0.53505	0.92617
3B Instruct	0.70444	0.58216	0.67675	0.41402	0.98026
1B + 1B Instruct	0.67772	0.56551	0.65469	0.39446	0.99854
1B + 3B	0.81928	0.68969	0.78111	0.53938	0.95614
1B + 3B Instruct	0.69062	0.57588	0.66619	0.40437	**1.00000**
1B Instruct + 3B	0.62936	0.53090	0.61228	0.36157	0.99854
1B Instruct + 3B Instruct	0.82082	0.69649	0.78469	0.54057	0.97880
3B + 3B Instruct	0.65377	0.54819	0.63377	0.37759	**1.00000**
1B + 1B Instruct + 3B	0.82404	0.70464	0.78967	0.54419	0.99927
1B + 1B Instruct + 3B Instruct	0.82558	0.70631	0.79113	0.54640	0.99854
1B + 3B + 3B Instruct	**0.88331**	**0.78236**	**0.85132**	**0.64313**	0.99854
1B Instruct + 3B + 3B Instruct	0.86719	0.75952	0.83389	0.61283	0.99854

Table 8: Detailed Feature Model ablation results on MAGE.

Configuration	Accuracy	F1	Macro-F1	Precision	Recall
Full feature set	**0.64969**	0.64815	**0.64968**	**0.65332**	0.64305
α	0.61383	0.63375	0.61269	0.60458	0.66589
β	0.57676	0.53462	0.57326	0.59629	0.48451
γ	0.57801	0.61785	0.57337	0.56619	0.67987
$\alpha + \beta$	0.63578	0.65005	0.63517	0.62757	0.67419
$\alpha + \gamma$	0.64167	0.65190	0.64136	0.63591	0.66871
$\beta + \gamma$	0.59747	0.59992	0.59745	0.59836	0.60148
1B	0.59139	0.58865	0.59138	0.59474	0.58268
1B Instruct	0.55791	0.60885	0.55028	0.54746	0.68574
3B	0.61823	0.59037	0.61645	0.63944	0.54830
3B Instruct	0.58071	0.58239	0.58070	0.58209	0.58268
1B + 1B Instruct	0.60223	0.65003	0.59466	0.58189	0.73624
1B + 3B	0.59648	0.61113	0.59591	0.59165	0.63193
1B + 3B Instruct	0.58168	0.62107	0.57711	0.56927	0.68325
1B Instruct + 3B	0.60598	**0.67185**	0.58944	0.57706	**0.80389**
1B Instruct + 3B Instruct	0.60277	0.58288	0.60186	0.61599	0.55315
3B + 3B Instruct	0.60474	0.66251	0.59282	0.57955	0.77318
1B + 1B Instruct + 3B	0.63680	0.64790	0.63644	0.63076	0.66599
1B + 1B Instruct + 3B Instruct	0.59274	0.60467	0.59237	0.58941	0.62074
1B + 3B + 3B Instruct	0.63298	0.61373	0.63206	0.65023	0.58111
1B Instruct + 3B + 3B Instruct	0.62909	0.61626	0.62868	0.64075	0.59358

Table 9: Detailed Feature Model ablation results on Mixset.

Configuration	Accuracy	F1	Macro-F1	Precision	Recall
Full feature set	0.72667	0.83534	0.51571	0.83871	0.83200
α	0.76500	0.86635	0.44697	0.82342	**0.91400**
β	0.50667	0.64337	0.42169	0.80909	0.53400
γ	0.74333	0.84752	0.51850	0.83922	0.85600
$\alpha + \beta$	0.72000	0.83399	0.47019	0.82422	0.84400
$\alpha + \gamma$	0.74500	0.84776	0.53157	0.84356	0.85200
$\beta + \gamma$	0.66333	0.78186	0.52232	0.84977	0.72400
1B	0.62500	0.73684	0.54233	0.88732	0.63000
1B Instruct	0.72833	0.83003	0.57684	0.86710	0.79600
3B	0.53667	0.65163	0.48004	0.87248	0.52000
3B Instruct	0.63167	0.74743	0.53371	0.87200	0.65400
1B + 1B Instruct	0.76000	0.85771	0.54588	0.84766	0.86800
1B + 3B	0.72500	0.82277	**0.60469**	**0.88863**	0.76600
1B + 3B Instruct	0.72500	0.83283	0.52909	0.84394	0.82200
1B Instruct + 3B	**0.78500**	**0.87464**	0.56012	0.85066	0.90000
1B Instruct + 3B Instruct	0.57167	0.68924	0.50012	0.87156	0.57000
3B + 3B Instruct	0.76500	0.86610	0.45346	0.82459	0.91200
1B + 1B Instruct + 3B	0.72833	0.83519	0.53134	0.84458	0.82600
1B + 1B Instruct + 3B Instruct	0.75500	0.85517	0.53029	0.84272	0.86800
1B + 3B + 3B Instruct	0.70167	0.81901	0.48533	0.82822	0.81000
1B Instruct + 3B + 3B Instruct	0.69500	0.81307	0.49251	0.83090	0.79600

Table 10: Detailed Feature Model ablation results on Beemo.

Configuration	Accuracy	F1	Macro-F1	Precision	Recall
Full feature set	**0.84664**	**0.90744**	**0.73024**	0.87732	0.93970
α	0.82533	0.89615	0.67349	0.85452	0.94204
β	0.77947	0.86395	0.64105	0.85291	0.87529
γ	0.79694	0.88136	0.58861	0.82741	**0.94284**
$\alpha + \beta$	0.83434	0.90000	0.70882	0.87028	0.93181
$\alpha + \gamma$	0.81756	0.88615	0.71361	0.88479	0.88752
$\beta + \gamma$	0.80389	0.88089	0.66306	0.85670	0.90649
1B	0.65876	0.76253	0.57821	0.86010	0.68484
1B Instruct	0.78235	0.86656	0.63829	0.85035	0.88340
3B	0.65697	0.75659	0.58797	0.87504	0.66638
3B Instruct	0.73448	0.82888	0.61831	0.85553	0.80384
1B + 1B Instruct	0.82666	0.89446	0.70481	0.87191	0.91821
1B + 3B	0.72789	0.81965	0.63285	0.87240	0.77292
1B + 3B Instruct	0.80434	0.88018	0.67355	0.86277	0.89832
1B Instruct + 3B	0.81733	0.89029	0.67245	0.85681	0.92650
1B Instruct + 3B Instruct	0.75354	0.84260	0.63750	0.86142	0.82459
3B + 3B Instruct	0.81806	0.89207	0.65665	0.84893	0.93981
1B + 1B Instruct + 3B	0.82693	0.89280	0.72198	0.88490	0.90083
1B + 1B Instruct + 3B Instruct	0.82926	0.89566	0.71310	0.87622	0.91598
1B + 3B + 3B Instruct	0.82949	0.89430	0.72679	**0.88708**	0.90163
1B Instruct + 3B + 3B Instruct	0.81235	0.88291	0.70532	0.88145	0.88437

Table 11: Detailed Feature Model ablation results on BUST.

Configuration	F1	Macro-F1	Micro-F1
Full feature set	**0.90967**	0.79376	**0.91036**
α	0.89386	**0.89509**	0.75670
β	0.83836	0.66050	0.82727
γ	0.86600	0.68571	0.87081
$\alpha + \beta$	0.88335	0.73964	0.88162
$\alpha + \gamma$	0.87857	0.73945	0.87241
$\beta + \gamma$	0.85468	0.68136	0.85009
1B	0.81088	0.63871	0.78556
1B Instruct	0.84814	0.65818	0.84713
3B	0.79661	0.63178	0.76384
3B Instruct	0.81146	0.62587	0.79065
1B + 1B Instruct	0.88137	0.73220	0.88092
1B + 3B	0.83228	0.68019	0.80964
1B + 3B Instruct	0.87198	0.87157	0.71083
1B Instruct + 3B	0.87632	0.71435	0.87873
1B Instruct + 3B Instruct	0.83735	0.67231	0.82108
3B + 3B Instruct	0.88195	0.72735	0.88425
1B + 1B Instruct + 3B	0.88867	0.76185	0.88272
1B + 1B Instruct + 3B Instruct	0.90027	0.77911	0.89805
1B + 3B + 3B Instruct	0.89149	0.77236	0.88395
1B Instruct + 3B + 3B Instruct	0.87992	0.74913	0.87117

C Additional Baseline Experiments

These additional experiments examine whether the observed detection performance arises primarily from the probabilistic features or from the sequence-modelling architecture of the Feature Model. We evaluated XGBoost and Logistic Regression classifiers using the same stored token-level probabilistic features as the Feature Model. In both cases, the first 128 positions were retained and flattened into a single input vector for each text. For sequences shorter than 128 tokens, every feature value at each padded position was stored as zero.

The XGBoost (Chen & Guestrin, 2016) classifier used a maximum of 3000 estimators, a learning rate of 0.03, a maximum tree depth of 6, histogram-based tree construction, column subsampling of 0.7, and a binary logistic objective. Validation AUC was used for early stopping with a patience of 100 iterations. The best iteration was 2998, with a validation AUC of 0.9168. A decision threshold of 0.25 was selected on the validation set, where it produced an F1 score of 0.8757, and was then applied unchanged to the held-out datasets.

For Logistic Regression, the flattened features were standardised before classification. The classifier used the `lbfgs` solver with a maximum of 1000 iterations. A decision threshold of 0.21 was selected on the validation set, where it produced an F1 score of 0.8030, and was then applied to the labelled held-out datasets. Table 12 reports the results of both baselines.

Table 12: Results of the additional baseline experiments using token-level probabilistic features. Dashes indicate unavailable metrics.

Dataset	Model	Accuracy	F1	Macro-F1	Precision	Recall
GRiD	XGBoost	**0.530**	**0.472**	**0.524**	**0.309**	**1.000**
	Logistic Regression	0.332	0.386	0.327	0.239	**1.000**
MAGE	XGBoost	0.550	0.650	**0.511**	**0.533**	0.832
	Logistic Regression	**0.551**	**0.670**	0.484	0.531	**0.907**
Mixset	XGBoost	0.798	0.886	**0.501**	**0.837**	0.942
	Logistic Regression	**0.805**	**0.892**	0.454	0.830	**0.964**
Beemo	XGBoost	0.803	0.888	**0.540**	**0.815**	0.974
	Logistic Regression	**0.807**	**0.891**	0.502	0.809	**0.993**
BUST	XGBoost	–	**0.736**	**0.604**	–	–
	Logistic Regression	–	0.570	0.468	–	–

Overall, both baselines show that the probabilistic features contain substantial information for detecting machine-generated text without sequence modelling. Logistic Regression obtains stronger positive-class F1 on MAGE, Mixset, and Beemo, largely through higher recall, while XGBoost performs better on GRiD and BUST. XGBoost also achieves higher Macro-F1 across all five datasets, indicating more balanced performance between the two classes. The gap between positive-class F1 and Macro-F1 is particularly pronounced for Logistic Regression, showing that its strong detection of machine-generated texts is accompanied by weaker recognition of human-written texts.

Compared with these flattened-feature baselines, the Feature Model provides more balanced results on most datasets and achieves higher F1 and Macro-F1 on Beemo and BUST. The baseline results therefore confirm that the probabilistic features account for an important part of the detection performance, while the sequence-modelling architecture remains relevant for learning relationships between token positions and achieving more consistent generalisation across datasets.

D Existing Assets and Licences

Tables 13–15 report the official sources and licence information for the external assets used in this study. The assets are cited and used according to the information provided by their official sources. No third-party dataset, pretrained weights, merged training corpus, or derived checkpoint is redistributed with this paper.

Table 13: Official sources and licence information for datasets. "No specific licence" indicates that the official source does not identify a licence.

Asset	Official source	Licence
Datasets		
M4 (Wang et al., 2024b)	https://github.com/mbzuai-nlp/M4	No specific licence
CUDRT (Tao et al., 2024)	https://github.com/TaoZhen1110/CUDRT	No specific licence
GRiD (Qazi et al., 2024)	https://github.com/madlab-ucr/GriD	No specific licence
MAGE (Li et al., 2024)	https://github.com/yafuly/MAGE	Apache-2.0
MixSet (Zhang et al., 2024)	https://github.com/Dongping-Chen/MixSet	No specific licence
Beemo (Artemova et al., 2025)	https://github.com/Toloka/beemo	MIT, CC-BY-NC-4.0
BUST (Cornelius et al., 2024)	https://github.com/IDSIA-NLP/BUST	MIT

Table 14: Official sources and licence information for pretrained models and comparison methods.

Asset	Official source	Licence
Pretrained models		
RoBERTa Base (Liu et al., 2019)	https://huggingface.co/FacebookAI/roberta-base	MIT
Llama 3.2 1B	https://huggingface.co/meta-llama/Llama-3.2-1B	Llama 3.2 Community License
Llama 3.2 1B Instruct	https://huggingface.co/meta-llama/Llama-3.2-1B-Instruct	Llama 3.2 Community License
Llama 3.2 3B (Grattafiori et al., 2024)	https://huggingface.co/meta-llama/Llama-3.2-3B	Llama 3.2 Community License
Llama 3.2 3B Instruct (Grattafiori et al., 2024)	https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct	Llama 3.2 Community License
Qwen2.5-1.5B (Qwen, 2025)	https://huggingface.co/Qwen/Qwen2.5-1.5B	Apache-2.0
Comparison detectors and implementations		
IMGTB (Spiegel & Macko, 2023)	https://github.com/kinit-sk/IMGTB	MIT
Binoculars (Hans et al., 2024)	https://github.com/ahans30/Binoculars	BSD-3-Clause
ChatGPT-QA (Guo et al., 2023)	https://huggingface.co/Hello-SimpleAI/chatgpt-detector-roberta	No specific licence
LLMDet (Wu et al., 2023)	https://github.com/TrustedLLM/LLMDet	MIT
LoRA E5-small	https://huggingface.co/MayZhou/e5-small-lora-ai-generated-detector	MIT
OpenAI Detector (Solaiman et al., 2019)	https://huggingface.co/openai-community/roberta-base-openai-detector	MIT
RADAR (Hu et al., 2023)	https://huggingface.co/TrustSafeAI/RADAR-Vicuna-7B	No specific licence

Table 15: Official sources and licence information for software.

Asset	Official source	Licence
PyTorch	https://github.com/pytorch/pytorch	BSD-3-Clause
Lightning	https://github.com/Lightning-AI/pytorch-lightning	Apache-2.0
Transformers	https://github.com/huggingface/transformers	Apache-2.0
PEFT	https://github.com/huggingface/peft	Apache-2.0
Datasets	https://github.com/huggingface/datasets	Apache-2.0
Evaluate	https://github.com/huggingface/evaluate	Apache-2.0
bitsandbytes	https://github.com/bitsandbytes-foundation/bitsandbytes	MIT
scikit-learn (Pedregosa et al., 2011)	https://github.com/scikit-learn/scikit-learn	BSD-3-Clause
XGBoost (Chen & Guestrin, 2016)	https://github.com/dmlc/xgboost	Apache-2.0
NumPy	https://github.com/numpy/numpy	BSD-3-Clause
Pandas	https://github.com/pandas-dev/pandas	BSD-3-Clause
Matplotlib	https://github.com/matplotlib/matplotlib	Matplotlib License
Seaborn	https://github.com/mwaskom/seaborn	BSD-3-Clause
tqdm	https://github.com/tqdm/tqdm	MPL-2.0, MIT