

On the Generalizability of AI-Generated Text Detection

by

Amir David

A thesis
presented to the University Of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2026

© Amir David 2026

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

As large language models (LLMs) become ubiquitous, reliably distinguishing their outputs from human writing is critical for academic integrity, content moderation, and preventing model collapse from synthetic training data. This thesis examines the generalizability of LLM-text detectors across evolving model families and domains. We compiled a comprehensive evaluation dataset from commonly-used human corpora and generated corresponding samples using recent OpenAI and Anthropic models spanning multiple generations. Comparing the state-of-the-art zero-shot detector (Binoculars) against supervised RoBERTa/DeBERTa classifiers, we arrive at four main findings. First, zero-shot detection fails on newer models. Second, supervised detectors maintain high TPR in-distribution but exhibit asymmetric cross-generation transfer. Third, commonly reported metrics such as AUROC can obscure poor performance at deployment-relevant thresholds: detectors achieving high AUROC yield near-zero TPR at low FPR, and existing low-FPR evaluations often lack statistical reliability due to small sample sizes. Fourth, through tail-focused training and calibration, we reduce FPR by up to $4\times$ (from $\sim 1\%$ to $\sim 0.25\%$) while maintaining 90% TPR. Our results suggest that robust detection requires continually recalibrated, model-aware pipelines rather than static universal detectors.

Acknowledgements

I would like to thank my family for always being there for me, and supporting me throughout my life. I could not have done this without you.

I would also like to thank my supervisor, Professor Florian Kerschbaum, for his continued guidance throughout this journey.

Lastly, I also want to thank all the friends I made along the way. Each and every one of you has left their mark on me.

Table of Contents

Author’s Declaration	ii
Abstract	iii
Acknowledgements	iv
List of Tables	vii
1 Introduction	1
2 Background	4
3 Related Works	7
3.1 Supervised Detectors	7
3.2 Zero-Shot Detectors	8
3.3 Detection Benchmarks	9
4 Methodology	11
4.1 Evaluation Objectives and Overview	11
4.2 Experimental Setup	13
4.2.1 Language Models	13
4.2.2 Datasets	13

4.2.3	Data Generation	14
4.3	Zero-shot Detection Pipeline	15
4.4	Supervised Detection Pipeline	16
4.4.1	Model Training	16
4.4.2	Calibration	19
4.4.3	Evaluation	21
5	Results	23
5.1	Summary of Key Findings	23
5.2	Supervised Detection vs. Zero-Shot (RQ1)	24
5.3	Cross-Model Generalization (RQ2)	25
5.3.1	Within-Family Transfer	26
5.4	Universal Detectors (RQ3)	27
5.4.1	Performance at Fixed Low FPR (0.1%)	27
5.4.2	Performance at Fixed High TPR (90%)	29
5.5	Seed Instability and Data Constraints	30
6	Discussion	32
7	Limitations	34
8	Future Work	36
9	Conclusion	38
	References	40
	APPENDICES	48
A	Additional info	49

List of Tables

5.1	Performance of the fine-tuned RoBERTa-base detector against the SOTA zero-shot Binoculars detector on OpenAI models. We report the True Positive Rate (TPR) at a fixed False Positive Rate (FPR) of 1%.	24
5.2	Performance of the fine-tuned RoBERTa-base detector against the SOTA zero-shot Binoculars detector on Anthropic models. We report the True Positive Rate (TPR) at a fixed False Positive Rate (FPR) of 1%.	25
5.3	Cross-model transferability of the fine-tuned RoBERTa-base detector on OpenAI models . Each column corresponds to the LLM used for training, while each row corresponds to the LLM used for testing. We report the mean True Positive Rate (TPR) at 1% False Positive Rate (FPR), averaged over 10 random seeds. The diagonal (gray) indicates detection performance of a model on its own data, while off-diagonal cells show transfer performance.	26
5.4	Cross-model transferability of the fine-tuned RoBERTa-base detector on Anthropic models . Each column corresponds to the LLM used for training, while each row corresponds to the LLM used for testing. We report the mean True Positive Rate (TPR) at 1% False Positive Rate (FPR), averaged over 10 random seeds. The diagonal (gray) indicates detection performance of a model on its own data, while off-diagonal cells show transfer performance.	26
5.5	Universal OpenAI detector performance at a fixed human FPR of 0.1%. Entries are mean \pm std across 3 seeds; range shows best–worst seed TPR.	28
5.6	Universal Anthropic detector performance at a fixed human FPR of 0.1%. Entries are mean \pm std across 3 seeds; range shows best–worst seed TPR.	28
5.7	Universal detector for both model families performance at a fixed human FPR of 0.1%. Entries are mean \pm std across 3 seeds; range shows best–worst seed TPR.	29

5.8	OpenAI universal detector: FPR at TPR = 90% (worst-case across human pools). Entries are mean \pm std across 3 seeds; range shows best–worst seed FPR.	29
5.9	Anthropic universal detector: FPR at TPR = 90% (worst-case across human pools). Entries are mean \pm std across 3 seeds; range shows best–worst seed FPR.	30
5.10	Universal detector for both model families: FPR @ TPR = 90% (worst-case across human pools). Entries are mean \pm std across 3 seeds; range shows best–worst seed FPR.	30
6.1	Performance of the fine-tuned RoBERTa-base detector on OpenAI models, reported using AUROC (%). This table uses the same setup and data as Table 5.1, but replaces TPR with AUROC.	32
6.2	Performance of the fine-tuned RoBERTa-base detector on OpenAI models, reported using accuracy (%). Same detectors and data as Table 6.1.	32
6.3	Performance of the zero-shot Binoculars detector on OpenAI models, reported using AUROC (%). This table corresponds to the Binoculars rows in Table 5.1, but replaces TPR with AUROC.	33
6.4	Performance of the zero-shot Binoculars detector on OpenAI models, reported using accuracy (%). Same detectors and data as Table 6.3.	33

Chapter 1

Introduction

Large Language Models (LLMs) are AI systems capable of generating high-quality text and performing a wide range of linguistic tasks, including summarization, question answering, information retrieval, and code generation, at scale. The deployment of these systems has accelerated rapidly in recent years, particularly following the public release of ChatGPT in late 2022 [1]. As of October 2025, OpenAI CEO Sam Altman reported that ChatGPT had reached 800 million weekly active users [2][3]. The Stanford 2025 Artificial Intelligence Index Report indicates that 78% of organizations used AI in 2024 [4], while a survey by the UK's Higher Education Policy Institute found that 92% of students use AI tools in their studies [5].

As LLMs become increasingly integrated into daily workflows, from document drafting and email composition to software development, their outputs have grown harder to distinguish from human writing. This challenge is compounded by evidence that humans perform poorly at identifying AI-generated text. For instance, one study found that participants correctly identified AI-written excerpts from German theses only 57% of the time, barely above chance [6]. Even automated detection remains difficult: OpenAI's now-discontinued classifier achieved a true positive rate of only 26% in some cases, while misclassifying 9% of human-written text as AI-generated, with particularly poor performance on texts shorter than 1,000 characters [7]. Despite these difficulties, theoretical work suggests that reliable detection is possible in principle as long as the distributions of human-written and AI-generated text are not identical [8]. This result motivates the search for practical detection methods that are robust to evolving models and usage patterns.

The ability to reliably distinguish between human-written and AI-generated text is crucial for several reasons. First, it safeguards integrity across multiple domains. Academic

integrity is threatened when students submit AI-generated work as their own [9][10][11]. Information integrity is undermined by automated campaigns that leverage AI to spread political propaganda and health misinformation, or manipulate markets at scale [12][13][14][15][16]. Even scientific integrity is at risk, with cases of researchers publishing fully AI-generated papers [17][18]. Digital trust erodes when AI is used to generate fake product reviews, fabricated social media comments, or large volumes of spam [19][20][21].

Second, detection supports authenticity and accountability. In education, institutions must verify that submitted work reflects genuine student learning and critical thinking. In professions such as law, medicine, journalism, and software engineering, questions arise about who bears responsibility for errors introduced through AI use. In creative industries, concerns mount over AI systems trained on copyrighted material generating content that plagiarizes original works [22].

Third, detection is essential for data hygiene in AI development itself. As synthetic content proliferates online, there is growing risk that future LLMs will be trained on AI-generated text, creating feedback loops that may lead to model collapse, a degradation in model quality and diversity [23][24].

These concerns are not merely hypothetical; they already manifest in concrete incidents across domains. In early 2025, the Japanese company Sakana AI showed that a fully AI-generated scientific paper could pass double-blind peer review and be accepted to a workshop at ICLR, one of the premier machine learning conferences [25]. In July 2025, a federal judge in Alabama disqualified three lawyers from a case after they submitted court filings containing fabricated citations generated by AI [26]. Similar patterns appear in the information domain: in 2024, OpenAI reported removing coordinated operations by foreign actors using its models to generate multilingual social media posts and articles [27]. Around the same time, Meta dismantled a network using AI-generated comments, posing as local citizens and targeting posts by news outlets and U.S. lawmakers [28]. In early 2025, Reuters documented a Russia-linked network of AI-content websites targeting German audiences with misleading narratives ahead of national elections [29].

Given the growing deployment of LLMs and the documented misuse of AI-generated text, this thesis sets out to evaluate the generalizability of existing LLM detection methods. Existing approaches to detection are often grouped into zero-shot detectors, which make predictions without task-specific training data, typically using statistical features of the text, and supervised detectors, which are explicitly trained on labeled examples of human and AI-generated text. While a number of detectors have been proposed, existing work provides only a partial picture of how these systems behave when faced with newer or previously unseen LLMs. We conduct an empirical analysis of state-of-the-art detectors

on text generated by both established and recent LLMs across multiple domains. Our investigation is guided by the following research questions:

RQ1: How do zero-shot and supervised detectors compare in detection performance and generalizability on text from both seen and unseen LLMs?

RQ2: For supervised detectors, how does detection capability transfer within a model family?

RQ3: How does a universal detector trained on multiple models perform compared to specialized detectors trained on individual models?

Beyond addressing these questions, our analysis reveals significant methodological concerns in how AI-generated text detection is currently evaluated. We show that common choices of metrics and experimental setups can lead to overly optimistic or misleading conclusions about detector robustness, particularly under model and domain shift. We discuss these issues in detail and propose more informative evaluation practices.

Chapter 2

Background

Large language models (LLMs) are neural networks trained on massive text corpora to predict the next token in a sequence. When prompted, they can generate human-like text and can be applied to tasks such as document summarization, question answering, information retrieval, code generation, and more.

LLM Detectors are systems capable of systematically distinguishing between human-written and machine (LLM) generated text, typically outputting a confidence score.

Zero-shot Detectors are a type of LLM detectors that make predictions without any task-specific labeled examples (hence “zero-shots”), typically using statistical properties of the text.

Supervised Detectors are a type of LLM detectors that are typically a classifier that is trained on human-written and machine-generated samples using supervised learning.

Perplexity is a measure of how surprising a text is to a language model. Given a sequence of tokens $w_{1:N}$ and a model p , its perplexity is

$$\text{PPL}(w_{1:N}) = \exp \left(-\frac{1}{N} \sum_{i=1}^N \log p(w_i | w_{<i}) \right)$$

so lower perplexity means the text is less surprising to the model.

Binoculars [30] is a zero-shot detector that works by contrasting the perplexity of a text under two different but similar language models. It defines a cross-perplexity metric based on the premise that machine-generated text tends to have lower perplexity (that is, it is less surprising) than human-written text to certain LLMs.

True positive rate (TPR) measures the fraction of correctly detected AI-generated texts out of all AI-generated texts. It is defined as

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where TP and FN denote the numbers of true positives and false negatives, respectively.

False positive rate (FPR) measures the fraction of human-written texts that the detector incorrectly classifies as AI-generated. It is defined as

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

where FP and TN denote the numbers of false positives and true negatives.

Area under the receiver operating characteristic curve (AUROC) summarizes the trade-off between TPR and FPR across all possible decision thresholds. It is equal to the probability that a randomly chosen AI-generated text is assigned a higher detection score than a randomly chosen human-written text.

Detection score is the scalar value produced by an LLM detector to indicate how AI-like a text appears. In this thesis we often define the score as the difference between the model’s logits for the LLM and human classes, so that higher scores correspond to text that is more likely to be LLM-generated. A decision threshold on this score is used to classify each text as human or AI-generated.

Accuracy is the overall fraction of correctly classified texts, combining both human and AI-generated examples:

$$\text{Acc} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}.$$

Specialized Detectors are supervised detectors trained on text from a single target LLM (together with human-written text). For example, a detector may be trained only on GPT-4o outputs and humans.

Universal Detectors are supervised detectors trained on pooled data from multiple LLMs and humans, with the goal of detecting a wider range of generation sources using a single model.

Tail-focused Fine-Tuning refers to an additional training stage that puts extra emphasis on examples near the decision boundary, especially (i) human texts that currently

look suspiciously AI-like and (ii) LLM texts that look very human-like. The goal is to improve the detector’s behavior in the extreme tails of the score distribution, where low-FPR operation is determined.

Document-max (docmax) Window Selection is a training strategy for long documents in which each document is split into several overlapping windows, the detector scores each window, and the single window with the highest (most AI-like) score is used as the training instance. This focuses learning on the most suspicious part of each document.

Window-level tail probability is, in our calibration procedure, the fraction of human windows in a given length bin whose detection score is at least as large as the score of a particular window. Formally, for a window w in bin $b(w)$ with score $S(w)$, we define

$$p(w) \triangleq \Pr_{\text{human}} (S(W) \geq S(w) \mid b(W) = b(w)).$$

Intuitively, $p(w)$ measures how unusually “AI-like” w is compared to typical human writing of similar length. We treat this as an empirically calibrated tail probability (analogous to a one-sided p -value), rather than a formal hypothesis-test p -value.

Document-level score in our pipeline is obtained by aggregating window-level tail probabilities across the windows of a document. We take the minimum tail probability across windows and transform it into a score S_{doc} (e.g., via $-\log_{10} p_{\text{min}}$), so that larger scores correspond to documents whose most suspicious window is very unlikely under the empirical human distribution. This document-level score is then thresholded to classify the whole document.

Chapter 3

Related Works

Broadly speaking, detection methods can be categorized into two paradigms: supervised detectors that require training a classifier on a set of samples of human and machine-generated text, and zero-shot methods that operate without access to such training data, often by leveraging statistical properties of the generated text or the source model itself.

3.1 Supervised Detectors

Supervised detection frames the problem as a binary or multi-class classification task. The dominant approach involves fine-tuning a pre-trained transformer model, such as RoBERTa [31], to distinguish between human and machine-written text.

A straightforward application of this paradigm is presented in LLM-DetectAIve [32], which fine-tunes transformer models to perform fine-grained, four-way classification, identifying text as purely human, purely machine, machine-generated but humanized, or human-written and machine-polished. This work highlights the need for more nuanced detection beyond a simple binary choice.

Other supervised approaches focus on engineering discriminative features rather than end-to-end fine-tuning. In Ghostbuster [33], the authors propose a method that passes documents through a series of weaker language models to extract a feature set, over which a linear classifier is then trained. Similarly, GPT-who [34] proposes a statistical detector that trains a logistic regression classifier on features derived from the Uniform Information Density (UID) principle, a psycholinguistic hypothesis about how humans distribute information during language production.

To improve the robustness of supervised detectors against evasive paraphrasing or adversarial attacks, researchers have explored adversarial training frameworks. In RADAR [35], a paraphraser and a detector are jointly trained in a two-player game, where the paraphraser learns to rewrite MGT to evade detection, and the detector learns to identify both original and paraphrased MGT. A different approach, OUTFOX [36], improves detector robustness by using in-context learning. An "attacker" LLM is prompted with examples of a detector's predictions to generate adversarial essays that are harder to detect, which are then used as few-shot examples to strengthen the detector against such attacks. Although powerful, these supervised methods inherently risk overfitting to the specific models and domains seen during training, potentially limiting their generalizability to novel, unseen LLMs.

3.2 Zero-Shot Detectors

Zero-shot detectors aim to overcome the generalizability limitations of supervised methods by avoiding training on text from the target LLM. These methods often rely on the hypothesis that text sampled from an LLM exhibits distinct statistical characteristics compared to human-written text.

A prominent line of work analyzes the log-probability function of the source LLM. In DetectGPT [37], it is proposed that MGT typically lies in regions of negative curvature of the model's log-probability function. The method detects this by measuring the drop in log-probability when the text is slightly perturbed by a mask-filling model (e.g., T5 [38]). Fast-DetectGPT [39] builds directly on this, proposing a more efficient formulation based on conditional probability curvature, which significantly accelerates the detection process. A different approach, DNA-GPT [40], proposes a "truncate-and-regenerate" method. It analyzes the divergence between the original text completion and new completions generated by an LLM from a truncated prefix, using N-gram analysis to quantify this difference.

Another family of methods contrasts the outputs of two different models. Binoculars [30] introduces a highly effective zero-shot detector that computes a score by contrasting the perplexity of a text under one model (an "observer") with the cross-perplexity between the observer and a slightly different "performer" model. This ratio proves to be a robust signal that normalizes for content-specific difficulty and isolates the statistical signature of machine generation.

Shifting from probabilistic to geometric properties, some work examines the manifold of text embeddings. In Intrinsic Dimension Estimation [41], the authors propose that

the manifold of contextual embeddings for human-written text has a consistently higher intrinsic dimension than that of AI-generated text. They propose a detector based on estimating this property using persistent homology, offering a novel, training-free signal for detection.

Our work seeks to systematically evaluate the trade-off in generalizability between these two paradigms, analyzing how supervised and zero-shot detectors perform as the target LLMs they are designed to detect continue to evolve.

3.3 Detection Benchmarks

As the number of detection methods grows, several benchmarks have been developed to systematically evaluate their performance, robustness, and generalizability. These benchmarks are critical for understanding the current state-of-the-art (SOTA) and identifying the most promising detection paradigms.

Early survey papers and benchmarks such as A Survey on LLM-Generated Text Detection [42] and MGTBench [43] provided a structured overview of the landscape, categorizing detectors, and evaluating them on curated datasets. For instance, MGTBench found that a supervised "LM Detector" (a fine-tuned BERT model) generally outperformed metric-based approaches on their tested domains, establishing a strong baseline for supervised methods.

More recent benchmarks have focused on more challenging and realistic evaluation settings, particularly adversarial robustness and out-of-domain generalization. The RAID [44] benchmark, one of the largest to date, evaluates detectors against a wide array of adversarial attacks, decoding strategies, and unseen models. Their extensive evaluation confirms that, while many detectors perform well in ideal conditions, their accuracy degrades significantly under adversarial pressure. Similarly, DetectRL [45] introduces a benchmark focused on simulating real-world scenarios, including human revisions and varied prompt strategies.

A consistent finding across these recent, rigorous benchmarks is the emergence of clear top performers in both the supervised and zero-shot categories.

1. The DetectRL leaderboard shows that supervised RoBERTa-based models achieve the highest overall F1 scores and that Binoculars is the top-performing zero-shot method, significantly outperforming other statistical approaches like Log-Rank and DetectGPT.

2. The RAID benchmark corroborates this, finding that RoBERTa-based classifiers and Binoculars are the most robust detectors in their respective paradigms against a battery of adversarial attacks and decoding variations.
3. The Beemo benchmark [46], which focuses on the nuanced task of detecting expert-edited and LLM-human collaborative text, also identifies Binoculars as the most effective zero-shot detector across its varied multi-author scenarios.

Across these evaluations, a consensus emerges. Within the supervised paradigm, fine-tuned RoBERTa models remain the SOTA, offering high accuracy when the training distribution is representative of the test case. In the zero-shot paradigm, Binoculars consistently demonstrates superior performance and robustness compared to other statistical methods, particularly in adversarial and out-of-domain settings. This consensus in the literature justifies our selection of a fine-tuned RoBERTa model and Binoculars as the representative SOTA baselines for evaluating the supervised and zero-shot paradigms, respectively, in our comparative analysis of detector generalizability.

Chapter 4

Methodology

This chapter describes our methodology for evaluating the generalizability of LLM-generated text detectors. As established in Chapter 3, recent benchmarks consistently identify fine-tuned RoBERTa/DeBERTa models and Binoculars as the state-of-the-art representatives of the supervised and zero-shot paradigms, respectively; we therefore adopt these as our primary detection methods. We first outline the models and datasets used in our experiments, then detail our data generation process, the zero-shot detection pipeline, and the supervised detection pipeline.

4.1 Evaluation Objectives and Overview

Our empirical study is designed to answer the three research questions stated in Chapter 1:

- **RQ1:** How do zero-shot and supervised detectors compare in detection performance and generalizability on text from both seen and unseen LLMs?
- **RQ2:** For supervised detectors, how does detection capability transfer within a model family?
- **RQ3:** How does a universal detector trained on multiple models perform compared to specialized detectors trained on individual models?

To address these questions, we conduct a set of coordinated evaluations built on two classes of detectors:

1. A *zero-shot* detector based on the Binoculars method of Hans et al. [30], instantiated with pairs of open-source LLMs (Section 4.3).
2. A *supervised* detector based on the RoBERTa encoder fine-tuned on labeled human vs. LLM-generated text, which is the method implemented in DetectRL [45] for the initial baseline experiments (section 4.4).
3. A *supervised* detector based on a DeBERTa-v3-large using an advanced training and calibration techniques for the low-FPR experiments (Section 4.4).

Across these detectors, we run the following experimental suites:

Baseline comparison (RQ1). We first establish baseline performance by evaluating (i) Binoculars in its default configuration (Falcon-7B / Falcon-7B-Instruct) and with alternative modern LLM pairs, and (ii) a supervised RoBERTa-base detector closely following the original DetectRL pipeline [45], trained on balanced data for each target generator. This baseline uses a single-stage fine-tuning setup with dev-based threshold selection at approximately 1% FPR. In later sections we introduce a stronger DeBERTa-v3-based pipeline with tail-focused training and document-level calibration for operation at 0.1% FPR. Both detectors are evaluated on held-out text from the same generators as used in prior work as well as on newer, stronger models (Sections 4.2–4.3).

Generalization within families (RQ1, RQ2). Using the supervised pipeline, we train *specialized detectors* on text from a single LLM (e.g., one model within the OpenAI or Anthropic family) and evaluate them on: (i) held-out text from the same model, (ii) and other models in the same family. This directly quantifies how well supervised detectors transfer within model families under distribution shift.

Universal vs. specialized detectors (RQ1, RQ3). We then train *universal detectors* on pooled data from multiple LLMs spanning both families, using the same supervised training and calibration pipeline. We compare these to specialized detectors on the same test sets to assess whether a single detector can match or exceed the performance of per-model detectors when constrained to a fixed false positive rate.

The remainder of this chapter details the models and datasets used (Section 4.2), the zero-shot Binoculars pipeline (Section 4.3), and the supervised training, calibration, and evaluation procedures (Sections 4.4–4.4.3).

4.2 Experimental Setup

4.2.1 Language Models

We evaluate detectors on text generated by two major families of proprietary large language models, spanning multiple generations. All models are accessed through their respective APIs.

OpenAI family. We include four models from the OpenAI family: GPT-3.5-Turbo (March 2023) [47], GPT-4o (May 2024) [48], GPT-4.1 (April 2025) [49], and o4-mini (April 2025) [50].

Anthropic family. We include four models from the Anthropic family: Claude-3-Opus (March 2024) [51], Claude-3.5-Sonnet (June 2024) [52], Claude-3.7-Sonnet (February 2025) [53], and Claude-4-Sonnet (May 2025) [54].

4.2.2 Datasets

To cover diverse writing styles, we use human-written text from three domains:

- **Creative writing:** Stories and narratives sourced from the WritingPrompts community on Reddit [55][56].
- **Academic writing:** Scientific articles from the arXiv dataset [57].
- **News writing:** Journalistic articles from the BBC, utilizing the Extreme Summarization (XSum) dataset [58].

These domains were selected for two reasons. First, they align with established detection benchmarks: Binoculars [30] evaluates on creative writing (WritingPrompts) and news, while DetectRL [45] uses XSum, arXiv, and WritingPrompts. This alignment facilitates comparison with prior work. Second, the three domains capture stylistically distinct writing styles, providing a meaningful test of detector generalization across the styles.

For each domain, these human-written texts serve both as negative examples for detection and as prompts to elicit LLM-generated text, as described in Section 4.2.3.

4.2.3 Data Generation

Prompting strategy. For each human-written document in our corpus, we construct a prompt and query each LLM to generate a corresponding completion in the same domain and similar length range. Concretely, we construct prompts from associated metadata—the article title for academic writing, the WritingPrompts prompt for creative writing, or the reference summary for news. We prepend a task-specific system instruction describing the generation task.

Preprocessing. All texts undergo unified preprocessing: (1) Unicode normalization via `ftfy`, (2) removal of boilerplate prefixes and wrapper quotes, (3) whitespace normalization, and (4) For supervised detectors, model inputs are scored using 256-token windows (maximum sequence length 256). Longer texts may yield multiple overlapping windows. This aggressive normalization prevents detectors from exploiting formatting shortcuts rather than genuine linguistic differences. (See Appendix A for details). To avoid contamination between supervised training data and our human-only calibration/evaluation pools, we exclude any human text used in paired training/test datasets when constructing $H_{\text{calib_pool}}$ and $H_{\text{test_pool}}$. Human negatives are sampled from disjoint documents from the same underlying corpora.

Dataset sizes. For each target LLM and each domain, we construct a balanced dataset with 50% human-written and 50% LLM-generated examples. For most OpenAI models we use approximately $N \approx 4000$ labeled examples per domain; for GPT-4.1 we use $N = 667$ per domain; and for o4-mini and all Anthropic models we use $N = 800$ per domain. OpenAI experiments use three domains (arXiv, XSum, WritingPrompts), while Anthropic experiments use two domains (arXiv and WritingPrompts). Each per-model dataset is then split into train/dev/test subsets using a stratified procedure.

Human calibration and evaluation pools. In addition to the training data, we maintain large human-only pools for calibration and evaluation of the supervised detector. These pools contain millions of documents across all domains and are never mixed with LLM-generated text. They enable statistically reliable estimation of false positive rates (FPR) at stringent levels (e.g., 0.1%) and are used in the calibration and FPR validation procedures.

4.3 Zero-shot Detection Pipeline

Our zero-shot evaluation is built around the *Binoculars* detector introduced by Hans et al. [30], which proposes *cross-perplexity* as a signal for LLM-generated text. Let a string s be tokenized into indices $\vec{x} = (x_1, \dots, x_L)$ by a tokenizer T with vocabulary V . Given s as input, a language model M produces next-token probabilities $\mathcal{M}(s) \in [0, 1]^{L \times |V|}$, where the i -th row $\mathcal{M}(s)_i$ is a distribution over V . The log-perplexity (average negative log-likelihood) of s under M is

$$\log \text{PPL}_M(s) = -\frac{1}{L} \sum_{i=1}^L \log \mathcal{M}(s)_{i, x_i}. \quad (4.1)$$

Binoculars uses a pair of closely related LMs, M_1 and M_2 , based on the intuition that cross-perplexity (which measures how much M_1 's predictions disagree with M_2 's) behaves differently for human vs. LLM text. Formally, the *cross-perplexity* of M_2 as judged by M_1 is defined by:

$$\log \text{X-PPL}_{M_1, M_2}(s) = -\frac{1}{L} \sum_{i=1}^L \mathcal{M}_1(s)_i^\top \log \mathcal{M}_2(s)_i, \quad (4.2)$$

i.e., the average cross-entropy between the predictive distributions of M_1 and M_2 on the same contexts. The *Binoculars score* is then the ratio

$$B_{M_1, M_2}(s) = \frac{\log \text{PPL}_{M_1}(s)}{\log \text{X-PPL}_{M_1, M_2}(s)}. \quad (4.3)$$

Intuitively, $\log \text{PPL}_{M_1}$ measures how surprising s is to M_1 , while $\log \text{X-PPL}_{M_1, M_2}$ measures how surprising the token predictions of M_2 are when graded by M_1 . Because LLMs are typically more similar to each other than to human writing, Hans et al. show that for suitable pairs (M_1, M_2) the ratio $B_{M_1, M_2}(s)$ tends to be smaller for LLM-generated text. A text is classified via a threshold τ as

$$\hat{y}(s) = \begin{cases} \text{LLM-generated,} & B_{M_1, M_2}(s) < \tau, \\ \text{human-generated,} & B_{M_1, M_2}(s) \geq \tau. \end{cases} \quad (4.4)$$

In our experiments we follow the official open-source implementation¹ of *Binoculars*, which realizes $\log \text{PPL}_{M_1}$ and $\log \text{X-PPL}_{M_1, M_2}$ using masked token-wise cross-entropy losses and provides reference thresholds calibrated for the Falcon-7B / Falcon-7B-Instruct pair.

¹<https://github.com/ahans30/Binoculars>

Extending Binoculars to modern LLMs. We evaluate Binoculars with a variety of contemporary LLMs (Falcon, LLaMA, Gemma, Qwen, etc.), including very large models that are sharded across multiple GPUs and/or quantized. To make the original implementation robust in this setting, we introduce several engineering modifications that leave the underlying score definition B_{M_1, M_2} unchanged. See Appendix A for details.

4.4 Supervised Detection Pipeline

Our supervised detection pipeline consists of three stages: (1) model training, (2) calibration, and (3) evaluation. In our initial baseline experiments (Sections 5.2 and 5.3), we also train a simpler RoBERTa-base detector following DetectRL [45]; our main low-FPR experiments employ the full two-stage DeBERTa-v3 pipeline described below. We focus on achieving very low *false positive rate* (FPR), defined as the fraction of human texts incorrectly classified as LLM-generated, while maintaining a high *true positive rate* (TPR), the fraction of LLM-generated texts correctly detected. This objective aligns with the Neyman–Pearson classification paradigm: we first fix a maximum acceptable rate of false positives (type I errors, i.e., human texts wrongly flagged as AI-generated), then maximize the rate of true positives (correctly detected AI text) subject to that constraint. [59]. Prior work has documented problematic false positive behavior in existing AI-text detectors, particularly in educational settings [60], motivating our emphasis on stringent FPR control.

4.4.1 Model Training

We adopt a two-stage training procedure. Stage 1 performs standard supervised fine-tuning to obtain a strong baseline detector. Stage 2 refines the model with a focus on the decision boundary, targeting the tail of the human score distribution to achieve reliable performance at low FPR. This progression from general to boundary-focused training is inspired by curriculum learning strategies that adapt the training distribution over time [61].

Throughout, we represent the detector as a neural network f_θ producing logits over the two classes (human vs. LLM). For any text x , we define a scalar score

$$S(x) \triangleq f_\theta(x)_{\text{LLM}} - f_\theta(x)_{\text{human}}, \tag{4.5}$$

so that larger $S(x)$ indicates more LLM-like text.

Stage 1: Standard Fine-Tuning

For our main low-FPR pipeline, we initialize from Microsoft’s DeBERTa-v3-large model [62, 63], a transformer-based encoder with strong performance on natural language understanding benchmarks. Inputs are tokenized with a maximum sequence length of 256 tokens. In our initial baseline experiments, we also train an otherwise identical RoBERTa-base model with the same Stage 1 setup, to mirror the DetectRL configuration [45] and enable direct comparison.

For each random seed, we split the training pool into training and development sets. We train the model using standard cross-entropy loss over the human/LLM label, with a batch size of 16, for 5 epochs. This stage yields a baseline detector that achieves reasonable separation between human and LLM-generated text but is not yet optimized for very low FPR.

Stage 2: Tail-Focused Fine-Tuning

Stage 2 addresses the challenge of maintaining high TPR while controlling FPR at very low thresholds (e.g., 0.1%). Standard training optimizes average-case performance but does not focus on the critical region near the decision boundary, where most errors occur. Stage 2 introduces three categories of modifications: (1) mining strategies that identify and oversample difficult examples [64], (2) specialized loss functions that penalize errors in the tail [65], and (3) training adjustments that stabilize learning. We describe each in turn.

Per-bin threshold calibration for training. We first calibrate length-specific thresholds using human samples only. We partition texts into token-length bins:

$$\mathcal{B} = \{< 60, 60\text{--}100, 100\text{--}160, 160\text{--}200, 200\text{--}256\}.$$

For each bin $b \in \mathcal{B}$, we score human texts and choose a threshold τ_b so that only a small fraction of human texts in that bin have scores $S(x)$ above τ_b . Concretely, we sort all human scores in bin b and pick the value such that roughly α_{target} of them lie above it. When we have many examples in a bin we also experiment with fitting a simple statistical model to the largest scores to smooth this estimate; if its prediction disagrees too much with the observed data, we ignore it and keep the directly estimated threshold. The resulting map $\tau_{\text{map}} : b \mapsto \tau_b$ tells us which humans in each length bin are in the extreme tail and will be treated as “hard” during Stage 2.

Hard negative mining. Following the principle of online hard example mining [64], we identify human samples that the current model finds most LLM-like. Within each (length bin, domain) cell, we select the top fraction (e.g., 5%) of humans with the highest scores $S(x)$. These “hard negatives” are replicated multiple times and added back into the training set, increasing the model’s exposure to challenging human examples and encouraging it to push down the right tail of the human score distribution.

Hard positive mining. Symmetrically, we emphasize LLM-generated samples that lie close to the decision boundary. We focus on long documents (200–256 tokens) and let τ_{long} denote the corresponding human threshold in this bin. For each domain, we compute the median LLM score in the 200–256 bin. Domains where this median is closer to the human threshold τ_{long} receive higher oversampling weights, since they contain more difficult examples. Within each domain, we oversample LLM texts whose scores fall in a small band around τ_{long} —that is, LLM texts that are plausibly confused with human writing.

Document-max window selection. To focus on the most suspicious portion of each document, we split each text into overlapping windows. We score all windows with the current model and select the window with the maximum score,

$$S_{\text{docmax}}(x) = \max_{w \in \text{windows}(x)} S(w),$$

as the training input. This can be viewed as a multiple-instance learning setup with max pooling over instances [66], focusing optimization on the most suspicious region of each text. At evaluation time we use a more principled document-level score based on calibrated tail frequencies over all windows; see Section 4.4.2.

Bin-aware batch sampling. We employ a custom batch sampler that ensures each mini-batch contains: (1) a fixed number of high-scoring human samples (which we call anchors) near τ_b in each length bin, (2) a fixed number of near-threshold LLM positives in each bin, and (3) random examples to fill the remaining capacity. This guarantees that every batch is enriched with tail examples near the decision boundary, where the specialized loss terms are most useful.

Tail-aware loss functions. Stage 2 augments the cross-entropy objective with a composite loss comprising three additional terms. These terms are related to focal loss [65], which reshapes cross-entropy to emphasize hard examples, and to pairwise ranking objectives [67]:

1. **Human tail penalty:** Human samples whose scores exceed a high within-batch quantile (e.g., the 98th percentile, up to an optional margin) are penalized, explicitly pushing down the right tail of the human score distribution and reducing false positives.
2. **Positive margin loss:** For LLM samples in bin b , we encourage scores to exceed $\tau_b + m$ for a margin $m > 0$, applied only to LLMs whose scores are not already far above τ_b . This prevents LLM scores from collapsing at the human threshold and stabilizes separation.
3. **Pairwise ranking loss:** For each bin, we select top- K human anchors and near-threshold LLM positives and apply a hinge loss enforcing

$$S(x^+) - S(x^-) \geq m,$$

where x^+ is an LLM example and x^- is a human anchor from the same bin. In words, we directly penalize the model whenever a hard human scores too close to or above a nearby LLM example, and push their scores farther apart.

Partial unfreezing. During Stage 2, we keep most encoder layers frozen and unfreeze only the classification head and the last K encoder layers (e.g., $K = 6$). Following the gradual unfreezing strategy of Howard and Ruder [68], we train for a small number of additional epochs with a reduced learning rate, using the composite loss. This preserves the robust representations learned in Stage 1 while adapting the model to tail-focused objectives.

With a trained detector in hand, we next describe how to calibrate its outputs for statistically interpretable decision-making.

4.4.2 Calibration

After supervised training, we calibrate the detector to obtain statistically interpretable document-level scores and to select a global decision threshold. Crucially, calibration uses only human data. This choice is both practical and principled: human-written text is freely available at scale, whereas text generated from state-of-the-art LLMs requires costly API access. Moreover, since FPR is determined entirely by the human score distribution, a large human calibration pool suffices for statistically reliable threshold estimation at stringent levels (e.g., 0.1%).

Window-Level Scoring and Binning

For each human document in the calibration pool $H_{\text{calib_pool}}$, we split the text into overlapping windows and compute a score

$$s = S(w) = f_{\theta}(w)_{\text{LLM}} - f_{\theta}(w)_{\text{human}} \quad (4.6)$$

for each window w . Each window is assigned to a length bin $b(w) \in \mathcal{B}$ based on its token count.

Bin-Specific Tail Frequencies

For each length bin b , we build a simple lookup table that tells us, for any score s , what fraction of human windows in that bin have a score at least s . Formally, for each bin b we estimate

$$\text{Tail}_b(s) = \Pr_{\text{human}}(S(w) \geq s \mid b(w) = b), \quad (4.7)$$

by counting, among all human windows in bin b , the proportion whose score is at least s . We apply a small amount of smoothing to avoid zero counts when s is larger than any score seen in the calibration data. This gives a mapping from a raw score to a “how surprising” value:

$$p(w) = \text{Tail}_{b(w)}(S(w)), \quad (4.8)$$

which we interpret as an empirically calibrated tail probability (analogous to a one-sided p -value): it is the fraction of human windows in the same length bin that look at least as suspicious as w .

Document-Level Aggregation

We aggregate window-level p -values into a document-level score. For each document d , with windows w_1, \dots, w_K and corresponding p -values $p(w_1), \dots, p(w_K)$, we define

$$p_{\min}(d) = \min_k p(w_k), \quad S_{\text{doc}}(d) = -\log_{10}(p_{\min}(d)). \quad (4.9)$$

This MinP aggregation is analogous to a Tippett (MinP) combination rule [69]. Because our windows overlap (and are therefore dependent) and because the tail probabilities are estimated empirically from data, we treat S_{doc} as a calibrated detection statistic rather than a formal hypothesis test. Thus, $S_{\text{doc}}(d)$ measures how improbable the most suspicious window in d is under the human distribution: higher $S_{\text{doc}}(d)$ indicates a document that is less consistent with human writing.

Threshold Selection

Given a target human FPR α (e.g., $\alpha = 10^{-3}$ for 0.1%), we select a global threshold τ on S_{doc} using only human documents. We compute $S_{\text{doc}}(d)$ for all $d \in H_{\text{calib_pool}}$ and choose τ so that only about an α fraction of these human documents have $S_{\text{doc}}(d) \geq \tau$. Concretely, we sort all human S_{doc} values and pick the score level at which roughly α of them lie above it, taking care to handle ties in a conservative way. We then optionally lower τ slightly so that the realized FPR across both the calibration pool and a held-out human test pool remains below the target. No LLM-generated text is used in this step.

4.4.3 Evaluation

We evaluate each detector on held-out test sets for each target LLM family and validate its FPR on large human-only pools.

Test Sets and Scoring

For each LLM family, we construct a test set containing held-out human-written documents from the available domains (three for OpenAI; two for Anthropic), and LLM-generated documents matched by domain and approximate length.

For every test document d (human or LLM), we compute $S_{\text{doc}}(d)$ using the same windowing, binning, and aggregation procedure as in the Calibration Section. The bin-specific tail frequencies derived from human calibration data are held fixed.

Classification Rule

Given the calibrated threshold τ , we classify each document via

$$\hat{y}(d) = \begin{cases} \text{LLM}, & \text{if } S_{\text{doc}}(d) \geq \tau, \\ \text{human}, & \text{otherwise.} \end{cases}$$

Metrics and Diagnostics

We report the following metrics:

- **True Positive Rate (TPR):** Fraction of LLM-generated documents correctly classified ($\hat{y}(d) = \text{LLM}$).
- **False Positive Rate (FPR):** Fraction of human-written documents incorrectly flagged as LLM-generated ($\hat{y}(d) = \text{LLM}$).
- **TPR at fixed FPR:** TPR at the target human FPR level (e.g., 0.1%), to enable fair comparison across detectors.

FPR validation on large human pools. To ensure calibration quality, we apply the same threshold τ to both the calibration pool $H_{\text{calib_pool}}$ and the held-out human test pool $H_{\text{test_pool}}$ and report the maximum FPR observed across any individual human pool or domain.

These metrics provide conservative estimates of the detector’s true false positive behavior on real-world human-written text and verify that it maintains the target FPR across diverse human populations.

Chapter 5

Results

This chapter presents our empirical findings, organized around the research questions introduced in Chapter 1. We first compare zero-shot and supervised detectors on text from both established and recent LLMs (RQ1), then examine how supervised detectors transfer within model families (RQ2), and finally evaluate universal detectors trained on multiple LLMs (RQ3). Throughout, we report TPR at a fixed FPR to enable fair comparison under realistic operating conditions.

5.1 Summary of Key Findings

Our evaluation yields four principal findings:

1. **Zero-shot detection fails on newer LLMs.** Binoculars, the state-of-the-art zero-shot detector, achieves high TPR on older models (e.g., GPT-3.5) but collapses to near-zero TPR on recent models (e.g., GPT-4.1, o4-mini) in formal domains such as academic writing.
2. **Supervised detection remains robust.** A fine-tuned RoBERTa-based detector maintains $\geq 95\%$ TPR at 1% FPR across all tested LLMs, including the most recent releases, provided it is trained on data from the target model or a sufficiently similar one.
3. **Cross-model transfer is asymmetric and brittle.** Detectors trained on older LLMs often fail dramatically on newer models within the same family while detectors trained on newer LLMs show moderate transfer to older ones.

4. **Low-FPR operation is achievable but unstable.** Our tail-focused training pipeline can reduce FPR by up to 4× (from $\sim 1\%$ to $\sim 0.25\%$) while maintaining 90% TPR. However, performance at 0.1% FPR exhibits high variance across random seeds, reflecting sensitivity to limited training data and Stage-2 hyperparameters.

We also note that standard evaluation metrics such as AUROC can obscure poor performance at the low-FPR thresholds required for practical deployment. We discuss this issue further in Chapter 6.

5.2 Supervised Detection vs. Zero-Shot (RQ1)

We compare the supervised RoBERTa-based detector against the state-of-the-art zero-shot detector (Binoculars) across two model families and three text domains. For each detector, we report TPR at a fixed FPR of 1%.

Table 5.1: Performance of the fine-tuned **RoBERTa-base detector** against the SOTA zero-shot **Binoculars** detector on OpenAI models. We report the True Positive Rate (TPR) at a fixed False Positive Rate (FPR) of 1%.

Detector	Domain	GPT-3.5	GPT-4o	GPT-4.1	o4-mini
Supervised	Academic	99%	100%	100%	100%
	News	100%	100%	99%	100%
	Creative Writing	97%	97%	98%	97%
Zero-Shot	Academic	62%	0%	0%	0%
	News	41%	1%	9%	11%
	Creative Writing	99%	40%	27%	30%

For Anthropic models, we report results on Academic and Creative Writing domains only; News domain data is not available for these models.

Analysis. Tables 5.1 and 5.2 reveal a stark contrast between the two detection paradigms.

The **supervised detector** maintains near-perfect performance ($\geq 95\%$ TPR) across all models and domains, including the most recent releases (o4-mini, Claude-4-Sonnet). This robustness holds even though these models were released well after the detector’s base architecture (RoBERTa) was developed.

Table 5.2: Performance of the fine-tuned **RoBERTa-base detector** against the SOTA zero-shot **Binoculars** detector on Anthropic models. We report the True Positive Rate (TPR) at a fixed False Positive Rate (FPR) of 1%.

Detector	Domain	Claude 3 Opus	Claude 3.5 Sonnet	Claude 3.7 Sonnet	Claude 4 Sonnet
Supervised	Academic	100%	100%	99%	99%
	Creative Writing	99%	99%	97%	95%
Zero-Shot	Academic	49%	11%	5%	6%
	Creative Writing	94%	80%	62%	81%

The **zero-shot detector** exhibits severe degradation on newer LLMs. On academic text, Binoculars achieves 62% TPR on GPT-3.5 but drops to 0% on GPT-4o, GPT-4.1, and o4-mini. A similar pattern appears for Anthropic models: TPR falls from 49% (Claude-3-Opus) to 5–6% on newer releases. This collapse is consistent with the hypothesis that newer LLMs produce text whose statistical properties more closely resemble human writing, undermining the cross-perplexity signal that Binoculars relies upon.

Domain effects. Creative writing is notably easier for zero-shot detection across both families (99% on GPT-3.5, 62–94% on Claude models), likely because creative prose exhibits more stylistic variation that distinguishes human from LLM writing. In contrast, academic text appears hardest to detect in zero-shot settings, suggesting that modern LLMs have converged toward human-like formal exposition.

The strong zero-shot performance on GPT-3.5 creative writing (99%) may reflect the similarity between GPT-3.5 and the Falcon models used internally by Binoculars to compute cross-perplexity scores; older, smaller LLMs share more statistical regularities with Falcon than do frontier models.

Given the clear failure of zero-shot detection on newer LLMs, the remainder of our analysis focuses on supervised detectors.

5.3 Cross-Model Generalization (RQ2)

We investigate how well a supervised detector trained on one LLM generalizes to text from other LLMs, within a model family.

5.3.1 Within-Family Transfer

Table 5.3: Cross-model transferability of the fine-tuned RoBERTa-base detector on **OpenAI models**. Each column corresponds to the LLM used for training, while each row corresponds to the LLM used for testing. We report the mean True Positive Rate (TPR) at 1% False Positive Rate (FPR), averaged over 10 random seeds. The diagonal (gray) indicates detection performance of a model on its own data, while off-diagonal cells show transfer performance.

Tested on	Trained on			
	GPT-3.5	GPT-4o	GPT-4.1	o4-mini
GPT-3.5	97%	89%	65%	56%
GPT-4o	92%	98%	80%	76%
GPT-4.1	57%	81%	96%	96%
o4-mini	22%	53%	75%	98%

Table 5.4: Cross-model transferability of the fine-tuned RoBERTa-base detector on **Anthropic models**. Each column corresponds to the LLM used for training, while each row corresponds to the LLM used for testing. We report the mean True Positive Rate (TPR) at 1% False Positive Rate (FPR), averaged over 10 random seeds. The diagonal (gray) indicates detection performance of a model on its own data, while off-diagonal cells show transfer performance.

Tested on	Trained on			
	Claude 3 Opus	Claude 3.5 Sonnet	Claude 3.7 Sonnet	Claude 4 Sonnet
Claude 3 Opus	85%	58%	43%	48%
Claude 3.5 Sonnet	69%	91%	81%	87%
Claude 3.7 Sonnet	55%	79%	89%	92%
Claude 4 Sonnet	58%	85%	88%	96%

Tables 5.3 and 5.4 report within-family transfer for the RoBERTa-base baseline detectors. In both families, three consistent patterns emerge.

1. In-distribution performance is high. Along the diagonal, detectors achieve high TPR on the LLM they were trained on (typically $\geq 85\text{--}98\%$), confirming that supervised fine-tuning on a single model yields strong in-distribution detection at this operating point.

2. Transfer degrades with generational distance. Off-diagonal entries reveal a clear generational trend: detectors trained on older models transfer progressively worse to newer ones, and detectors trained on the newest models also lose performance when tested on the oldest. For example, in the OpenAI family a GPT-3.5-trained detector attains high TPR on GPT-3.5 but substantially lower TPR on GPT-4o, GPT-4.1, and especially o4-mini; conversely, an o4-mini-trained detector performs well on o4-mini but retains only moderate TPR on GPT-3.5. A similar pattern holds for the Anthropic family, where transfer is strongest between adjacent generations (e.g., Claude-3.5 \leftrightarrow 3.7, 3.7 \leftrightarrow 4) and weakest between the earliest and latest models (Claude-3-Opus vs. Claude-4-Sonnet).

3. Large generational gaps are intrinsically hard. Taken together, these results suggest that the stylistic and statistical differences between early and late generations within a family are large enough that a detector trained on one end of the spectrum cannot reliably cover the other at a fixed low FPR. Training on more recent models does help (backward transfer is generally stronger than forward transfer from the oldest model), but even the newest detectors do not fully close the gap on the oldest outputs. This highlights a fundamental challenge: as LLMs evolve, within-family distribution shift alone can already make supervised detection across generations non-trivial.

5.4 Universal Detectors (RQ3)

We next evaluate *universal detectors* trained on pooled data from multiple LLMs, comparing their performance to specialized single-model detectors. We consider three configurations: (1) trained on all OpenAI models, (2) trained on all Anthropic models, and (3) trained on both families combined. All results use our two-stage DeBERTa-v3-based training pipeline with tail-focused calibration.

5.4.1 Performance at Fixed Low FPR (0.1%)

Tables 5.5–5.7 report TPR at a stringent 0.1% FPR threshold, which is more representative of practical deployment scenarios (e.g., academic integrity screening where false accusations are costly).

Analysis. At the stringent 0.1% FPR threshold, mean TPR ranges from approximately 30% to 70% depending on the target model and training configuration. Several patterns are notable:

Table 5.5: Universal OpenAI detector performance at a fixed human FPR of 0.1%. Entries are mean \pm std across 3 seeds; range shows best–worst seed TPR.

	GPT-3.5	GPT-4o	GPT-4.1	o4-mini
TPR (%)@0.1% FPR	33.8 \pm 30.0	72.9 \pm 20.0	42.7 \pm 23.0	51.3 \pm 16.0
TPR range across seeds (%)	[11.4, 68.3]	[57.9, 95.4]	[26.6, 69.0]	[35.6, 67.1]

Calibration satisfied in all runs. Achieved human FPR worst-case across pools: 0.1%.

Seed values used for ranges: GPT-3.5 = {11.4, 21.6, 68.3}%, GPT-4o = {57.9, 65.5, 95.4}%, GPT-4.1 = {26.6, 32.4, 69.0}%, o4-mini = {35.6, 51.1, 67.1}%.

Table 5.6: Universal Anthropic detector performance at a fixed human FPR of 0.1%. Entries are mean \pm std across 3 seeds; range shows best–worst seed TPR.

	Claude 3 Opus	Claude 3.5 Sonnet	Claude 3.7 Sonnet	Claude 4 Sonnet
TPR (%)@0.1% FPR	47.9 \pm 8.7	58.6 \pm 28.1	50.1 \pm 26.6	52.8 \pm 25.9
TPR range across seeds (%)	[41.9, 57.9]	[37.4, 90.5]	[21.6, 74.2]	[25.3, 76.8]

Calibration satisfied in all runs. Achieved human FPR worst-case across pools: 0.1%.

Seed values used for ranges (TPR, %): Claude 3 Opus = {41.9, 43.9, 57.9}, Claude 3.5 Sonnet = {37.4, 48.0, 90.5}, Claude 3.7 Sonnet = {21.6, 54.6, 74.2}, Claude 4 Sonnet = {25.3, 56.5, 76.8}.

Single-family detectors outperform all-family detectors. Comparing Tables 5.5 and 5.6 to Table 5.7, we observe that pooling data from both families *reduces* mean TPR across most models. For example, the OpenAI-only detector achieves 72.9% mean TPR on GPT-4o, while the all-families detector achieves only 38.0%. This suggests that training on more diverse data introduces conflicting signals that make low-FPR calibration more difficult.

GPT-4o is the easiest target. Across all configurations, GPT-4o consistently yields the highest TPR (57.9–95.4% in the best seed), suggesting its outputs retain more detectable statistical signatures than other models.

Anthropic detectors show lower variance on Claude-3-Opus. The TPR range for Claude-3-Opus ([41.9, 57.9]) is notably tighter than for other models, possibly because this older model’s outputs are more consistently distinguishable.

Table 5.7: Universal detector for both model families performance at a fixed human FPR of 0.1%. Entries are mean \pm std across 3 seeds; range shows best–worst seed TPR.

	OpenAI family				Anthropic family			
	GPT-3.5	GPT-4o	GPT-4.1	o4-mini	Claude 3 Opus	Claude 3.5 Sonnet	Claude 3.7 Sonnet	Claude 4 Sonnet
TPR (%) @0.1% FPR	36.2 \pm 45.2	38.0 \pm 45.2	33.1 \pm 39.3	29.8 \pm 32.7	36.0 \pm 38.6	34.9 \pm 41.1	35.5 \pm 29.6	36.8 \pm 36.2
TPR range across seeds (%)	[4.7, 88.0]	[7.7, 90.0]	[3.0, 77.6]	[3.9, 66.6]	[7.4, 79.9]	[7.5, 82.2]	[18.3, 69.7]	[15.0, 78.5]

Calibration satisfied in all runs (target 0.1%).

Seed values used for ranges (TPR, %): GPT-3.5 Turbo = {4.7, 15.9, 88.0}, GPT-4o = {7.7, 16.4, 90.0}, GPT-4.1 = {3.0, 18.6, 77.6}, o4-mini = {3.9, 19.0, 66.6}, Claude 3 Opus = {7.4, 20.8, 79.9}, Claude 3.5 Sonnet = {7.5, 15.1, 82.2}, Claude 3.7 Sonnet = {18.3, 18.5, 69.7}, Claude 4 Sonnet = {15.0, 16.8, 78.5}.

Table 5.8: OpenAI universal detector: FPR at TPR = 90% (worst-case across human pools). Entries are mean \pm std across 3 seeds; range shows best–worst seed FPR.

	GPT-3.5	GPT-4o	GPT-4.1	o4-mini
FPR (%) @90% TPR	1.16 \pm 0.76	0.23 \pm 0.17	0.73 \pm 0.56	0.43 \pm 0.27
FPR range across seeds (%)	[0.60, 2.03]	[0.05, 0.38]	[0.29, 1.36]	[0.27, 0.75]

TPR target fixed at 90%.

5.4.2 Performance at Fixed High TPR (90%)

To characterize the minimum achievable FPR, we also report the FPR required to achieve 90% TPR. This perspective directly addresses whether our pipeline can reduce false positives relative to the \sim 1% baseline common in prior work.

How we compute FPR at a fixed TPR. To characterize the operating point required to achieve 90% TPR, we perform an offline inverse search over the calibration target α (equivalently, the decision threshold) and report the smallest α for which the TPR reaches 90%.

Analysis. Our tail-focused training pipeline achieves substantial FPR reductions compared to the \sim 1% baseline typical of prior work:

FPR reduction at 90% TPR. For GPT-4o, the OpenAI universal detector achieves a mean FPR of 0.23% at 90% TPR (Table 5.8), a \sim 4 \times reduction relative to a 1% FPR operating point. In the best seed, FPR reaches 0.05% (a 20 \times reduction), highlighting substantial seed sensitivity.

Table 5.9: Anthropic universal detector: FPR at TPR = 90% (worst-case across human pools). Entries are mean \pm std across 3 seeds; range shows best–worst seed FPR.

	Claude 3 Opus	Claude 3.5 Sonnet	Claude 3.7 Sonnet	Claude 4 Sonnet
FPR (%) @90% TPR	0.53 \pm 0.23	0.41 \pm 0.30	0.43 \pm 0.28	0.36 \pm 0.20
FPR range across seeds (%)	[0.30, 0.76]	[0.07, 0.62]	[0.17, 0.73]	[0.14, 0.55]

TPR target fixed at 90%.

Table 5.10: Universal detector for both model families: FPR @ TPR = 90% (worst-case across human pools). Entries are mean \pm std across 3 seeds; range shows best–worst seed FPR.

	OpenAI family				Anthropic family			
	GPT-3.5	GPT-4o	GPT-4.1	o4-mini	Claude 3 Opus	Claude 3.5 Sonnet	Claude 3.7 Sonnet	Claude 4 Sonnet
FPR (%) @90% TPR	0.59 \pm 0.51	0.53 \pm 0.42	0.55 \pm 0.33	0.60 \pm 0.30	0.58 \pm 0.40	0.61 \pm 0.48	0.53 \pm 0.32	0.52 \pm 0.38
FPR range across seeds (%)	[0.10, 1.12]	[0.08, 0.92]	[0.19, 0.83]	[0.27, 0.87]	[0.19, 0.99]	[0.16, 1.12]	[0.20, 0.84]	[0.13, 0.89]

TPR target fixed at 90%.

Anthropic models are more consistent. The Anthropic universal detector (Table 5.9) achieves FPR in the range 0.36–0.53% across all models, with tighter variance than the OpenAI detector. This may reflect greater homogeneity in Anthropic model outputs.

All-families detector maintains low FPR. Despite training on more diverse data, the all-families detector (Table 5.10) achieves mean FPR of approximately 0.5–0.6% across all models, demonstrating that a single universal detector can operate at substantially lower FPR than the 1% baseline without sacrificing 90% TPR.

5.5 Seed Instability and Data Constraints

A consistent pattern across our low-FPR results is high variance across random seeds. For example, the all-families detector’s TPR at 0.1% FPR ranges from 3.0% to 90.0% depending on the seed (Table 5.7). We attribute this instability to several factors:

Limited training data. Our datasets contain approximately 4,000 training samples per domain per model for most OpenAI models, and 800 samples per domain per model for

Anthropic models and some OpenAI models (GPT-4.1: 667; o4-mini: 800). At extremely low FPR thresholds, the effective number of human samples in the tail is small, making threshold estimation sensitive to sampling noise.

Short document lengths. Human samples are truncated to 256 tokens, limiting the information available per document and potentially increasing overlap between human and LLM score distributions.

Stage-2 sensitivity. The tail-focused fine-tuning stage involves multiple interacting components (hard mining, specialized losses, partial unfreezing) whose hyperparameters were not exhaustively tuned. Small changes in initialization or batch composition can substantially affect the learned decision boundary.

These constraints suggest that performance at very low FPR could be improved with (1) larger training corpora, (2) longer documents, and (3) more comprehensive hyperparameter optimization—directions we leave for future work.

Chapter 6

Discussion

We now revisit the baseline OpenAI experiments from Section 5.2. There, we compared zero-shot and supervised detectors primarily in terms of TPR at a fixed FPR. Here we report the *same* experiments using two aggregate metrics—AUROC and overall accuracy—to highlight how these can be misleading in low-FPR regimes.

Table 6.1: Performance of the fine-tuned **RoBERTa-base** detector on OpenAI models, reported using AUROC (%). This table uses the same setup and data as Table 5.1, but replaces TPR with AUROC.

	GPT-3.5	GPT-4o	GPT-4.1	o4-mini
Academic (arxiv)	99%	100%	100%	100%
News (XSum)	100%	100%	99%	100%
Creative Writing	99%	99%	99%	99%

Table 6.2: Performance of the fine-tuned **RoBERTa-base** detector on OpenAI models, reported using accuracy (%). Same detectors and data as Table 6.1.

	GPT-3.5	GPT-4o	GPT-4.1	o4-mini
Academic (arxiv)	99%	100%	99%	100%
News (XSum)	100%	100%	99%	99%
Creative Writing	99%	99%	97%	99%

Tables 6.1–6.4 are the AUROC and accuracy counterparts of Table 5.1: they report the very same OpenAI experiments, detectors, and datasets, but replace TPR-at-fixed-FPR

Table 6.3: Performance of the zero-shot **Binoculars** detector on OpenAI models, reported using AUROC (%). This table corresponds to the Binoculars rows in Table 5.1, but replaces TPR with AUROC.

	GPT-3.5	GPT-4o	GPT-4.1	o4-mini
Academic (arxiv)	99%	88%	93%	80%
News (XSum)	99%	97%	85%	95%
Creative Writing	99%	98%	91%	97%

Table 6.4: Performance of the zero-shot **Binoculars** detector on OpenAI models, reported using accuracy (%). Same detectors and data as Table 6.3.

	GPT-3.5	GPT-4o	GPT-4.1	o4-mini
Academic (arxiv)	99%	81%	86%	73%
News (XSum)	98%	91%	77%	89%
Creative Writing	99%	92%	83%	92%

with aggregate AUROC and overall accuracy. When viewed through these metrics, both the supervised RoBERTa baseline and the zero-shot Binoculars detector appear to perform extremely well: AUROC values are typically in the 90–100% range and accuracies cluster around 95–100%. However, Table 5.1 shows that at a realistic low-FPR operating point (approximately 1% false positives on human text), a very different picture emerges. The RoBERTa-based detector maintains high TPR on all models (often $\geq 95\%$), whereas Binoculars collapses on newer models and formal domains: for GPT-4o and GPT-4.1 in academic and news text, TPR falls to well below 1% despite AUROC values of 88–97% and accuracies above 80–90%. In other words, a detector can rank machine vs. human text reasonably well in aggregate (high AUROC) and achieve high average classification accuracy, yet be essentially useless at the low-FPR operating points that matter in settings like academic integrity screening. AUROC reflects average ranking quality across all thresholds; low-FPR operation depends only on the extreme right tail of the human score distribution, where even a small overlap can collapse TPR. These results emphasize that AUROC and accuracy are fundamentally insufficient for evaluating LLM-text detectors under deployment-relevant constraints; performance must instead be reported at fixed, application-appropriate FPR levels (e.g., 0.1–1%) to reveal whether a detector is viable in practice.

Chapter 7

Limitations

While our study aims to provide a systematic evaluation of LLM-text detectors under model and domain shift, several limitations constrain the scope and strength of our conclusions.

Model and family coverage. We focus on two proprietary model families (OpenAI and Anthropic) and three text domains (academic, news, creative writing), all in English. Our findings may not directly generalize to other families (e.g., Google’s Gemini, Alibaba’s Qwen) or to multilingual settings.

Short document lengths. All supervised experiments operate on relatively short passages: after preprocessing and truncation, texts are limited to a maximum of 256 tokens. This reflects the nature of our human data, which consists of short abstracts, summaries, and stories. To ensure a fair comparison, LLM-generated completions were matched in length to their human counterparts, so both the human and machine sides of the training and test sets are short-form.

Limited training data. Although we use thousands of training samples per model and domain, dataset sizes remain modest by modern standards, especially for some recent LLMs. At very low FPR thresholds, only a small fraction of human examples lie in the relevant tail, making threshold estimation and Stage-2 fine-tuning sensitive to sampling noise.

Seed instability and hyperparameter sensitivity. Our low-FPR DeBERTa-v3 pipeline exhibits substantial variability across random seeds, particularly at the 0.1% FPR target. We did not perform an exhaustive search over Stage-2 hyperparameters so some of this instability may be attributable to suboptimal settings.

Simplified threat model. Our main experiments consider text directly generated by LLMs in response to prompts, without systematic adversarial paraphrasing or intensive

human editing. Prior work shows that such transformations can sharply degrade detector performance. As a result, our results should be interpreted as upper bounds on robustness in non-adversarial settings.

Detector baselines. On the supervised side we evaluate a RoBERTa/DeBERTa-based detector, and on the zero-shot side we focus on Binoculars, which recent benchmarks consistently identify as SOTA. We do not re-evaluate older zero-shot methods (e.g., DetectGPT, DNA-GPT) nor commercial detectors such as GPTZero. Although existing benchmarks suggest these are weaker than our chosen baselines, a more comprehensive head-to-head comparison is left for future work.

Chapter 8

Future Work

Our results suggest several concrete directions for extending and strengthening LLM-text detection.

Richer detector baselines. An immediate extension is to broaden the set of detectors evaluated. On the zero-shot side, it would be valuable to compare Binoculars more systematically to other perplexity- and embedding-based methods, and to explore variants that use fine-tuned observer/performer models rather than base LMs. On the supervised side, incorporating commercial detectors such as GPTZero into the benchmark would provide a more complete picture of the current landscape.

Leveraging model logits where available. Our study deliberately assumes only text-level access to proprietary LLMs. In settings where APIs expose token-level probabilities or logits, future work could investigate detectors that combine our tail-focused pipeline with direct information from the target model, potentially improving both zero-shot (Binoculars-style) and supervised approaches.

Adversarial and edited text. A natural next step is to evaluate detectors under stronger threat models that include paraphrasing, style transfer, and human editing. Integrating such transformations into both training (e.g., through adversarial augmentation) and calibration would stress-test robustness beyond the vanilla generations considered here.

Longer documents and richer calibration. Extending the low-FPR pipeline to longer documents is an important direction. Longer contexts may provide additional signal for detection.

Scaling data and stabilizing Stage-2. Increasing the size and diversity of both human calibration pools and machine-generated training data may reduce variance in the

tails and improve stability at 0.1% FPR. Complementary to this, a more systematic study of Stage-2 objectives and hyperparameters (including ablations and automated tuning) could identify configurations that are less seed-sensitive while retaining strong low-FPR performance.

Progressive adaptation to new models. In realistic deployment, detectors will need to adapt as new LLM releases appear. One promising line of work is progressive or data-efficient fine-tuning of a universal detector on small labeled samples from a new target model, measuring how quickly TPR can be recovered at a fixed FPR budget. This would more closely mimic the life-cycle of a deployed detector that is periodically updated.

Broader model and language coverage. Finally, extending our evaluation to additional families (such as Gemini, Qwen, and other emerging models), as well as to non-English and multilingual settings, would help clarify how general our conclusions are and where family- or language-specific effects dominate.

Chapter 9

Conclusion

This thesis set out to evaluate the generalizability of state-of-the-art LLM-text detectors across evolving model families and domains, guided by three research questions (RQ1–RQ3). We compared a supervised RoBERTa baseline and the zero-shot Binoculars detector on text from two major LLM families (OpenAI and Anthropic) and three domains (academic, news, creative writing), analyzed within-family transfer across multiple generations of models, and introduced a DeBERTa-v3-based supervised pipeline with tail-focused training and document-level calibration for operation at FPRs as low as 0.1%.

Our experiments lead to three main conclusions. First, zero-shot methods such as Binoculars, while attractive for their lack of supervision, are not reliable for frontier models: they can achieve high AUROC and accuracy, yet collapse to near-zero TPR at realistic low-FPR operating points, particularly on formal text. Second, supervised detectors remain practically viable, but only in a model-aware regime: detectors trained on one LLM generation transfer poorly to distant generations within the same family, and universal detectors operating at 0.1% FPR require careful calibration and still exhibit substantial variability across seeds. Robust deployment is therefore unlikely to look like a single, static “AI detector” and more like a continually recalibrated, family-aware detection pipeline. Third, our comparison of AUROC and accuracy against TPR-at-fixed-FPR shows that commonly reported aggregate metrics can be misleading in exactly the low-FPR regimes that matter for high-stakes applications such as academic integrity or content moderation. Evaluation practices for LLM detection should therefore standardize on reporting TPR and realized FPR at application-appropriate thresholds, along with variability across seeds and models, rather than relying solely on headline AUROC or accuracy figures.

The real-world incidents documented in our Introduction reinforce the urgency of re-

liable detection. Our results suggest that addressing these challenges will require not a single 'AI detector' but a continually updated, model-aware detection infrastructure with appropriate human oversight.

References

- [1] OpenAI. Introducing chatgpt, November 2022. Published November 30, 2022.
- [2] Jennifer Sor. Sam altman touts chatgpt’s 800 million weekly users, double all its main competitors combined, October 2025. Published October 8, 2025.
- [3] Rebecca Bellan. Sam altman says chatgpt has hit 800m weekly active users, October 2025. Published October 6, 2025.
- [4] Nestor Maslej, Loredana Fattorini, Yolanda Gil, Vanessa Parli, Raymond Perrault, et al. Artificial Intelligence Index Report 2025. Technical report, Stanford Institute for Human-Centered AI (HAI), Stanford, CA, USA, April 2025. Co-Directors: Yolanda Gil and Raymond Perrault. Accessed: 2025-12-01.
- [5] Josh Freeman. Student generative AI survey 2025. Technical Report Policy Note 61, Higher Education Policy Institute (HEPI), February 2025. Published 26 February 2025.
- [6] Alexandra Fiedler and Jörg Döpke. Do humans identify AI-generated text better than machines? evidence based on excerpts from German theses. *International Review of Economics Education*, 49:100321, June 2025.
- [7] Jan Hendrik Kirchner, Lama Ahmad, Scott Aaronson, and Jan Leike. New AI classifier for indicating AI-written text, January 2023. Published January 31, 2023; OpenAI noted on July 20, 2023 that the classifier was withdrawn due to low accuracy.
- [8] Souradip Chakraborty, Amrit Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. Position: On the possibilities of AI-generated text detection. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6093–6115. PMLR, July 2024.

- [9] Kyle Bittle and Omar El-Gayar. Generative AI and academic integrity in higher education: A systematic review and research agenda. *Information*, 16(4):296, April 2025. Published 8 April 2025.
- [10] Jan Henrik Gruenhagen, Peter M. Sinclair, Julie-Anne Carroll, Philip R. A. Baker, Ann Wilson, and Daniel Demant. The rapid rise of generative AI and its implications for academic integrity: Students’ perceptions and use of chatbots for assistance with assessments. *Computers and Education: Artificial Intelligence*, 7:100273, December 2024.
- [11] Nguyen Van Hanh and Nguyen Thi Duyen. AI-assisted academic cheating: a conceptual model based on postgraduate student voices. *Frontiers in Computer Science*, 7:1682190, November 2025. Published 26 November 2025.
- [12] Giovanni Spitale, Nikola Biller-Andorno, and Federico Germani. AI model GPT-3 (dis)informs us better than humans. *Science Advances*, 9(26):eadh1850, June 2023.
- [13] Morgan Wack, Carl Ehrett, Darren Linvill, and Patrick Warren. Generative propaganda: Evidence of AI’s impact from a state-backed disinformation campaign. *PNAS Nexus*, 4(4):pgaf083, April 2025.
- [14] Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Maria Bielikova. Disinformation capabilities of large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14830–14847, Bangkok, Thailand, August 2024. Association for Computational Linguistics.
- [15] António Bandeira, Luis Henrique Gonçalves, Felix Holl, Juliet Ugbedeajo Shaibu, Mariana Laranjo Gonçalves, Ronan Payinda, Sagun Paudel, Alessandro Berionni, Young WFPHA, Tina D Purnat, and Tim Mackey. Viewpoint on the intersection among health information, misinformation, and generative AI technologies. *JMIR Infodemiology*, 5(1):e69474, September 2025. Published 15 Sep 2025.
- [16] David Byrd. Exploring sentiment manipulation by LLM-enabled intelligent trading agents, 2025. v1, submitted 22 Feb 2025.
- [17] Diomidis Spinellis. False authorship: an explorative case study around an AI-generated article published under my name. *Research Integrity and Peer Review*, 10:8, May 2025. Published 27 May 2025.

- [18] Reese A. K. Richardson, Spencer S. Hong, Jennifer A. Byrne, Thomas Stoeger, and Luís A. Nunes Amaral. The entities enabling scientific fraud at scale are large, resilient, and growing rapidly. *Proceedings of the National Academy of Sciences of the United States of America*, 122(32):e2420092122, August 2025. Published 12 Aug 2025; Epub 4 Aug 2025.
- [19] Weiyao Meng, John Harvey, James Goulding, Chris James Carter, Evgeniya Lukinova, Andrew Smith, Paul Frobisher, Mina Forrest, and Georgiana Nica-Avram. Large language models as 'hidden persuaders': Fake product reviews are indistinguishable to humans and machines, June 2025. v1, submitted 16 Jun 2025.
- [20] Malte Josten and Torben Weis. Investigating the effectiveness of Bayesian spam filters in detecting LLM-modified spam mails. In Sanjay Goel, Ersin Uzun, Mengjun Xie, and Sumantra Sarkar, editors, *Digital Forensics and Cyber Crime. ICDF2C 2024*, volume 613 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 285–295, Cham, May 2025. Springer.
- [21] Valdemar Danry, Pat Pataranutaporn, Matthew Groh, Ziv Epstein, and Pattie Maes. Deceptive AI systems that give explanations are more convincing than honest AI systems and can amplify belief in misinformation, July 2024. v1, submitted 31 Jul 2024.
- [22] James Hutson. Rethinking plagiarism in the era of generative AI. *Journal of Intelligent Communication*, 3(2):20–31, September 2024.
- [23] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget, May 2023. v3, last revised 14 Apr 2024.
- [24] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631:755–759, July 2024. Published 24 Jul 2024; Author Correction 21 Mar 2025.
- [25] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery, August 2024. arXiv:2408.06292 [cs.AI], v3 (1 Sep 2024).
- [26] Sara Merken. Judge disqualifies three butler snow attorneys from case over AI citations, July 2025. Published July 24, 2025.

- [27] Gerrit De Vynck. Openai finds russian and chinese groups used its tech for propaganda campaigns, May 2024. Updated May 30, 2024.
- [28] Katie Paul. Meta identifies networks pushing deceptive content likely generated by AI, May 2024. Published May 29, 2024; Reporting by Katie Paul; Editing by Rod Nickel.
- [29] Andrey Sychev. Russia-linked AI websites aim to dupe german voters, study finds, January 2025. Published January 23, 2025.
- [30] Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Anirudha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting llms with binoculars: zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning, ICML'24*. JMLR.org, 2024.
- [31] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized BERT pre-training approach with post-training. In Sheng Li, Maosong Sun, Yang Liu, Hua Wu, Kang Liu, Wanxiang Che, Shizhu He, and Gaoqi Rao, editors, *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China, August 2021. Chinese Information Processing Society of China.
- [32] Mervat Abassy, Kareem Elozeiri, Alexander Aziz, Minh Ngoc Ta, Raj Vardhan Tomar, Bimarsha Adhikari, Saad El Dine Ahmed, Yuxia Wang, Osama Mohammed Afzal, Zhuohan Xie, Jonibek Mansurov, Ekaterina Artemova, Vladislav Mikhailov, Rui Xing, Jiahui Geng, Hasan Iqbal, Zain Muhammad Mujahid, Tarek Mahmoud, Akim Tsvigun, Alham Fikri Aji, Artem Shelmanov, Nizar Habash, Iryna Gurevych, and Preslav Nakov. LLM-DetectAIve: a tool for fine-grained machine-generated text detection. In Delia Irazu Hernandez Farias, Tom Hope, and Manling Li, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 336–343, Miami, Florida, USA, November 2024. Association for Computational Linguistics.
- [33] 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, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1702–1717, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

- [34] Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. GPT-who: An information density-based machine-generated text detector. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 103–115, Mexico City, Mexico, June 2024. Association for Computational Linguistics.
- [35] Xiaomengc Hu, Pin-Yu Chen, and Tsung-Yi Ho. Radar: robust ai-text detection via adversarial learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA, 2023. Curran Associates Inc.
- [36] Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, Vancouver, Canada, February 2024.
- [37] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning, ICML'23*. JMLR.org, 2023.
- [38] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [39] 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*, 2023.
- [40] Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. DNA-GPT: divergent n-gram analysis for training-free detection of gpt-generated text. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024.
- [41] Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. Intrinsic dimension estimation for robust detection of ai-generated texts. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA, 2023. Curran Associates Inc.

- [42] Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F. Wong, and Lidia S. Chao. A survey on llm-generated text detection: Necessity, methods, and future directions. *CoRR*, abs/2310.14724, 2023.
- [43] Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. MGTBench: Benchmarking Machine-Generated Text Detection. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2024.
- [44] 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)*, pages 12463–12492, Bangkok, Thailand, August 2024. Association for Computational Linguistics.
- [45] 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 *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, Vancouver, Canada, 2024. Neural Information Processing Systems Foundation.
- [46] 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.
- [47] OpenAI. Introducing ChatGPT and Whisper APIs. <https://openai.com/blog/introducing-chatgpt-and-whisper-apis>, March 2023. Describes the introduction of the `gpt-3.5-turbo` model used in ChatGPT and the OpenAI API.
- [48] OpenAI. Hello GPT-4o. <https://openai.com/index/hello-gpt-4o>, May 2024. Product announcement and overview of the multimodal GPT-4o model.
- [49] OpenAI. Introducing GPT-4.1 in the API. <https://openai.com/index/gpt-4-1>, April 2025. Technical/product description of the GPT-4.1 model family and its capabilities.

- [50] OpenAI. Introducing OpenAI o3 and o4-mini. <https://openai.com/index/introducing-o3-and-o4-mini>, April 2025. Blog post introducing the o-series reasoning models, including o4-mini.
- [51] Anthropic. Introducing the next generation of Claude: The Claude 3 model family. <https://www.anthropic.com/news/claude-3-family>, March 2024. Announcement of the Claude 3 family (Haiku, Sonnet, Opus), including Claude-3-Opus.
- [52] Anthropic. Claude 3.5 sonnet model card addendum. Technical report, Anthropic PBC, 2024. <https://www.anthropic.com/news/claude-3-5-sonnet>.
- [53] Anthropic. Claude 3.7 sonnet: Hybrid reasoning model. <https://www.anthropic.com/news/claude-3-7-sonnet>, February 2025. Anthropic announcement describing the claude-3-7-sonnet model.
- [54] Anthropic. Claude sonnet 4. <https://www.anthropic.com/news/claude-4>, May 2025. Product page and announcement for the Claude 4 model family.
- [55] Euclaise. WritingPrompts_curated. https://huggingface.co/datasets/euclaise/WritingPrompts_curated, 2023.
- [56] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, 2018.
- [57] arXiv.org submitters. arxiv dataset, 2024.
- [58] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don’t give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745, 2018.
- [59] Xin Tong, Yang Feng, and Jialiang Li. Neyman–Pearson classification algorithms and NP receiver operating characteristics. *Science Advances*, 4(2):eaao1659, 2018.
- [60] Debora Weber-Wulff, Alla Anohina-Naumeca, Sonja Bjelobaba, Tomáš Foltýnek, and Jean Guerrero-Dib. Testing of detection tools for AI-generated text. *International Journal for Educational Integrity*, 19(1):26, 2023.
- [61] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 41–48, 2009.

- [62] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021.
- [63] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021.
- [64] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 761–769, 2016.
- [65] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.
- [66] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pages 2127–2136, 2018.
- [67] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 89–96, 2005.
- [68] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 328–339, 2018.
- [69] Jennifer T. Kost and Michael P. McDermott. Combining dependent p-values. *Statistics & Probability Letters*, 60(2):183–190, 2002.
- [70] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

APPENDICES

Appendix A

Additional info

Data generation. Concretely, we reuse the same prompt that originally elicited the human text (e.g., the article title for academic writing, the original WritingPrompts prompt for creative writing, or the reference summary for news) as the user message, and prepend a task-specific system instruction that explicitly describes the generation task (for example: “You will be given a writing prompt, and you will provide a detailed creative story of at least 300 words based on the prompt. Generate the text directly without preceding it with anything.”). We use default model settings and a temperature of $T = 1.0$.

Preprocessing. All texts (human and LLM-generated) undergo a unified normalization and cleaning step. We first apply Unicode normalization using Python’s `ftfy` package to correct encoding artifacts and non-standard characters. We then remove common boilerplate prefixes such as “Title:”, “Abstract:” and “Article:” (case-insensitively), as well as wrapper quote blocks (e.g., “...””) that some APIs prepend or append to model outputs. Next, we collapse all sequences of whitespace into a single space and strip leading and trailing whitespace, discarding empty or extremely short documents. Finally, we tokenize with the detector tokenizer and restrict sequence length to at most 256 tokens, truncating longer documents. Crucially, we adopt this relatively aggressive normalization to remove non-linguistic “shortcut” features (e.g., systematic prefixes, quoting conventions, or unusual Unicode artifacts) that could allow the supervised DeBERTa-based detector to separate human and LLM-generated texts by exploiting formatting cues rather than genuine linguistic differences.

Binoculars evaluation. Given a CSV with columns `text` and `label` (“human” or “machine”), we load the data with `datasets`, map labels to $\{0, 1\}$ (`human`→ 0, `machine`→ 1), and perform a stratified dev/test split. On the dev split, we run a single Binoculars instance in batch mode to obtain scores $B_{M_1, M_2}(s_i)$. For each experimental condition we report test-set accuracy, precision, recall, F1 for the machine class, AUROC, and TPR at a fixed low FPR.

Binoculars extending to modern LLMs **Tokenization and sequence-length alignment.** We always tokenize with the M_1 tokenizer, truncate to a maximum of L tokens, and pad as needed. The same tokenized batch is then fed to both M_1 and M_2 . For large sharded models loaded with `device_map="auto"`, we observed that the returned logits can have a sequence length slightly larger than the input (due to internal padding). Before computing log-perplexity and cross-perplexity, we truncate all logits and tokenization tensors (`input_ids`, `attention_mask`) to the minimum common sequence length across models and inputs, ensuring that both log PPL and log X-PPL are evaluated on aligned token positions.

Threshold interface. The reference implementation exposes two pre-computed operating points, “accuracy” and “low-fpr”. We wrap these, and any user-specified numeric threshold, in a unified `set_threshold` interface. This allows us to either use the original global thresholds of Hans et al. or plug in dataset-specific thresholds calibrated as described next.

Threshold calibration and evaluation. For each dataset and model pair (M_1, M_2) , we calibrate the decision threshold on a held-out development split and then evaluate on a disjoint test split. We perform a stratified split into development and test sets, calibrate the threshold on the development set, and evaluate on the test set.