
Deep Learning Approaches for AI-generated
Content Detection:
From Benchmark Models to Masked Convolutional
Self-Attention (MCSA)

*A thesis submitted in fulfilment of the requirements
for the degree of*

Master of Science (Research)

in

Computing Sciences

by

Zhiben Song

to

School of Computer Science

Faculty of Engineering and Information Technology

University of Technology Sydney

NSW - 2007, Australia

October 2025

© 2025 by Zhiben Song

All Rights Reserved

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Zhiben Song*, declare that this thesis is submitted in fulfilment of the requirements for the award of Master of Science (Research) in Computing Sciences, in the *School of Computer Science, Faculty of Engineering and Information Technology* at the University of Technology Sydney. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research was supported by an Australian Government Research Training Program (RTP) Scholarship doi.org/10.82133/C42F-K220.

SIGNATURE: Production Note:
 Signature removed prior to publication.

[Zhiben Song]

DATE: 25th October, 2025

PLACE: Sydney, Australia

ACKNOWLEDGMENTS

First and foremost, I would like to express my heartfelt gratitude to my principal supervisor, Dr. Xianzhi Wang. Throughout my academic journey, Dr. Wang has provided not only insightful academic guidance but also unwavering emotional support and encouragement. He has always been patient in listening to my ideas—no matter how immature—and has generously shared his time and expertise to help me refine and elevate my thinking. His rigor in research, kindness in mentorship, and trust in his students have left a profound impact on me.

During moments of self-doubt and low motivation, it was Dr. Wang's words of encouragement that helped calm my anxieties and restore my confidence. When I faced challenges in life, academic pressure, and even legal complications, he remained understanding and supportive. I am well aware that I am not the most outstanding student, yet Dr. Wang never lowered his expectations of me, nor did he ever give up on me. His mentorship has been a guiding light, both in research and in life. I will always be deeply grateful for everything he has done for me.

I would also like to extend my deepest thanks to my family. Your constant support, understanding, and love have been my source of strength throughout this journey. Thank you for standing by me during times of stress, uncertainty, and near breakdowns. Your unwavering belief in me gave me the courage to persevere. I am especially grateful to my mother, whose care and encouragement helped me stay grounded and hopeful.

Over the past year, I went through a very difficult period. When I encountered an unexpected legal crisis, it was the university's legal advisory service that helped me find and connect with my current lawyer. Their timely assistance provided much-needed clarity and direction during a chaotic and overwhelming time.

Emotionally, I was also at a low point—experiencing symptoms that bordered on depression. Thankfully, I received support from the university's counseling service. The professional, warm, and empathetic conversations I had with the counselors helped me regain emotional balance and inner strength. That experience taught me the importance of seeking help and reminded me that I wasn't alone.

I would also like to thank the faculty members, administrative staff, and fellow researchers at the School of Computer Science, Faculty of Engineering and Information Technology, University of Technology Sydney. The collaborative, inclusive, and intellectually stimulating environment you helped foster allowed me to stay focused and continue to grow both academically and personally.

Lastly, I wish to express my sincere appreciation to everyone who has offered me support, guidance, or encouragement throughout this journey. Whether it was a helpful reply, a warm greeting, a thought-provoking discussion, or simply a quiet presence during tough times—you have all contributed in meaningful ways.

This thesis is not only the product of academic exploration, but also a testament to the compassion, encouragement, and belief I have received from so many. I will forever

carry the memory of this journey and the light it brought me through the darkest times.

LIST OF PUBLICATIONS

LIST OF PUBLICATIONS:

1. **Zhiben, S** (2025). Detecting AI-generated content: A survey (submitted)

ABSTRACT

With large language models becoming widely deployed, the ability to differentiate AI-generated text from human-authored writing has grown increasingly important in domains such as academic integrity, cybersecurity, digital forensics, and content moderation. However, existing detection approaches often suffer from limitations such as shortcut learning and performance degradation under distribution shift, which reduce their reliability in real-world scenarios. This thesis investigates and compares various deep learning-based architectures for the detection of AI-generated content, aiming to evaluate their effectiveness, robustness, and limitations in classifying textual data based on its origin.

The research begins with a comprehensive review of prior studies in AI-generated content (AIGC) detection, identifying key challenges, methodological trends, and evaluation metrics. Building upon these insights, this work implements and systematically evaluates a set of neural network models, including Recurrent Neural Networks (RNN), Long Short-Term Memory networks (LSTM), LSTM with Attention, Self-Attention mechanisms, Convolutional Neural Networks (CNN), and Recurrent Convolutional Neural Networks (RCNN). *Motivated by empirical findings from this comparative evaluation, we further introduce a lightweight hybrid architecture, Masked Convolutional Self-Attention (MCSA), designed as a diagnostic hybrid that combines multi-scale convolutional feature extraction with a self-attention encoder via position-wise gated fusion.* Each model is trained and evaluated using benchmark datasets that include both human-authored and machine-generated text, and performance is measured using multiple classification metrics such as accuracy, precision, recall, and F1-score, *under a unified evaluation protocol that includes mild robustness probes and distribution shift settings.*

The evaluation reveals how different architectures handle the nuanced linguistic features and statistical patterns of AI-generated text. Attention-based models, particularly those incorporating self-attention or hybrid convolutional-recurrent structures, demonstrate improved capability in capturing subtle contextual cues, outperforming baseline models on various metrics. *Within this landscape, MCSA achieves competitive standard accuracy while improving robustness under word-order perturbations and train-test mismatch, benefiting from local n -gram and subword cues (via convolutions) and global dependencies (via self-attention).* Nevertheless, the analysis also identifies limitations, including reduced performance on shorter texts and sensitivity to adversarial paraphrasing, highlighting areas for future improvement.

By offering a comparative analysis of leading models used for AIGC detection *and introducing MCSA as a balanced diagnostic hybrid motivated by empirical findings*, this thesis contributes valuable empirical evidence and methodological insights for advancing detection robustness and reliability. The findings not only inform future development of AI content detectors but also support broader efforts in safeguarding digital spaces against manipulation, deception, and academic dishonesty in the era of generative AI.

TABLE OF CONTENTS

List of Publications	iv
List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Background	1
1.2 Motivation	2
1.3 Research Questions, Objectives, and Claims	3
1.3.1 Research Questions	3
1.3.2 Research Objectives	4
1.3.3 Claims	4
1.4 Existing Efforts and Challenges	5
1.5 Significance	6
1.6 Proposed Solutions, Contributions and Impact	6
1.6.1 Proposed Solutions	6
1.6.2 Contributions	7
1.6.3 Practical Impact	8
1.7 Thesis Outline	8
2 Literature Review	10
2.1 Introduction	10
2.2 Rule-based and Symbolic Systems	10
2.3 Detection Techniques	13
2.3.1 Overview and Taxonomy	13
2.3.2 Traditional Stylometric and Statistical Methods	14
2.3.3 Supervised Deep Learning Detection Models	15
2.3.4 Fine-tuning Based Detectors	16
2.3.5 Zero-shot and Black-box Detection Approaches	16
2.3.6 Watermarking and Provenance-based Detection	18
2.3.7 Summary of Detection Paradigms	19

2.4	Key Challenges in AIGC Detection	20
2.5	Summary and Research Gap	21
3	Evaluation Framework and Comparisons	23
3.1	Introduction	23
3.2	Dataset Description	24
3.3	Methodology	25
3.3.1	Convolutional Neural Network (CNN)	25
3.3.2	Recurrent Neural Network (RNN)	26
3.3.3	Long Short-Term Memory (LSTM)	27
3.3.4	Recurrent Convolutional Neural Network (RCNN)	27
3.3.5	Attention-based LSTM	28
3.3.6	Self-Attention-based Model	29
3.4	Experimental Setup and Sanity Checks	30
3.4.1	Data Preprocessing	30
3.4.2	Training Protocol and Metrics	32
3.4.3	Standard Test Set Results	32
3.4.4	Sanity Checks: Motivation and Design	33
3.4.5	Sanity Check Results and Analysis	35
3.5	Cross-Model Comparison under Strong Sanitization	36
3.6	Summary	38
4	A Masked Convolutional Self-Attention (MCSA) Approach	40
4.1	Motivation	40
4.2	Related Work	41
4.3	Method: MCSA Architecture and Formulation	43
4.4	Experimental Setup	44
4.5	Results and Analysis	46
4.6	Discussion and Limitations	48
4.7	Summary	49
5	Conclusion & Future Work	50
5.1	Conclusion	50
5.2	Future Work	51
	Bibliography	53

LIST OF FIGURES

FIGURE	Page
3.1 TextCNN architecture.	26
3.2 RNN architecture (2-layer bidirectional RNN).	27
3.3 Bidirectional-LSTM: X-Input, Y-Output, A-Forward & Backward units	28
3.4 RCNN architecture.	28
3.5 Attention-based LSTM [1]	29
3.6 Self-attention architecture [2]	31
4.1 MCSA architecture	44
4.2 Protocol-A: Std Test vs. Shuffled Val (Macro-F1)	46
4.3 Protocol-B: Relative Drop (Val)	46

LIST OF TABLES

TABLE	Page
3.1 Performance on the standard test set without sanitization.	32
3.2 Sanity check results (macro-F1). The TF-IDF+LR baseline is computed once on validation data.	35
3.3 CNN performance under different sanitization intensities.	36
3.4 Performance of all models under strong sanitization settings. The shuffled- order column shows macro-F1 on Sanity Check 2.	37
4.1 Protocol-A/B comparison (Macro-F1). Relative Drop is computed on Protocol-B validation: $F1_{\text{clean}} - F1_{\text{clean+shuffle}}$. Best per column in bold	46

INTRODUCTION

1.1 Background

In recent years, the rapid expansion of large-scale language models (LLMs) such as GPTs, Claude, and Grok has greatly enhanced artificial intelligence systems' ability to produce text that closely resembles human writing. This rapid evolution of generative AI, commonly referred to as AI-generated content (AIGC), has led to an influx of synthetic text across domains including academic publishing, journalism, social media, and customer service [3, 4]. While these technologies provide remarkable efficiency and scalability in content creation, they have also introduced critical concerns regarding authorship, authenticity, intellectual property, and misinformation [5, 6].

AIGC is often indistinguishable from human-written content, posing significant challenges in evaluating its origin and trustworthiness. This surge has generated a pressing need for reliable detection methods that can determine whether a given text is human-written or produced by an AI system [7, 8]. Applications of such detection systems span critical domains such as plagiarism prevention in academia, content moderation on digital platforms, cybersecurity, and protection of intellectual property rights [9, 10].

To address this emerging issue, researchers have begun developing computational models to detect AIGC using both linguistic patterns and statistical features. Among these approaches, deep learning models have become particularly prominent because of their ability to learn and represent intricate linguistic patterns[11, 12]. Conventional machine learning classifiers frequently exhibit limited generalization when faced with varied text genres and often fail to adapt to the broad diversity of contemporary generative models. Deep neural network architectures-such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNNs)-have

achieved strong results in natural language understanding tasks and are considered promising approaches for detecting AI-generated content [13].

Furthermore, enhancements such as attention mechanisms and hybrid models (e.g., LSTM + Attention, Self-Attention, RCNN) have shown potential in capturing subtle semantic and syntactic differences between human-written and machine-generated content [2, 14]. These architectural mechanisms enable models to selectively attend to the most relevant segments of a sequence and capture long-distance relationships more efficiently, a capability that can be essential for detecting subtle generative patterns present in AI-created text[15].

This research investigates the comparative performance of six popular deep learning architectures-RNN, LSTM, LSTM + Attention, Self-Attention, CNN, and RCNN-on the task of binary classification: distinguishing between human-authored and AI-generated text. By assessing these models on established benchmark datasets, this work seeks to offer empirical evidence regarding their strengths, limitations, and overall applicability to real-world AIGC detection scenarios.

1.2 Motivation

The rapid evolution of AI-based generative technologies has resulted in the extensive creation and dissemination of AI-generated content (AIGC) across numerous fields, including education, journalism, digital marketing, and social media. These models are capable of generating text that is remarkably fluent, logically structured, and contextually aligned with human writing, which makes it progressively harder for readers to reliably tell AI-generated content apart from human-authored text.[4, 16].

While AIGC brings efficiency and scalability, it also introduces significant societal and ethical risks. In academic contexts, students may misuse AI-generated text to complete assignments, threatening academic integrity [17, 18]. In cybersecurity, malicious actors can employ AIGC for phishing, disinformation, and impersonation attacks [9]. Furthermore, the unchecked proliferation of AI-authored content undermines trust in authorship, blurs the boundary of intellectual property rights, and challenges legal and educational institutions worldwide.

The motivation for this research is grounded in the growing need for reliable and generalizable detection mechanisms that can differentiate AI-generated content from human-authored text across various contexts. Traditional detection approaches-typically relying on stylometric cues or basic statistical features-have difficulty keeping up with the increasing sophistication of modern generative models[8]. As newer architectures such as Transformer-based generators and diffusion models emerge, detection frameworks must evolve accordingly, embracing scalable, explainable, and model-agnostic techniques.

Moreover, there is a pressing need for fairness and transparency in detection systems. Many existing detectors exhibit biases against specific writing styles, domains, or non-native English usage. Without proper fairness evaluation and interpretability, such systems may misclassify content, resulting in ethical and legal consequences. Therefore, this study aims not only to improve detection performance but also to ensure that such systems are trustworthy, interpretable, and equitable across diverse users and use cases.

By investigating a diverse set of deep learning models (RNN, LSTM, LSTM + Attention, Self-Attention, CNN, RCNN) on the task of classifying text as AI- or human-generated, this research seeks to contribute practical, interpretable, and robust detection solutions. These efforts are crucial to preserving authenticity and accountability in a world increasingly shaped by machine-generated language.

1.3 Research Questions, Objectives, and Claims

1.3.1 Research Questions

To better understand the effectiveness of deep learning models in detecting AI-generated content (AIGC), this study formulates the following research questions:

- **RQ1:** How do different deep learning architectures perform in distinguishing AI-generated from human-written text across varying content types and text lengths?
- **RQ2:** What are the comparative strengths and weaknesses of recurrent-based, convolutional-based, and attention-based models in the context of AIGC detection?
- **RQ3:** How sensitive are these models to specific challenges such as short text input, domain shifts, or style overlaps?

Beyond benchmarking. Motivated by Chapter 3, which suggests local–global complementarity between CNN and self-attention, we add one method-oriented question:

- **RQ4:** Does the proposed hybrid MCSA-combining multi-scale CNN and self-attention via position-wise gated fusion-maintain competitive standard accuracy while improving robustness to word-order perturbations and train-test distribution shifts (i.e., smaller relative drops), compared with CNN and pure self-attention?
- **Operational Definition:** In this study, an improved tradeoff refers to maintaining comparable standard accuracy (within approximately 1–2%) while demonstrating reduced performance degradation under robustness evaluation settings, such as word-order perturbation and distribution shift scenarios.

1.3.2 Research Objectives

To address these questions, this research pursues the following objectives:

- **RO1:** Select and implement a set of representative deep learning architectures—including RNN, LSTM, LSTM with Attention, Self-Attention, CNN, and RCNN—for the task of AIGC detection.
- **RO2:** Conduct systematic performance evaluation using benchmark datasets, focusing on metrics such as accuracy, precision, recall, and F1-score.
- **RO3:** Analyze how model architecture influences detection performance under various conditions, including input length, semantic complexity, and generative source.
- **RO4:** Build and evaluate the proposed MCSA under the Chapter 3 protocol, and quantify its accuracy–robustness trade-off versus CNN and self-attention.

1.3.3 Claims

This study is grounded on the following key assumptions and hypotheses, and-motivated by the comparative evidence in Chapter 3—includes a hypothesis on a lightweight hybrid model (MCSA):

1. Deep learning models exhibit varied effectiveness in AIGC detection, as shown in prior studies [19, 20], with attention-enhanced models likely outperforming simpler recurrent or convolutional baselines on *standard* accuracy, while convolutional models tend to be more invariant to local reordering.
2. No single architecture is universally optimal, and prior studies [21, 22] suggest that performance differences are highly context-dependent, particularly with respect to input characteristics and text length.
3. Comparative empirical analysis under a unified, sanitization-aware protocol has been shown to offer valuable insights that *inform* the practical design of future detection tools [23, 24].
4. Motivated by the comparative findings in Chapter 3, this thesis hypothesizes that a simple hybrid that fuses multi-scale convolutional features with self-attention via position-wise gated fusion (i.e., *MCSA*) can achieve a *stronger accuracy–robustness trade-off* than either family alone, especially under word-order perturbation and train–test mismatch.

1.4 Existing Efforts and Challenges

In recent years, research on detecting AI-generated content (AIGC) has expanded rapidly, motivated by the increasing availability and rising sophistication of large language models. Existing efforts can be broadly classified into two categories: feature-based methods and neural classifier-based methods. Feature-based approaches focus on stylometric, statistical, or linguistic patterns to detect generated text. Early works utilized lexical richness, n-gram frequencies, syntactic structures, and entropy-based metrics such as perplexity [25, 26]. However, as generative models became more fluent and context-aware, these methods struggled to differentiate content reliably. To address this, neural-based detectors emerged, particularly leveraging transformer architectures like BERT and RoBERTa [21, 27]. Models like DetectGPT and GPTZero have demonstrated improved detection by exploiting generation-specific probability distributions or perturbation-based scoring [22, 28]. Nonetheless, most models still exhibit performance degradation when applied to unseen model outputs or non-English content [23]. Another line of work focuses on watermarking-based detection, embedding imperceptible patterns in generated text that can be algorithmically traced [29]. While promising in theory, watermark-based methods face adoption challenges due to the requirement of model-owner cooperation and vulnerability to paraphrasing attacks.

Despite these advancements, several key challenges remain:

- **Generalization:** Many detection systems overfit to specific generators (e.g., GPT-2) and lose effectiveness when applied to outputs from newer or fine-tuned models [20, 21].
- **Short Texts:** Existing methods often struggle with brief or low-context inputs, which lack distinctive linguistic cues [19].
- **Adversarial Robustness:** Generative models can easily rephrase content to evade detection, particularly when no semantic understanding is enforced in the classifier [24].
- **Fairness and Bias:** Studies have shown that detectors tend to misclassify non-native English writing as AI-generated, raising concerns about algorithmic bias [23].

Addressing these challenges will require not only stronger model architectures but also careful attention to evaluation design, data diversity, and social implications of automated detection in real-world applications.

Although fairness and bias are important considerations in AI-generated content detection, this thesis primarily focuses on accuracy and robustness evaluation. A system-

atic fairness analysis, such as subgroup performance across text length, writing style, or demographic variations, is beyond the scope of this work and is left for future research.

1.5 Significance

As AI-generated content becomes increasingly indistinguishable from human-authored text, reliable detection mechanisms are essential to preserve trust in digital communication, academic integrity, and intellectual property rights. The significance of this study lies in its empirical focus on evaluating and comparing deep learning architectures for AIGC detection—a task that remains underexplored despite its growing importance. Unlike studies proposing new detection architectures, this research provides a systematic comparison of widely-used deep learning models—RNN, LSTM, LSTM with Attention, Self-Attention, CNN, and RCNN—under controlled conditions. Such comparative insights can directly inform practical choices for detection system design and deployment, especially in educational, journalistic, and security contexts where performance trade-offs (e.g., between accuracy and speed) are critical [24, 30]. Moreover, this study sheds light on how variations in model architecture affect robustness when dealing with difficult scenarios, including short inputs, diverse generative styles, and content that has been paraphrased or intentionally obfuscated. These insights are valuable for the future development of adaptive and lightweight detection systems that can be deployed in real-time or resource-constrained environments [22, 29]. By benchmarking existing models rather than introducing new ones, this study fills an important gap in the current literature: helping the community understand "what works best, when, and why." It encourages evidence-based model selection and fosters methodological transparency, ultimately supporting the responsible and equitable use of detection tools in the generative AI era.

1.6 Proposed Solutions, Contributions and Impact

1.6.1 Proposed Solutions

This work follows a two-stage plan. First, we establish a unified, apples-to-apples *comparative evaluation* of six representative architectures for AI-generated content (AIGC) detection: **RNN**, **LSTM**, **LSTM with Attention**, **Self-Attention**, **CNN**, and **RCNN**. Second, informed by the comparative findings, we **propose a new hybrid detector, Masked Convolutional Self-Attention (MCSA)**, that fuses multi-scale convolutions (for robust local n -gram/subword cues) with a self-attention encoder (for long-range/global consistency) via position-wise gated fusion.

Unified evaluation protocol. To ensure fair comparison across models and datasets, we adopt:

- **Standardized preprocessing and input formatting** (lowercasing, URL/email removal, numeric/symbol normalization), fixed max sequence length, and a training-only vocabulary with <PAD>/<UNK> handling.
- **Sanitization to suppress lexical shortcuts:** top- K ($K=5000$) discriminative n -grams (1-2-gram) are identified on training text via TF-IDF + χ^2 and mapped to <UNK> at encoding; training further applies word dropout (prob. 0.20).
- **Two complementary protocols** for robustness-aware assessment:
 - **Protocol A (A-all):** sanitization on train/val/test (dropout on train; top- K banning on val/test).
 - **Protocol B (B-train-only):** sanitization on training only; validation/test remain clean, probing distribution shift.
- **Metrics:** Macro-F1 and ROC-AUC as primary scores; robustness is further probed by shuffled word order and by reporting gaps/drops between standard vs. stressed conditions.

Proposed model (MCSA). MCSA employs a *multi-branch depthwise-separable convolutional front-end* (kernels $k \in \{3, 5, 7\}$) to harvest local cues, a *Transformer encoder* for global dependencies, and a *position-wise gating* to adaptively mix local and global evidence at each token. Masked mean/max pooling yields a sequence representation for classification. The model is designed to maintain high standard accuracy while *reducing robustness gaps* under shuffling and train-ban $\rightarrow$ test-clean shift.

1.6.2 Contributions

This research makes the following contributions:

1. **A unified benchmark for AIGC detection.** We provide the first systematic, protocol-driven comparison of six popular deep learning architectures under controlled sanitization and robustness probes, clarifying how inductive biases (local vs. global) shape detector behavior.
2. **A new hybrid architecture (MCSA).** We introduce *Masked Convolutional Self-Attention*, which combines multi-scale convolutions with self-attention via gated fusion. Experiments show that MCSA stays competitive with Self-Attention on standard accuracy while substantially narrowing robustness gaps and approaching CNN-like stability under shuffling and distribution shift.

3. **Reproducible protocol and diagnostics.** We standardize preprocessing, sanitization, training, and reporting (Macro-F1, ROC-AUC, robustness gaps/drops), and release an evaluation pipeline that enables consistent comparisons and future ablations.

1.6.3 Practical Impact

The findings directly inform the selection and design of AIGC detectors in *education*, *journalism*, and *cybersecurity*:

- **Evidence-based model choice:** when standard accuracy dominates, Self-Attention is a strong option; when invariance to word-order perturbations and train-test mismatch matters, CNN is most stable; when a balanced accuracy–robustness trade-off is desired, **MCSA** is recommended.
- **Deployment guidance:** the sanitization protocol and mild robustness probes (e.g., shuffling) provide practical checks against shortcut overfitting and help anticipate performance under real-world shifts.
- **Sustainable benchmarking:** the unified pipeline serves as a transparent, adaptable testbed for tracking future detectors and for integrating additional signals (e.g., watermarking/provenance) in operational settings.

1.7 Thesis Outline

This thesis is structured into five chapters, with each chapter focusing on a major aspect of the study.

- **Chapter 1: Introduction**

Presents the context and motivation for AI-generated content (AIGC) detection, formulates the research questions and objectives, and outlines the overall plan: a unified comparative evaluation of representative architectures followed by the proposal of a new hybrid detector (MCSA). It also previews the study’s contributions and practical relevance.

- **Chapter 2: Literature Review**

Reviews AI-generated content (AIGC) detection techniques spanning early rule-based and statistical methods to contemporary deep learning approaches, and also covers related areas including evaluation frameworks, watermarking and provenance tracking, adversarial robustness, fairness considerations, and challenges arising from dataset shift. The review identifies gaps that motivate a hybrid local–global architecture.

- **Chapter 3: Baselines and Experimental Evaluation**

Describes the unified experimental protocol used to fairly compare six architectures—RNN, LSTM, LSTM with Attention, Self-Attention, CNN, and RCNN—including standardized preprocessing, *sanitization* (top- K n -gram banning and word dropout), and two complementary evaluation protocols (A-all and B-train-only). It reports comparative results (Macro-F1, ROC-AUC, robustness to shuffling and train→test shift) and distills key takeaways about local vs. global inductive biases.

- **Chapter 4: Proposed Method (MCSA) and Results**

Introduces the proposed **Masked Convolutional Self-Attention (MCSA)** model, motivated by Chapter 3. It details the architecture (multi-scale depthwise-separable convolutions, Transformer encoder, position-wise gated fusion), training configuration, and evaluation under the same protocols and diagnostics as the baselines, followed by analyses of accuracy–robustness trade-offs (including shuffling and top- K banning). Figures and tables summarize head-to-head comparisons.

- **Chapter 5: Conclusion and Future Work**

Summarizes the main findings, discusses limitations and broader impact, and outlines future directions (e.g., sparse/linear attention, cross-domain/generalization, model compression, and integration with watermarking/provenance signals).

LITERATURE REVIEW

2.1 Introduction

The field of AI-generated content (AIGC) has rapidly evolved alongside breakthroughs in natural language processing and generative modeling. Text generation has evolved from early rule-based or template-driven methods into advanced large language model (LLM) systems that can produce output nearly indistinguishable from human-written text [3, 4]. This surge in generative capability has created both opportunity and risk across domains including education, media, cybersecurity, and law.

Simultaneously, the need to detect, verify, and analyze AIGC has become a pressing area of research. Early detection methods relied on linguistic heuristics, statistical signals such as perplexity, or rule-based classifiers [25]. However, these methods quickly fell short as generative models advanced in fluency and coherence. More recent work has shifted toward machine learning and deep neural networks for detection, often leveraging transformer-based architectures [19, 22].

This chapter reviews the historical development of AIGC technologies, surveys state-of-the-art detection techniques, and highlights ongoing challenges including model generalization, robustness, and fairness. The review draws primarily from recent academic literature, including survey studies [13, 16], and provides context for the experimental evaluation presented in Chapter 3.

2.2 Rule-based and Symbolic Systems

The development of AI-generated content (AIGC) has gone through several key phases, each marked by distinct modeling paradigms and technological breakthroughs.

Rule-based and Symbolic Systems

The earliest approaches to AI-generated content (AIGC) date back to the symbolic AI era, marked by manually designed rules and structured grammars. Systems from this period relied heavily on symbolic manipulation, logical inference, and predefined templates to produce textual output. Notable examples include ELIZA, a conversational program developed by Weizenbaum in 1966, which mimicked a psychotherapist through pattern matching and keyword-triggered responses [31]. Similarly, SHRDLU, developed by Winograd in 1971, demonstrated natural language understanding within a strictly limited virtual environment using rule-based parsing and semantic interpretation [32].

These early symbolic systems had limited capabilities and were restricted to very narrow application domains. Their outputs typically appeared artificial and repetitive, reflecting a lack of genuine linguistic variation and contextual adaptability. Furthermore, due to their reliance on handcrafted rules and knowledge bases, symbolic systems struggled with scalability and adaptability when extended to broader or more dynamic contexts [33, 34].

Nevertheless, these early efforts laid essential groundwork for the future of automated content generation. The focus on syntactic rules and semantic parsing provided foundational insights into linguistic structure and meaning representation, contributing to later developments in statistical and neural approaches to natural language processing.

Statistical Language Models and SMT

During the 1990s and early 2000s, the field of natural language generation underwent a paradigm shift from symbolic rule-based systems to data-driven statistical approaches. This transition was driven by several factors, including the increasing availability of large digital corpora, improvements in computational power, and the success of probabilistic models in tasks like speech recognition and information retrieval [35].

A key model from this period was the n -gram language model, which predicts the likelihood of a word based on the preceding $n - 1$ words. These models enabled more fluent and flexible generation than symbolic approaches by capturing the local co-occurrence statistics of words. However, they remained limited by their reliance on short contexts and their inability to model long-range syntactic or semantic dependencies [36].

Statistical Machine Translation (SMT) further advanced generative capabilities, particularly in the context of multilingual content creation. Unlike rule-based translation systems, SMT models learned probabilistic alignments between parallel corpora of source and target languages. Phrase-based and hierarchical SMT architectures, such as those used in the Moses framework, significantly improved the fluency and contextual coherence of generated translations [37, 38].

Template-based generation methods also gained popularity during this period, especially for domains like weather reporting, finance, and sports commentary. These systems used statistical models to populate pre-structured sentence templates with relevant entities, verbs, and attributes. While they offered high grammaticality and domain accuracy, they lacked diversity and adaptability in open-ended text generation [39].

Despite these advances, the outputs of statistical generation systems often lacked deep semantic understanding and failed to exhibit meaningful discourse-level coherence. Moreover, detection of such outputs remained relatively easy using heuristic or pattern-based methods, as statistical generators tended to produce repetitive structures and avoid complex phrasing.

Recurrent Neural Networks and Attention Models

The inability of statistical language models to effectively capture long-distance dependencies and higher-level semantic structures prompted the rise of neural network-based sequence modeling approaches in the early 2010s. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), became foundational architectures for modeling temporal and linguistic patterns in sequential data [40, 41].

LSTM networks were specifically designed to overcome the vanishing gradient problem in traditional RNNs by introducing memory cells and gating mechanisms that could retain contextual information over longer sequences. These models allowed for more coherent and grammatically accurate generation, especially in domains like dialogue systems, poetry, and code autocompletion [42, 43].

Despite these improvements, RNN-based models still suffered from limited parallelism during training and were computationally expensive to scale. To address these bottlenecks, attention mechanisms were introduced, enabling models to selectively focus on relevant parts of the input sequence during generation or translation [44]. Attention-based models, often integrated with encoder-decoder architectures, became the de facto standard in neural machine translation (NMT), significantly outperforming traditional SMT systems in both fluency and adequacy.

A landmark architecture in this era was the sequence-to-sequence (Seq2Seq) framework with attention, which allowed flexible input and output length modeling and supported complex generation tasks such as summarization, translation, and question answering [45].

The success of these models marked a critical transition from shallow statistical learning to deep semantic modeling, greatly enhancing the expressive power of generative systems and paving the way for the transformer revolution.

Transformer Architecture and Large Language Models

The introduction of the Transformer architecture by Vaswani et al. (2017) marked a major turning point in natural language generation. In contrast to earlier recurrent architectures, the Transformer models sequence relationships solely through self-attention and position-wise feed-forward networks, eliminating the need for any recurrent operations[2]. This innovation enabled greater parallelization, reduced training time, and improved performance on long-range context modeling.

Building on this breakthrough, subsequent generations of LLMs were created and trained on vast text corpora using various Transformer-based architectures. These include autoregressive models like GPT-2 [46], GPT-3 [3], and GPT-4, which generate text token-by-token based on a left-to-right probability distribution. These models demonstrated remarkable fluency, coherence, and few-shot learning abilities, dramatically expanding the use of AI-generated content in education, journalism, marketing, and creative writing.

In parallel, bidirectional models such as BERT [12] and encoder-decoder architectures like T5 [47] introduced alternative training objectives, including masked language modeling and text-to-text tasks. While not primarily designed for generation, these models were used for conditional generation in summarization, translation, and question answering.

OpenAI's ChatGPT, built upon GPT-3.5 and later GPT-4, leveraged reinforcement learning with human feedback (RLHF) to fine-tune model behavior for dialogue applications. Its conversational quality and accessibility led to unprecedented public adoption, generating over 100 million users within two months of launch.

The rapid advancement of LLMs has brought significant challenges for AIGC detection. Unlike earlier rule-based or statistical systems, LLM outputs closely mimic human writing in terms of syntax, semantics, and style. As a result, traditional stylometric and lexical-based detection methods often fail. The scale, opacity, and closed-source nature of many LLMs further complicate forensic analysis and model attribution.

This technological leap underscores the urgency of developing more robust, adaptable, and generalizable detection methods-topics that are addressed in the following sections.

2.3 Detection Techniques

2.3.1 Overview and Taxonomy

As AI-generated content (AIGC) becomes more pervasive and human-like, the ability to distinguish machine-generated text from authentic human writing has emerged as a critical research challenge. Detection systems aim to identify subtle linguistic, structural, or statistical traces left by generative models and classify whether a given text segment is of human or AI origin.

Existing detection approaches can be broadly categorized along three major dimensions:

- **Input Modality:** Most techniques focus on textual input, but some explore multi-modal detection combining text with metadata, image, or audio features [13, 48].
- **Detection Principle:** Methods range from analyzing stylistic or statistical cues (e.g., perplexity, burstiness), to learning latent representations through supervised or fine-tuned models, to embedding detectable signals via watermarking.
- **Model Awareness:** Some detectors assume access to the generation model (white-box), others work without any assumptions (black-box), and many operate in a semi-informed setting using general transformer pretraining knowledge.

2.3.2 Traditional Stylometric and Statistical Methods

The earliest approaches to detecting AI-generated content focused on handcrafted linguistic and statistical features rooted in stylometry. These methods aim to capture measurable differences in the writing style, lexical distribution, and syntactic structure of human-authored versus machine-generated text.

Perplexity, a widely used metric in language modeling, estimates how well a language model predicts a given text. Lower perplexity indicates higher confidence from the model. Early detection systems used human-trained language models to assign perplexity scores to target text; AI-generated text often had unusually low perplexity because it closely aligned with model training distributions [25].

Burstiness is another signal often used in tools like GPTZero. It captures the variability of sentence-level perplexity. Human writing typically exhibits higher burstiness-variation in complexity and predictability-whereas LLM outputs tend to have uniformly consistent patterns [28].

Stylometric features include type-token ratio (lexical diversity), mean sentence length, part-of-speech distributions, and syntactic parse tree depth. These features have long been used in authorship attribution and deception detection [49]. Some tools use POS tagging and dependency parsing to distinguish AI-generated outputs based on shallow syntactic cues.

A prominent example in this category is **GLTR (Giant Language Model Test Room)**, which visualizes the likelihood of each token by comparing it against the predictions of a reference language model. Tokens generated with high probability are colored green, while surprising tokens are marked in red or purple. Texts dominated by green tokens are often AI-generated [25].

Despite their interpretability and ease of deployment, traditional stylometric methods face major limitations. First, their performance degrades significantly against advanced

LLMs that produce more stylistically diverse and semantically coherent text. Second, handcrafted feature extraction is language- and domain-specific, limiting generalization. Finally, adversaries can easily obfuscate or rewrite content to evade detection based on surface-level metrics.

Nevertheless, these techniques laid the foundation for more advanced machine learning approaches, and some stylometric cues remain useful when integrated into hybrid or explainable detection systems.

2.3.3 Supervised Deep Learning Detection Models

With the rise of large-scale transformer models, supervised learning has become a dominant approach in AIGC detection. These methods train binary or multiclass classifiers to distinguish between human-written and AI-generated text, leveraging the expressive power of deep neural networks to learn latent patterns in language.

Transformer-based detectors, including fine-tuned versions of **BERT** [12], **RoBERTa** [50], **DeBERTa** [51], and **XLNet** [52], have been widely used for supervised detection tasks. These models undergo large-scale pre-training on extensive human-authored corpora and are subsequently fine-tuned on annotated datasets that include both human- and AI-generated samples. Their attention-based architectures allow for robust capture of syntactic and semantic signals across entire text segments.

One of the most notable approaches in this category is **DetectGPT** [22], which uses a model’s own log-likelihood curvature under perturbation as a self-referential detection signal. Instead of training a separate classifier, DetectGPT evaluates how sensitive a model’s output is to small edits-AI-generated text tends to occupy low-curvature regions.

Another influential tool is **GPTZero**, which combines sentence-level perplexity and burstiness metrics to assign a human-likeness score [28]. Though not a neural model itself, GPTZero exemplifies how handcrafted statistical signals can complement deep classifiers in hybrid settings.

To support benchmarking, Uchendu et al. introduced the **TuringBench** framework [24], which aggregates labeled examples from multiple generators (e.g., GPT-2, GPT-3, Grover, XLM) and evaluates supervised classifiers using standardized metrics. Their experiments showed that fine-tuned RoBERTa outperforms traditional ML baselines, but still degrades on unseen generators.

Supervised detectors offer high accuracy, interpretability (through attention heatmaps), and are often easy to integrate into NLP pipelines. However, their performance is strongly tied to the availability, quality, and coverage of labeled training datasets. Many classifiers fail to generalize to outputs from unseen models, prompting the need for continuous retraining or model-agnostic approaches.

2.3.4 Fine-tuning Based Detectors

Fine-tuning has emerged as a common strategy for tailoring pre-trained language models to particular AI-generated content detection tasks. These methods leverage the linguistic knowledge learned from large-scale corpora and adjust the model parameters on a labeled dataset containing both AI-generated and human-written text.

In this approach, a general-purpose transformer model-such as RoBERTa [50], DeBERTa [51], or T5 [47]-is trained on classification objectives to distinguish between human and machine-authored content. Fine-tuning can be performed using standard cross-entropy loss, optionally with data augmentation techniques such as paraphrasing, back-translation, or synthetic adversarial examples to improve generalization.

Studies such as [24] demonstrate that RoBERTa, when fine-tuned on labeled datasets, achieves strong in-distribution detection performance. These models effectively capture subtle distributional and syntactic differences in text, including repetition patterns, token predictability, and discourse coherence.

However, the performance of fine-tuned detectors often degrades on out-of-distribution data, such as content from unseen generators or domain shifts (e.g., from academic writing to social media). Moreover, fine-tuning requires costly labeled datasets, which may not be available for every new generator or language variant. The risk of overfitting to superficial generation artifacts also limits their robustness to paraphrasing or style imitation.

To address these challenges, some research has proposed multi-generator training, continual learning frameworks, or domain-adaptive fine-tuning. However, maintaining performance across unseen generators without sacrificing in-domain accuracy remains an open challenge in AIGC detection.

2.3.5 Zero-shot and Black-box Detection Approaches

To overcome the limitations of supervised and fine-tuned methods, zero-shot and black-box detection techniques have gained traction. These approaches operate without requiring labeled training data or access to the parameters of the generative model. Instead, they rely on intrinsic statistical or structural properties of machine-generated text that remain consistent across different models.

DetectGPT [22] is a landmark zero-shot method that estimates the local curvature of log-likelihood scores under small textual perturbations. The intuition is that AI-generated text typically resides in low-curvature regions of the model’s output space, whereas human-written text exhibits higher sensitivity to perturbation. This approach allows for detection without retraining or supervision, making it applicable across a wide variety of LLMs.

Fast-DetectGPT [53] simplifies this idea further by eliminating the need for perturbation, instead approximating curvature via conditional probability estimates and rank

metrics. It improves detection speed significantly while maintaining robust performance across LLM outputs.

GLTR [25] is another early zero-shot method that uses token rank distributions from a reference language model to identify over-regularity in AI-generated text. Human writing typically contains more unpredictability, while machine outputs are often composed of high-probability, low-surprise tokens.

Self-consistency-based approaches, such as **DetectGPT-SC** and **AuthentiGPT** [54], perform masked token prediction and semantic denoising to estimate whether a given text aligns with expected language patterns. These methods treat the language model as both the generator and the detector, enabling unsupervised detection without relying on explicit generation traces.

Zero-shot detectors are highly attractive for practical deployment because they can adapt to new models and content types without retraining. However, their accuracy is generally lower than that of supervised models, particularly for adversarially paraphrased content or very short texts. Additionally, some approaches still require access to a strong base language model for inference, which may introduce computational overhead or licensing constraints.

Nonetheless, as generative models diversify and labeled data becomes harder to obtain, zero-shot and black-box methods represent a crucial direction for scalable, model-agnostic AIGC detection.

2.3.5.1 Critical Comparison and Motivation for Robustness-Oriented Design

Although supervised deep learning detectors have achieved strong performance on benchmark datasets, they often rely on dataset-specific patterns and may exhibit shortcut learning behaviour. As a result, these models can suffer from performance degradation when evaluated on outputs generated by unseen models or under distribution shift conditions. This limitation highlights the need for evaluation protocols that explicitly assess robustness beyond standard accuracy.

Fine-tuning-based detectors, particularly those built on large pre-trained transformers such as BERT or RoBERTa, typically achieve high classification accuracy by leveraging rich contextual representations. However, these models may overfit to the training distribution and remain sensitive to input variations such as short text length, stylistic differences, or adversarial paraphrasing. Consequently, strong performance under in-domain evaluation does not necessarily translate to reliable detection in real-world scenarios.

Zero-shot and black-box detection approaches aim to improve generalization by avoiding supervised training on specific generators. While these methods demonstrate improved transferability, they often rely on statistical signals such as perplexity, probability curva-

ture, or perturbation-based scoring. These signals may become less reliable as generative models improve, and such approaches may still exhibit instability under robustness stress, including word-order perturbations or domain shifts.

Taken together, these limitations suggest that improving robustness requires both architectural design considerations and evaluation protocols that explicitly test generalization and perturbation resistance. Motivated by this observation, this thesis adopts a unified evaluation protocol incorporating robustness probes and proposes a lightweight hybrid architecture, Masked Convolutional Self-Attention (MCSA). By combining multi-scale convolutional features with global self-attention through position-wise gated fusion, MCSA aims to capture complementary local and global cues, thereby reducing reliance on superficial shortcuts and improving robustness under distribution shift.

In this design, convolutional components are reused to capture local linguistic patterns, self-attention mechanisms are adapted to model long-range dependencies, and the gated fusion mechanism represents a lightweight architectural contribution tailored to robustness-oriented detection.

2.3.6 Watermarking and Provenance-based Detection

Watermarking techniques represent a proactive class of detection strategies that embed identifiable patterns into AI-generated content during the generation process. Unlike classifier-based methods that operate post hoc, watermarking allows for direct attribution and verification by embedding statistical, lexical, or structural signals into generated text [13, 29].

Zero-bit watermarks indicate only whether a watermark is present, typically for binary detection. In contrast, **multi-bit** watermarks encode additional information, such as model ID, timestamp, or author attribution. The design of a watermarking scheme must balance **robustness** (resistance to obfuscation), **semantic fidelity** (preserving fluency and meaning), and **unforgeability** (difficulty for adversaries to replicate).

A popular zero-bit watermarking method is the “green list” sampling algorithm [29], which probabilistically biases token selection to introduce detectable frequency deviations without compromising output quality. Detection is performed by analyzing token frequency deviations under the same language model. This approach is both lightweight and interpretable, but assumes access to the original model or API.

Evaluation of watermarking methods includes metrics such as:

- **Bit Error Rate (BER):** Measures accuracy of multi-bit watermark recovery.
- **Perplexity Shift:** Quantifies degradation in output fluency due to watermarking.
- **Semantic Similarity:** Ensures the embedded watermark does not change meaning.

Detection protocols can be either **private**, where only the model owner can verify the watermark using a secret key or decoder, or **public**, where any third-party can verify watermark presence using an open algorithm. Private watermarks are more secure, but public watermarks promote transparency and trust.

Despite their promise, watermarking techniques face several challenges. Text paraphrasing, synonym replacement, or back-translation can remove or dilute watermark signals. Moreover, open-source models that do not cooperate with watermarking mechanisms remain out of scope. Detection also assumes availability of the original language model or generation trace, which may not always be feasible.

Nonetheless, watermarking offers strong guarantees for content provenance, and is particularly useful in educational, legal, and platform moderation settings where accountability and model traceability are essential.

2.3.7 Summary of Detection Paradigms

The landscape of AI-generated content detection encompasses a diverse array of strategies, each with its own strengths, limitations, and application contexts.

- **Stylometric and statistical methods** provide intuitive and interpretable detection signals based on language complexity, token frequency, and syntactic variation. They are lightweight and easy to deploy but degrade significantly against modern LLMs.
- **Supervised deep learning models** offer strong performance on in-distribution data and can capture nuanced semantic cues. However, they require extensive labeled datasets and retraining to handle new generators or domains.
- **Fine-tuned detectors** adapt pre-trained language models such as RoBERTa or T5 for downstream classification tasks. They perform well on known styles but struggle to generalize, making them suitable for fixed detection settings.
- **Zero-shot and black-box methods** like DetectGPT or GLTR do not require model-specific training, enabling quick adaptation. Their scalability and model-agnostic nature are attractive, though their accuracy remains lower in many settings.
- **Watermarking-based detection** shifts the paradigm toward proactive provenance control. While powerful when embedded at generation time, these methods depend on cooperation from the model provider and may be vulnerable to paraphrasing.
- **Hybrid systems**—combining statistical features, neural classifiers, and watermarks—have emerged to balance detection accuracy, robustness, and interpretability. These ensembles are particularly useful in high-stakes applications such as education, law, and platform moderation.

Each paradigm plays a role in the evolving AIGC detection ecosystem. The choice of detection technique should reflect the deployment context, tolerance for false positives, adversarial risks, and availability of generation-side metadata or API access.

In the next sections, we explore the key challenges that hinder the effectiveness of these approaches and examine promising research directions that may enhance the reliability and fairness of AIGC detection systems.

2.4 Key Challenges in AIGC Detection

While the development of AIGC detection techniques has advanced considerably in recent years, several core challenges persist, limiting the accuracy, generalizability, and fairness of existing systems. These challenges are technical, practical, and ethical in nature, and they reflect the evolving complexity of generative models and the diversity of real-world application scenarios.

1. Generalization Across Generative Models

A major challenge in AIGC detection is ensuring that models trained on outputs from one generative system (e.g., GPT-2) can generalize effectively to content produced by others (e.g., GPT-4, Claude, LLaMA). Most supervised detectors show significant performance degradation when applied to texts from unseen models [19, 24]. This lack of cross-generator robustness is compounded by the closed-source nature of many commercial LLMs and the rapid evolution of new generation techniques.

2. Robustness to Adversarial Rewriting and Obfuscation

Machine-generated content can be easily manipulated using techniques such as paraphrasing, synonym replacement, back-translation, or style transfer. These methods preserve semantics but change surface features, often enough to evade detection by both stylometric and classifier-based systems [21, 23, 26]. Developing detectors that are robust to such adversarial transformations remains an open problem.

3. Detection in Short and Low-context Texts

Many real-world applications involve short-form content-such as tweets, product reviews, or student exam answers-which provides limited linguistic information for classification. Studies have shown that most detection models struggle with inputs under 100 tokens, leading to higher false-positive or false-negative rates [22, 28]. Short texts also increase model uncertainty, making confidence calibration a critical issue.

4. Fairness and Bias in Detection Outcomes

Recent research reveals that many detection systems disproportionately misclassify writing from non-native English speakers as AI-generated, due to stylistic and syntactic differences [23]. Biases also emerge across domains (e.g., academic vs. informal writing) and demographic groups. These issues raise serious ethical concerns in educational, employment, or legal contexts where false accusations of AI usage can lead to reputational or legal consequences.

5. Lack of Interpretability and Transparency

Deep neural detectors often operate as black-box classifiers with little visibility into why a piece of text was flagged. This opacity reduces user trust and complicates deployment in regulated environments. In high-stakes scenarios such as academic integrity enforcement, interpretability is essential for enabling appeals, audits, and policy compliance [55, 56]. Despite advances in attention visualization, most current systems still lack meaningful explanations.

6. Evaluation Inconsistency and Lack of Standard Benchmarks

Different studies use different datasets, generators, and evaluation metrics, making cross-paper comparison difficult. Many datasets are synthetic and fail to reflect the diversity and messiness of real-world content. A standardized, community-accepted benchmark suite-similar to GLUE or SuperGLUE for NLP-remains missing, though efforts like TuringBench and OpenLLMDetector are promising first steps [15, 24].

Together, these challenges highlight the need for more adaptive, transparent, and inclusive AIGC detection methods. Addressing them requires not only algorithmic improvements but also thoughtful dataset design, fairness auditing, and interdisciplinary collaboration across NLP, ethics, and law.

2.5 Summary and Research Gap

This chapter has presented a comprehensive review of AI-generated content (AIGC) and its detection. We first provided a historical survey outlining how the field progressed from early rule-based and statistical approaches to modern deep learning techniques and advanced large language models (LLMs). We then categorized detection techniques into stylometric methods, supervised classifiers, fine-tuned detectors, zero-shot approaches, and watermarking frameworks, analyzing their principles, strengths, and limitations.

Despite significant progress, AIGC detection remains a technically and socially complex problem. Deep neural models, while effective in-domain, suffer from generalization

failures when applied to new generators or low-context inputs. Watermarking methods promise content provenance but rely on cooperation from model providers. Meanwhile, zero-shot detectors offer flexibility but often trade off precision and reliability.

Existing literature also reveals methodological fragmentation: different studies use different datasets, evaluation metrics, and baselines, limiting comparability. In addition, relatively few works conduct large-scale, head-to-head comparisons of multiple neural architectures under consistent conditions. Most focus on one model or one class of detection methods.

Research Gap: While many techniques have been proposed, few studies have systematically compared diverse deep learning architectures-such as RNNs, LSTMs, CNNs, and attention-based models-for AIGC detection using standardized experimental settings. There is also limited understanding of how architectural choices affect detection performance across key variables such as text length, domain, and paraphrasing robustness.

This study addresses that gap by evaluating six popular neural architectures across benchmark datasets, with unified preprocessing and evaluation criteria. The goal is not to propose a new detection algorithm, but to empirically assess how well different model types handle the AIGC detection task. Through this, we aim to inform future detection system design and provide practical insights for deployment in real-world scenarios.

EVALUATION FRAMEWORK AND COMPARISONS

3.1 Introduction

The swift progression of large language models (LLMs) such as GPTs, Claude, and Grok has greatly enhanced the fluency, variability, and overall quality of machine-generated text, making it possible to produce human-like content at a scale that was previously unattainable. While these capabilities bring substantial benefits to areas such as education, creative writing, and software development, they also raise new challenges. In particular, distinguishing between AI-generated content (AIGC) and human-authored text has become increasingly important in contexts where authenticity, originality, and accountability are critical.

Accurate AIGC detection is essential for safeguarding academic integrity, upholding journalistic standards, preventing misinformation, and protecting intellectual property. However, this task is complicated by the growing stylistic similarity between AI and human writing, as well as by the potential overfitting of detection models to superficial lexical cues.

To better understand the strengths and limitations of current detection approaches, it is necessary to evaluate them under controlled conditions, including scenarios where such superficial cues are intentionally reduced. In this study, we concentrate on deep learning-based text classification models-including CNNs, RNNs, LSTMs, Attention-LSTM, Self-Attention, and RCNN-and evaluate their performance on a large-scale corpus comprising both human-authored and AI-generated text.

We further incorporate a set of diagnostic “sanity checks” to probe model behavior, such as random-label tests, word-order shuffling, and the removal of high-frequency terms via sanitization. These evaluations allow us to indirectly infer each model’s reliance

on different types of features, including surface-level lexical patterns and sequential dependencies.

This chapter presents an experimental framework for systematic comparison of six neural architectures for AIGC detection. Section 3.2 describes the dataset and preprocessing steps. Section 3.3 outlines the model architectures and training procedure. Section 3.4 introduces the sanity checks and baseline results. Section 3.5 reports findings from cross-model comparison under strong sanitization. Finally, Section 3.6 summarizes the key insights.

3.2 Dataset Description

The experimental evaluation in this study is conducted using the publicly available *human-vs-AI-generated-dataset* hosted on HuggingFace datasets [57]. This dataset is specifically designed for binary text classification, distinguishing between human-authored and AI-generated content, making it directly relevant to the objectives of this research.

The dataset comprises millions of text instances spanning diverse subject areas, with human-written material primarily drawn from Wikipedia and AI-generated samples produced by GPT-4 and Claude. The wide topical coverage ensures that the dataset captures diverse linguistic styles, subject matters, and levels of textual complexity, which is essential for training models to generalize effectively across domains.

Each data instance is associated with three primary attributes:

- **Text:** Raw textual content varying in length from very short snippets to long-form articles. Most samples fall within a range suitable for sentence- and paragraph-level analysis.
- **Label:** A binary indicator denoting the text origin: ‘0’ for human-authored and ‘1’ for AI-generated content.
- **Source:** Metadata specifying the origin of the text. For AI-generated samples, GPT-4 is the dominant source, while human-authored content comes largely from Wikipedia.

Before training, the dataset underwent a series of preprocessing steps aimed at enhancing quality and model readiness. These included:

1. **Text cleaning** that eliminate extraneous characters, excessive whitespace, and unnecessary special symbols.
2. **Tokenization** to segment the text into discrete tokens that can be effectively used in embedding-based representations.

3. **Word embedding mapping** using pre-trained GloVe vectors with 100-dimensional embeddings to capture semantic relationships between words.
4. **Sequence length normalization** by truncating or padding token sequences to a fixed maximum length to ensure compatibility with deep learning models.

The dataset’s combination of scale, diversity, and high-quality annotation makes it an ideal benchmark for AI-generated content detection. Moreover, its balanced representation of human- and AI-authored text (with a slight real-world class imbalance) allows for realistic performance evaluation. By leveraging this dataset, our experiments aim to assess the robustness, generalization capability, and practical applicability of various deep learning architectures in differentiating between AI-generated and human-written text.

Although the selected datasets provide controlled benchmark settings for evaluating AI-generated content detection, they primarily consist of English-language benchmark-style data. Therefore, the findings of this study should be interpreted within this scope, and generalization to broader domains, multilingual settings, or real-world heterogeneous data remains an important direction for future work.

3.3 Methodology

In this section, we present the deep learning architectures used for detecting AI-generated content. Six representative models are implemented and compared: CNN, RNN, LSTM, RCNN, Attention-based LSTM, and Self-Attention-based models. All models are initialized with 100-dimensional pre-trained GloVe embeddings and trained on the same dataset with identical preprocessing and experimental settings to ensure a fair and consistent comparison.

3.3.1 Convolutional Neural Network (CNN)

Although convolutional neural networks (CNNs)[58] were initially designed for computer vision applications, they have demonstrated strong effectiveness in natural language processing tasks, especially text classification. Within the domain of AIGC detection, CNNs are advantageous because they can identify local textual patterns-such as characteristic n-grams or stylistic signals-that help differentiate human-written content from machine-generated text.

In our implementation, each input text is first tokenized and converted into a fixed-length sequence of tokens. These tokens are mapped to 100-dimensional vectors using pre-trained GloVe embeddings, forming a two-dimensional matrix representation of the sentence (sequence length $\times$ embedding dimension).

Then we employ several parallel one-dimensional convolutional layers, each using different kernel sizes (such as 3, 4, and 5). Each kernel size allows the model to learn features of different granularity:

- Smaller kernels capture fine-grained local word patterns.
- Larger kernels capture broader contextual phrases.

Each convolutional feature map is processed with a ReLU activation, after which max-over-time pooling is applied to preserve the most informative feature from each map. The pooled outputs from all convolutional layers are concatenated into a single feature vector, which is then regularized by a dropout layer with $p = 0.5$ to reduce overfitting.

Finally, the resulting vector is fed into a fully connected linear layer that outputs classification logits. This architecture allows CNNs to efficiently learn discriminative local patterns while remaining computationally efficient, making them particularly suitable for large-scale AIGC detection tasks.

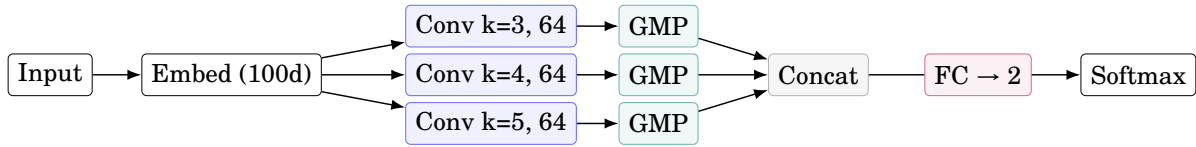


Figure 3.1: TextCNN architecture.

3.3.2 Recurrent Neural Network (RNN)

RNNs[59] constitute a family of models built to handle sequential inputs by updating an internal hidden state as each element in the sequence is processed. Unlike standard feed-forward architectures, RNNs are capable of capturing temporal relationships within sequences, which makes them particularly effective for natural language tasks where word order conveys essential meaning.

In this architecture, each input token is first mapped to a dense vector representation, and then processed sequentially by a bidirectional RNN layer. The bidirectional setup allows the model to capture contextual information from both past and future tokens, providing a more comprehensive understanding of the sentence. Stacking multiple RNN layers increases the model’s ability to capture more complex and abstract patterns within the data.

After the recurrent layers, a dropout layer is introduced to mitigate overfitting, and a subsequent fully connected layer transforms the final hidden-state representation into the desired classification output.

The main strength of RNNs lies in their ability to remember dependencies over varying time spans, making them effective for detecting subtle stylistic patterns or

context shifts that might distinguish AI-generated text from human-authored content. However, standard RNNs often face difficulties when processing long sequences because of vanishing or exploding gradients, which has led to the development of more sophisticated recurrent architectures such as LSTMs.

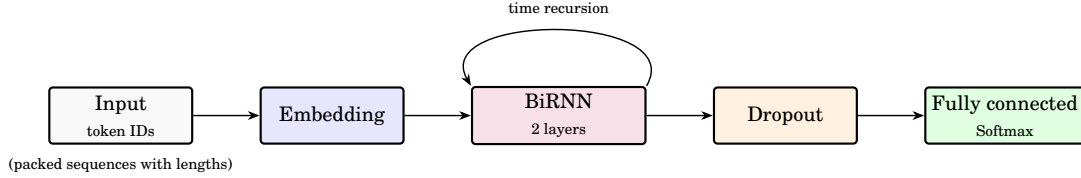


Figure 3.2: RNN architecture (2-layer bidirectional RNN).

3.3.3 Long Short-Term Memory (LSTM)

LSTM[40] network extends the standard recurrent neural network by introducing gating mechanisms that control how information is retained or discarded over time. These gates—input, forget, and output—enable the model to preserve relevant information across long sequences, mitigating the vanishing gradient problem that limits traditional RNNs.

In our setup, the LSTM ingests each sentence as a sequence of word embeddings and incrementally constructs a contextual representation by integrating information from both nearby and far-reaching tokens. This allows the model to identify subtle dependencies, such as the influence of early sentence structure on later meaning, which is critical in distinguishing between AI-generated and human-written content.

Structurally, the model consists of:

- An embedding layer that maps tokens to dense vector representations.
- A unidirectional LSTM layer with a hidden state size of 128 units, optimized for capturing sequential patterns.
- Dropout regularization to prevent overfitting.
- A fully connected layer that converts the learned sequence representation into classification scores.

By maintaining a controlled flow of information through its gates, the LSTM effectively models long-range dependencies and stylistic patterns, enhancing its ability to detect nuanced differences between AI-generated and human-authored text.

3.3.4 Recurrent Convolutional Neural Network (RCNN)

RCNN[60] combines the strengths of recurrent and convolutional architectures to capture both sequential context and local patterns in text. While the recurrent component models

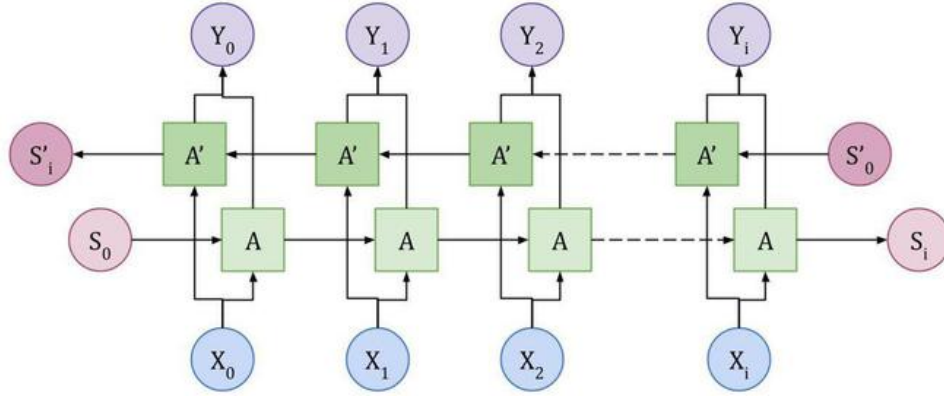


Figure 3.3: Bidirectional-LSTM: X-Input, Y-Output, A-Forward & Backward units

long-term dependencies, the convolutional layers focus on extracting local features such as key phrases and n-grams.

In our configuration, the RCNN first uses a bidirectional LSTM to encode the text, generating forward and backward context for each word. These context-aware word representations are then passed through a non-linear transformation and pooled using a max-over-time strategy, producing a compact sentence-level representation.

The architecture is composed of:

- An embedding layer that provides semantic representations for each token.
- A bidirectional LSTM to capture both preceding and succeeding word context.
- Concatenation of the contextual representation with the original word embedding for richer feature representation.
- A max-pooling layer to select the most informative features.
- A fully connected layer that produces the final classification output.

By combining sequence modeling with convolution-based feature extraction, RCNNs can identify intricate stylistic signals within text, which makes them especially effective at distinguishing human-written content from AI-generated material.

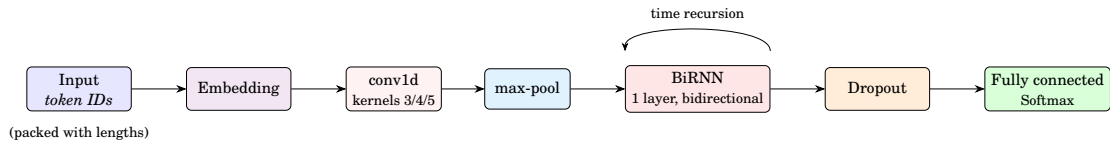


Figure 3.4: RCNN architecture.

3.3.5 Attention-based LSTM

The Attention-based LSTM [44] extends the standard LSTM architecture by adding an attention mechanism that assigns dynamic importance weights to each token in the

sequence. Rather than giving all words equal influence, the attention module highlights the portions of the text most relevant to the classification objective.

The model begins with a bidirectional LSTM encoder that captures contextual relationships in both forward and backward directions. The attention mechanism then evaluates each hidden state, assigning higher weights to tokens that contribute more significantly to the final prediction. This produces a weighted context vector that serves as a refined representation of the sentence.

Key components include:

- An embedding layer to transform tokens into dense vectors.
- A bidirectional LSTM layer to encode the full context of the sentence.
- An attention layer that computes relevance scores for each token.
- A dropout layer for regularization.
- A fully connected classification layer.

By focusing on the most informative segments of text, the Attention-based LSTM improves both interpretability and detection accuracy, offering insight into which parts of the content most strongly signal AI or human authorship.

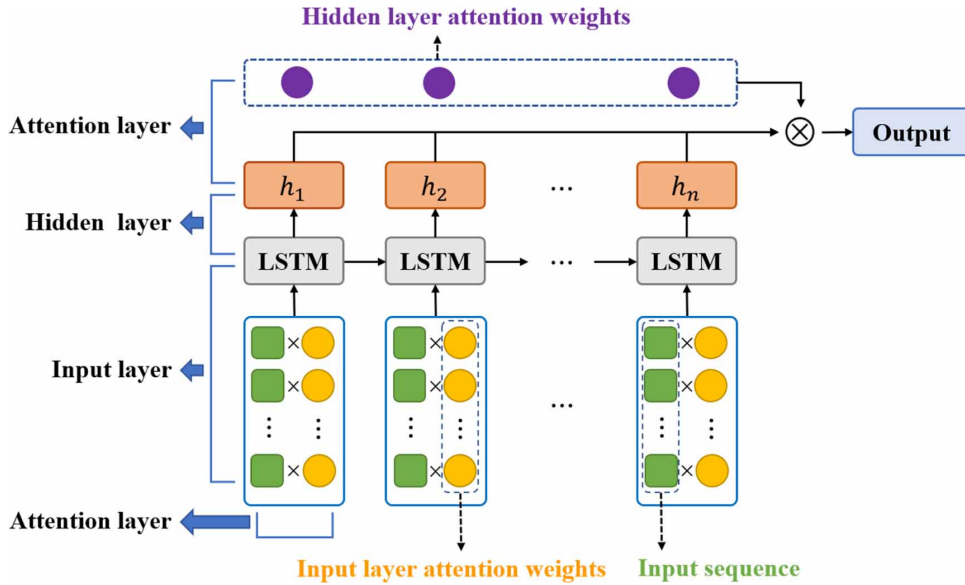


Figure 3.5: Attention-based LSTM [1]

3.3.6 Self-Attention-based Model

The Self-Attention-based model[2] is designed to capture multiple semantic aspects of the input sequence by applying multi-hop attention on top of a bidirectional LSTM.

The BiLSTM first encodes the contextual information in both forward and backward directions, producing hidden states of dimension $2 \times \text{hidden_size}$.

These hidden states are then transformed through a two-layer attention mechanism, where the first linear layer projects them to an intermediate attention space of dimension $d_a = 350$, and the second linear layer maps them into $r = 10$ distinct attention maps.

Each attention head emphasizes distinct semantic regions within the sentence, allowing the model to capture a diverse set of complementary features. The resulting r sentence-level representations are then concatenated and fed into a fully connected layer with 2000 units, which in turn connects to the final classification layer.

This design enables the model to integrate information from multiple interpretive viewpoints, thereby enhancing its capacity to identify subtle differences between AI-generated and human-authored text.

3.3.6.1 Interpretability Analysis of the Self-Attention Model

A key benefit of self-attention architectures is their interpretability, as the attention weights can highlight which words or phrases play the most significant role in guiding the model’s predictions. To investigate this aspect, we visualize the learned attention distributions for selected test samples.

For each input sentence, the multi-hop self-attention mechanism produces r distinct attention maps, with each map highlighting a different semantic aspect of the text. In our implementation, r is set to 30, meaning that the model attends to 30 different contextual perspectives for every sentence. These attention maps are normalized so that the sum of weights across the sequence is equal to one, allowing for direct comparison between different tokens.

We present case studies comparing human-generated and AI-generated texts. For human-written sentences, attention tends to be distributed across diverse syntactic and semantic elements, reflecting the model’s focus on varied linguistic cues. In contrast, AI-generated texts often exhibit concentrated attention on specific repetitive phrases or statistically salient tokens, suggesting that the model is detecting patterns characteristic of machine-generated content.

This interpretability not only provides insights into the internal reasoning of the classifier but also increases trust in the detection process, especially in contexts where explainability is crucial, such as academic integrity or legal compliance.

3.4 Experimental Setup and Sanity Checks

3.4.1 Data Preprocessing

We apply a lightweight but targeted pipeline before model training:

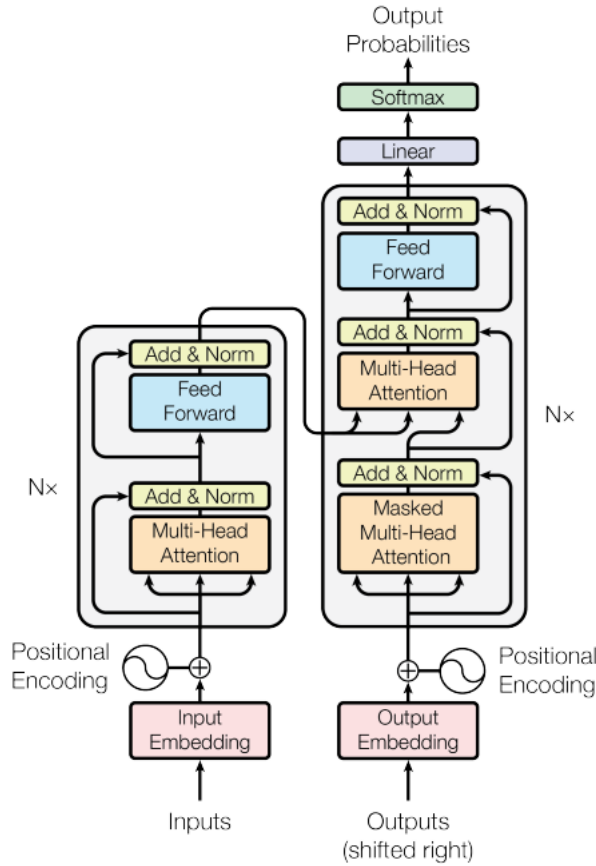


Figure 3.6: Self-attention architecture [2]

- **Style scrub.** We normalize case and remove dataset-specific markup and noise (URLs, emails, bracketed citations, numeric placeholders, wiki section headers).
- **Exact de-duplication.** We compute an MD5 hash for each normalized text and drop duplicates to avoid train/test leakage.
- **Length-balanced sampling.** To reduce trivial cues from text length, we stratify by quantiles of token length and balance classes within each bin.
- **Tokenization & vocab.** We perform word-level tokenization and build a vocabulary (no cap when sanitize=false); sequences are truncated/padded to `max_seq_len=200`. For all deep models we initialize 100-dimensional GloVe embeddings.

These steps aim to remove obvious artifacts while preserving the semantics needed for downstream detection.

3.4.2 Training Protocol and Metrics

Unless otherwise stated, we use the following settings for all six architectures (CNN, RNN, LSTM, RCNN, Attention-based LSTM, Self-Attention):

- **Embeddings:** GloVe-100d (frozen); **sequence length:** 200; **batch size:** 64.
- **Optimizer:** Adam ($\text{lr}=10^{-3}$, $\text{weight decay}=10^{-4}$), dropout keep probability 0.6.
- **Training:** up to 12 epochs with early stopping on validation loss.
- **Splits:** 80% train/val and 20% test; within the 80%, a 90/10 split forms train/val.
- **Metrics:** macro-F1 (primary), accuracy, ROC-AUC. We report test scores on the held-out standard test set.

Hardware: single-GPU environment. Empirically, CNN trains ~ 3 minutes/epoch; Self-Attention > 20 minutes/epoch, others in between.

3.4.3 Standard Test Set Results

Table 3.1 summarizes the baseline performance of all six models when trained on the original dataset without any sanitization (`sanitize=false`). The results are strikingly high across the board: all models achieve macro-F1 scores above 0.985, with Self-Attention, RCNN, and LSTM exceeding 0.992.

Table 3.1: Performance on the standard test set without sanitization.

Model	Accuracy	Macro-F1	ROC-AUC
RNN	0.9880	0.9880	0.9991
LSTM	0.9916	0.9916	0.9996
Attention-LSTM	0.9916	0.9916	0.9996
Self-Attention	0.9929	0.9929	0.9997
CNN	0.9851	0.9851	0.9989
RCNN	0.9928	0.9928	0.9997

Preliminary Observations. While these results might suggest that the task is essentially solved, such uniformly high performance warrants caution. Several characteristics of the dataset raise the possibility that models are exploiting superficial cues rather than learning genuine semantic distinctions:

- **Source disparity:** All human-written samples come exclusively from Wikipedia, whereas AI-generated samples originate from multiple LLMs. This introduces stylistic and domain-specific biases that may be easily detected.

- **Token distribution differences:** Preliminary token frequency analysis revealed systematic vocabulary usage gaps (e.g., Wikipedia-specific templates, citation markers, and topic distributions).
- **Length patterns:** Despite length matching during preprocessing, residual differences in sentence structure and information density could still provide indirect class signals.

These observations motivate a deeper investigation into model robustness. In particular, we need to verify whether these models truly capture semantic differences or simply memorize shallow patterns in the data. This leads to the design of the sanity checks described in Section 3.4.4.

3.4.4 Sanity Checks: Motivation and Design

To verify whether the high baseline scores reported in Section 3.4.3 reflect genuine semantic understanding or are instead driven by spurious correlations, we design four targeted *sanity checks*. Each check has a distinct purpose, manipulates the dataset in a controlled manner, and yields diagnostic outcomes that help isolate the source of the performance.

Sanity Check 1: Random Label Test. **Motivation:** If a model truly learns semantic patterns, training on randomly shuffled labels should prevent it from achieving meaningful performance. Any macro-F1 score significantly above random guessing (0.5 for binary classification) would indicate that the model is memorizing dataset-specific cues unrelated to the label semantics. **Design:** We randomly permute the labels in the training set while keeping the validation set labels intact, then retrain the model from scratch. **Expected outcomes:**

- *Macro-F1* ≈ 0.5 : Model behaves as expected; cannot learn from random labels.
- *Macro-F1* > 0.5 : Model is overfitting to superficial token patterns that correlate with the original (non-random) labels, possibly due to data leakage or strong domain/style cues.

Sanity Check 2: Word Order Shuffling Test. **Motivation:** Many deep text classifiers rely on contextual word sequences. If models can still perform well when the order is destroyed, it suggests reliance on bag-of-words style cues (e.g., token frequency) rather than sequence modeling. **Design:** We randomly shuffle the word order of each sentence in the validation set while keeping the vocabulary intact. The model is evaluated without retraining. **Expected outcomes:**

- *Sharp performance drop*: Indicates that the model depends on sequential structure for classification.
- *Minor performance drop*: Suggests that sequence order is not essential and that the model primarily leverages token-level statistics or domain-specific vocabulary.

Sanity Check 3: TF-IDF + Logistic Regression Baseline. **Motivation:** A simple bag-of-words classifier can serve as a lower-bound benchmark. If such a model achieves performance comparable to deep neural models, it implies that high scores may be driven by surface-level lexical cues. **Design:** TF-IDF features are computed from the training data and used to fit a logistic regression classifier, which is then assessed on the validation set. **Expected outcomes:**

- *Performance far below deep models*: Neural models likely benefit from more complex context representation.
- *Performance close to deep models*: Suggests that bag-of-words information alone is sufficient for classification, raising concerns about the true complexity of the task.

Sanity Check 4: Top-K Term Masking Test. **Motivation:** If models rely heavily on a small set of highly predictive terms (e.g., Wikipedia templates, citation markers), masking these terms should cause a significant drop in performance. **Design:** We identify the top K most predictive terms from the training set using mutual information, then replace them with a special <MASK> token in the validation set before evaluation. **Expected outcomes:**

- *Significant performance drop*: Confirms that the model relies on a small set of token-level cues.
- *Minimal performance drop*: Indicates that the model uses broader contextual or semantic information beyond the top predictive terms.

While the sanitization protocol aims to remove dataset-specific shortcuts, it may introduce potential side effects. For example, removing top- K n-grams could disproportionately affect one class if certain frequent patterns are more common in either human-written or AI-generated text. Additionally, sanitization may introduce artificial distribution shifts that differ from naturally occurring real-world variations. To mitigate these concerns, the sanitization procedure was applied consistently across datasets and evaluated using both sanitized and non-sanitized settings, allowing robustness comparisons while minimizing class-specific bias.

3.4.5 Sanity Check Results and Analysis

Table 3.2 reports the macro-F1 scores for all six models under the four sanity checks described in Section 3.4.4. The TF-IDF+LR baseline is shown once, as it depends only on the dataset rather than the neural architecture.

Table 3.2: Sanity check results (macro-F1). The TF-IDF+LR baseline is computed once on validation data.

Model	Random Label	Shuffle Order	TF-IDF+LR	Ban Top-K (500)
RNN	0.5124	0.7480	0.9693	0.8826
LSTM	0.4224	0.8463	–	0.9264
Attention-LSTM	0.3391	0.7870	–	0.9283
Self-Attention	0.3616	0.8485	–	0.8682
CNN	0.3975	0.8998	–	0.8930
RCNN	0.3333	0.7881	–	0.8680
Data baseline (TF-IDF+LR)		0.9693 (val macro-F1)		

Key observations. Several important findings emerge from the sanity check results:

- **Strong surface-level cues.** The TF-IDF+LR baseline achieves a macro-F1 of 0.9693 on the validation set, only slightly below the 0.98-0.99 range of deep models, indicating that bag-of-words features alone can nearly match them. This strongly suggests that the dataset contains domain- or style-specific lexical patterns (e.g., Wikipedia formatting, citation markers) that correlate with the labels.
- **Varying sensitivity to word order.** Under word-order shuffling, CNN retains the highest performance (0.8998) among all models, suggesting that it is relatively robust to order disruption—likely due to its reliance on local n-gram patterns rather than long-range dependencies. In contrast, recurrent models like RNN drop more substantially (to 0.7480), reflecting their stronger dependence on sequential structure.
- **Different reliance on high-information terms.** The Top-K Term Masking Test reveals that LSTM and Attention-LSTM maintain the highest performance (~ 0.926 – 0.928) even after masking 500 top predictive terms, suggesting a broader distribution of predictive cues in their learned representations. CNN drops to 0.8930, while Self-Attention and RCNN are more affected (~ 0.868).
- **Random label resistance is minimal, as expected.** All models produce macro-F1 close to 0.5 under Random Label Testing, with small deviations (e.g., RNN at 0.5124) likely due to overfitting noise or small non-random artifacts in the training process. Notably, the RNN achieves a macro-F1 of 0.5124 under random labeling, which is substantially higher than the ~ 0.50 expected for a balanced binary task.

This suggests that even with randomized labels, the model may still be exploiting superficial token distribution patterns or dataset artifacts, rather than learning meaningful semantic relationships.

Choice of the pivot model. For systematic parameter sensitivity (Section 3.5), we select CNN as the pivot model because:

- (i) **Diagnostic responsiveness:** CNN exhibits the clearest and most interpretable performance drops under both shuffling and term masking, making it sensitive to the manipulations we wish to explore in ablation studies.
- (ii) **Balanced dependence on cues:** It shows notable reliance on both local order and high-frequency terms, thus allowing us to observe how sanitization parameters (e.g., `ban_topk`, word dropout) interact with different types of signal.
- (iii) **Efficiency:** CNN trains in approximately 3 minutes per epoch on our setup, compared to 20+ minutes for the Self-Attention model, enabling wider and faster parameter sweeps.

Lead-in to Section 3.5. In the next section, we will conduct a systematic parameter sensitivity study on CNN, varying sanitization parameters to identify a configuration that mitigates spurious correlations while preserving generalization. This optimized configuration will then be applied to all six models to evaluate how different architectures behave under comparable, controlled sanitization conditions.

3.5 Cross-Model Comparison under Strong Sanitization

CNN under different sanitization intensities. Table 3.3 summarizes CNN performance under mild, moderate, and strong sanitization settings. These configurations progressively increase `ban_topk` (1500 $\rightarrow$ 5000) and word dropout rate (0.05 $\rightarrow$ 0.20). Macro-F1 are identical here because the dataset is perfectly balanced and both classes achieve similar precision and recall.

Table 3.3: CNN performance under different sanitization intensities.

Setting	Macro-F1	Shuffle-F1	Sanitization Params
Mild	0.8709	0.8553	<code>ban_topk=1500</code> , <code>dropout=0.05</code>
Moderate	0.8675	0.8464	<code>ban_topk=3000</code> , <code>dropout=0.10</code>
Strong	0.8662	0.8462	<code>ban_topk=5000</code> , <code>dropout=0.20</code>

Performance declines steadily as sanitization strength increases, confirming that both high-frequency token removal and word dropout weaken reliance on surface-level lexical cues. To evaluate the impact of strong sanitization on model performance, we applied the most aggressive preprocessing settings identified in preliminary trials: `ban_topk=5000` and `word_dropout=0.20`. These settings aim to remove high-frequency lexical cues and partially randomize the surface form of the text while preserving its semantic content, thereby reducing the availability of domain- or style-specific artifacts.

Results. Table 3.4 reports the macro-F1 scores and shuffled-order performance (Sanity Check 2) for all six deep models under strong sanitization. As a reference, the TF-IDF+LR baseline (data-only) achieves 0.9693 macro-F1 without sanitization.

Table 3.4: Performance of all models under strong sanitization settings. The shuffled-order column shows macro-F1 on Sanity Check 2.

Model	Macro-F1	Shuffle F1
RNN	0.8384	0.7355
LSTM	0.8615	0.7355
Attention-LSTM	0.8818	0.7498
Self-Attention	0.9238	0.7838
CNN	0.8762	0.8462
RCNN	0.8575	0.7715

Analysis. Several observations emerge:

- **Performance drops after sanitization.** Once the shortcut n-grams are removed, the overall difficulty of the task increases notably. Macro-F1 now ranges from 0.83 to 0.92, compared to roughly 0.97–0.98 before sanitization. This confirms that sanitization successfully removes the most discriminative lexical cues, forcing models to rely on deeper semantic features.
- **Variation in order sensitivity.** As shown by Sanity Check 2 results, sequence-modeling architectures (RNN, LSTM, Attention-LSTM, RCNN) still exhibit large performance drops under word-order shuffling (around 0.73–0.77), indicating their strong dependence on token sequence. In contrast, the CNN maintains a much higher Shuffle F1 (0.8462), reflecting stronger robustness to local perturbations. Self-Attention achieves the highest clean score (0.9238) but also shows moderate order sensitivity.
- **Moderate overall adaptation.** Despite lower scores, all deep models remain above 0.83 Macro-F1, suggesting that they can still adapt to sanitized inputs by exploiting remaining semantic and stylistic regularities. Sanitization increases task realism

but does not make it prohibitively difficult, which motivates exploring hybrid designs such as MCSA to balance accuracy and robustness.

Conclusion. Strong sanitization noticeably reduces model performance, confirming that the removal of highly discriminative lexical cues makes the task more challenging. Nevertheless, this finding remains informative: it highlights both the resilience of modern deep classifiers and the limitations of purely lexical sanitization strategies. In subsequent experiments, we therefore apply these sanitization settings across all architectures to ensure fair and consistent cross-model comparison under more realistic conditions.

Due to computational constraints, experiments were conducted using fixed random seeds to ensure deterministic evaluation. While multiple runs with variance reporting may provide additional statistical confidence, the consistent trends observed across models and evaluation protocols suggest that the reported results remain representative for comparative analysis. Future work may further investigate statistical variance across multiple runs.

3.6 Summary

This chapter established a rigorous, unified framework for evaluating AI-generated content detection models. Section 3.1 introduced the goals and motivations that drive our experimental design. Section 3.2 detailed the dataset sources, composition, and structural properties. Section 3.3 described the six architectures under study (CNN, RNN, LSTM, Attention-LSTM, Self-Attention, RCNN) and clarified their inductive biases (e.g., local n -gram vs. long-range dependency modeling).

In Section 3.4, we specified a standardized evaluation protocol, including text cleaning, vocabulary handling, and a *sanitization* procedure that suppresses lexical shortcuts (top- K banning and word dropout). We further defined two complementary evaluation regimes to probe robustness: a fully sanitized setting and a train-sanitized/test-clean setting that induces distribution shift. Baseline results under these regimes revealed consistent patterns: while standard accuracy is competitive across models, sensitivity to word-order perturbation and removal of highly discriminative tokens differs markedly by architecture.

Finally, Section 3.5 explored the impact of sanitization strength and related configuration choices. Mild to moderate sanitization generally led to small performance changes, whereas stronger sanitization amplified differences in reliance on lexical vs. sequential cues. Together, these findings identify complementary strengths-e.g., CNNs are more invariant to local reordering, while self-attention excels on standard accuracy-and motivate a hybrid approach that can reconcile local robustness with global consistency.

Overall, Chapter 3 provides the empirical basis and diagnostic evidence that motivate the new model introduced in the next chapter. In Chapter 4, we present *Masked Convolutional Self-Attention (MCSA)*, describe its architecture and training, and evaluate it under the same protocols to assess whether it achieves a stronger accuracy–robustness trade-off than individual baselines.

A MASKED CONVOLUTIONAL SELF-ATTENTION (MCSA) APPROACH

4.1 Motivation

Chapter 3 evaluated a spectrum of architectures for text classification, including **CNN**, **RNN/LSTM**, **RCNN**, **Attention-LSTM**, and **Self-Attention**. While their standard accuracy was broadly comparable (high Macro-F1), we observed structural differences in *inductive biases* and *sensitivity to perturbations*. In particular, **CNN** emphasizes *local n-gram and boundary cues* and is relatively insensitive to mild word-order changes, whereas **Self-Attention** emphasizes *long-range dependencies and global ordering*, offering advantages when global alignment is required. For robustness-focused analysis, we therefore selected **CNN** and **Self-Attention** as two *representative baselines* along the “local invariance vs. global alignment” axis.

Our preliminary diagnostics further indicated that under **word-order shuffling** and **top-K lexical banning**, architectures exhibit different *robustness gaps* (standard F1 minus stressed F1). Under **distribution shift**—training on sanitized text but validating/testing on clean text—some models also show larger *relative drops*. This suggests that single-path models (purely local or purely global) struggle to jointly optimize for high accuracy and high robustness.

Motivated by these observations, we propose **MCSA** (Masked Convolutional Self-Attention) as a *diagnostically motivated hybrid architecture*, rather than as a fully novel standalone design. MCSA is intended as an incremental but practically motivated architectural contribution that combines a *multi-scale convolutional front-end* for stable local cues and perturbation buffering with a *self-attention encoder* for long-range and global consistency. A *gated fusion* adaptively balances local vs. global evidence at each position.

To rigorously evaluate MCSA, we adopt the dual protocols consistent with Chapter 3: **Protocol A** (sanitization on train/val/test) and **Protocol B** (sanitization on train only; clean val/test). We report Macro-F1 and ROC-AUC, and quantify stability using *robustness gaps* and *relative drops*. Our study addresses two questions: (1) Can MCSA *reduce* robustness gaps under Shuffle and Top- K stress relative to pure self-attention without sacrificing standard accuracy? (2) Under distribution shift from sanitized training to clean evaluation, can MCSA achieve a *relative drop* comparable to or better than CNN?

4.2 Related Work

From single-path to hybrid paradigms. Text classification has evolved from *convolutional* paradigms that emphasize local n -gram and boundary cues (e.g., TextCNN) to *self-attention* paradigms that emphasize global, long-range interactions [2], and more recently to *hybrid CNN–Self-Attention* paradigms. Pure CNNs are relatively stable under mild order perturbations yet limited on long-range reasoning; pure self-attention often excels on standard accuracy but is more sensitive to ordering and lexical triggers. Hybrid designs aim to combine the strengths of both.

Hybrid evidence in NLP. In NLP, several lines of work demonstrate the effectiveness and portability of convolution–attention hybrids: (i) **Reading comprehension:** QANet [61] stacks depthwise separable convolutions with self-attention, achieving fast and accurate modeling via local aggregation plus global alignment. (ii) **Lightweight/efficient attention:** LightConv/DynamicConv [62] approximate/augment attention with convolutions, suggesting interchangeability and complementarity; ConvBERT [63] injects convolution into attention to enhance local modeling. (iii) **Long documents:** Longformer [64] and related sparse attention families mix local windows with global tokens, echoing the local–global division of labor found in conv–attention hybrids. (iv) **Text classification:** CNN+Attention variants commonly report steadier or higher accuracy on benchmarks (20News, IMDB, Amazon), especially when categories hinge on both local cues and contextual dependencies.

Cross-modal support (speech/vision/medical). Beyond NLP, hybrid architectures are well established. In **ASR**, the Conformer [65] inserts convolutional modules into Transformer blocks to jointly model short-term locality and long-term dependencies, becoming a dominant backbone. In **vision**, CoAtNet [66] and ConViT [67] fuse convolution and attention hierarchically, where early conv layers harvest fine-grained textures and later attention layers integrate global relations; medical imaging similarly benefits from CNN–Transformer hybrids. These cross-modal successes provide strong external

evidence: when tasks require both *stable local cues* and *global coordination*, hybrids offer a structural advantage.

Why hybrids fit AI-text detection. In human-vs-LLM authorship detection, highly discriminative n -grams/trigger phrases can induce *lexical shortcuts*. Under *sanitization* (top- K banning and word dropout), single-path models (pure attention or pure convolution) often face an accuracy–robustness trade-off. A hybrid is well suited here: a *convolutional front-end* buffers mild order perturbations and surface rewrites, while a *self-attention encoder* aligns global semantics across long ranges. A *gated/weighted fusion* can adaptively allocate trust between local and global evidence at each position, *narrowing robustness gaps* (standard vs. stressed F1) and improving train-sanitized $\rightarrow$ test-clean stability.

Position and originality. Existing surveys/systems for LLM-generated text detection predominantly rely on *Transformer-based* detectors and report standard metrics, but rarely conduct a *systematic robustness study* of *convolution-augmented self-attention* under unified sanitization with dual protocols (A/B), word-order *Shuffle*, and strong top- K bans. Positioning against prior CNN–Attention hybrids (e.g., QANet, Conformer, ConvBERT, CoAtNet), our **MCSA** differs in that it:

- employs a *parallel* multi-scale depthwise-separable CNN front-end ($k \in \{3, 5, 7\}$) rather than sequentially stacking conv and attention;
- projects to align dimensions before the Transformer encoder and *does not* add explicit positional embeddings, as local positional cues are injected by convolutions;
- uses *position-wise gated fusion* to adaptively balance local (P) and global (H_{attn}) evidence, instead of plain concatenation or residual summation;
- aggregates with *mask-aware mean+max pooling* for stable classification under variable-length inputs.

Under the same sanitization setup as Chapter 3, we **design and evaluate** MCSA with a dual-protocol (A/B) and multi-perturbation (Shuffle/Top- K) evaluation, and analyze *Robustness Gap* and *relative drop*. Quantitatively, structural complementarity yields practical robustness gains: MCSA maintains competitive standard accuracy while consistently narrowing robustness gaps relative to pure self-attention and approaching CNN-like stability under distribution shift.

4.3 Method: MCSA Architecture and Formulation

Notation. Given a token sequence $\mathbf{x} = (x_1, \dots, x_L)$, with vocabulary size V and embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times d_e}$, we obtain

$$\mathbf{X} = [\mathbf{e}_{x_1}, \dots, \mathbf{e}_{x_L}] \in \mathbb{R}^{L \times d_e},$$

and a padding mask $\mathbf{M} \in \{0, 1\}^L$ (1 for valid tokens).

(1) Multi-scale convolutional front-end. To harvest stable local n -gram cues, we use *multi-scale depthwise-separable convolutions* with kernel set $\mathcal{K} = \{3, 5, 7\}$:

$$\mathbf{C}_k = \text{Dropout} \circ \text{LN} \circ \phi \circ \text{PW} \circ \text{DW}_k(\mathbf{X}) \in \mathbb{R}^{L \times c}, \quad k \in \mathcal{K}.$$

where ϕ is GELU, DW/PW denote depthwise/pointwise conv. We then concatenate

$$\mathbf{H}_{\text{conv}} = \text{Concat}_{k \in \mathcal{K}} \mathbf{C}_k \in \mathbb{R}^{L \times (c|\mathcal{K}|)}.$$

(2) Projection to attention dimension. We align dimensions via a linear projection to d :

$$\mathbf{H}_c = \mathbf{H}_{\text{conv}} \mathbf{W}_p + \mathbf{b}_p \in \mathbb{R}^{L \times d}.$$

(3) Self-attention encoder. A Transformer encoder with N layers and h heads processes $\mathbf{H}_c$ under the padding mask $\mathbf{M}$:

$$\mathbf{H}_a = \text{TransformerEncoder}(\mathbf{H}_c; \mathbf{M}) \in \mathbb{R}^{L \times d}.$$

(4) Gated fusion (adaptive local-global mix). To avoid naive summation, we fuse local/global evidence via position-wise gating:

$$\alpha = \sigma([\mathbf{H}_c; \mathbf{H}_a] \mathbf{W}_g + \mathbf{b}_g) \in \mathbb{R}^{L \times 1}, \quad \mathbf{H} = \text{LN}(\alpha \odot \mathbf{H}_a + (1 - \alpha) \odot \mathbf{H}_c).$$

(5) Masked pooling and classifier. We compute masked mean/max pooling:

$$\mathbf{h}_{\text{mean}} = \frac{\sum_{i=1}^L (M_i \mathbf{H}_i)}{\sum_{i=1}^L M_i}, \quad \mathbf{h}_{\text{max}} = \max_{i: M_i=1} \mathbf{H}_i, \quad \mathbf{z} = [\mathbf{h}_{\text{mean}}; \mathbf{h}_{\text{max}}] \in \mathbb{R}^{2d}.$$

The logits are

$$\mathbf{o} = \mathbf{W}_{\text{cls}} \mathbf{z} + \mathbf{b}_{\text{cls}} \in \mathbb{R}^2.$$

(6) Training objective and implementation. We optimize the cross-entropy loss $\mathcal{L} = \text{CE}(\mathbf{o}, y)$ with optional class weights; the <PAD> embedding is zeroed. Typical hyperparameters are $c=128$, $|\mathcal{K}|=3$, $D=256$, $h=4$, $N=2$, with LayerNorm and dropout applied to improve stability.

(7) Complexity and intuition. The convolutional front-end costs $\mathcal{O}(L \cdot d_e \cdot \sum_k k)$ with reduced parameters due to depthwise separability; the encoder costs $\mathcal{O}(L^2 \cdot d/h)$ in the standard implementation (sparse/linear variants are possible). The gating is linear-time. Overall, MCSA injects *complementary inductive biases* at modest cost, aiming to maintain standard accuracy while *narrowing robustness gaps* under shuffling/banning and improving stability under distribution shift.

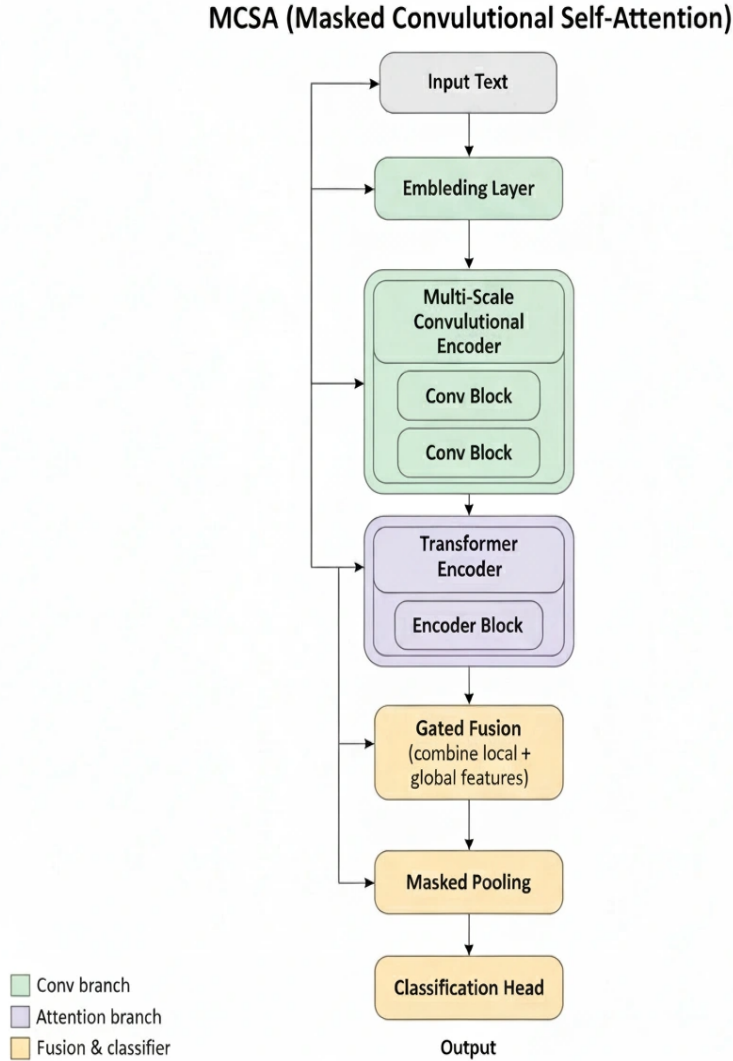


Figure 4.1: MCSA architecture

4.4 Experimental Setup

Task and data. We perform binary authorship classification (human vs. ai). After text cleaning (lowercasing, URL/email removal, numeric/symbol normalization), tokens are generated and truncated/padded to a maximum length of **200**. We deduplicate by content

hash and apply length-stratified sampling to balance human/ai across length bins. The split is approximately train/val/test = $(0.8 \times 0.9, 0.8 \times 0.1, 0.2)$.

Vocabulary and embeddings. A training-only vocabulary is built with `min_freq=2` and `max_vocab=50,000`, including `<PAD>` and `<UNK>`. We use **GloVe 6B 100d** when available; OOVs are randomly initialized and the PAD embedding is zeroed. Sequences longer than 200 are truncated and shorter ones are padded.

Sanitization (anti-shortcut). To suppress lexical shortcuts, we adopt two components: (i) **Top- K n -gram banning**: TF-IDF plus χ^2 on the training text selects the most discriminative n -grams (1-2-gram), with $K = 5000$. During encoding, unigram hits belonging to this set are mapped to `<UNK>`. (ii) **Word dropout**: with probability **0.20**, non-PAD tokens in training are replaced by `<UNK>`. These sanitization steps are shared across models for fair comparison.

Two protocols. **Protocol A (A-all)**: sanitization is applied to train/val/test (word dropout only for training; validation/test apply top- K banning). **Protocol B (B-train-only)**: sanitization is applied to training only; validation/test remain *clean* (no banning, no dropout). Protocol B probes stability under distribution shift from sanitized training to clean inputs.

Models. We compare **CNN**, **Self-Attention**, and our **MCSA**: CNN follows a TextCNN-style setup with kernels [3,4,5], 64 channels, dropout keep=0.6. Self-Attention uses a standard Transformer encoder (layers $N=2$, heads $h=4$). MCSA employs multi-scale depthwise-separable conv (kernels {3,5,7}, per-branch $c=128$) $\rightarrow$ linear projection to $d=256$ $\rightarrow$ Transformer encoder ($N=2$, $h=4$) $\rightarrow$ position-wise gated fusion $\rightarrow$ masked mean/max pooling $\rightarrow$ classifier. All models share the same embeddings, vocabulary, encoding, and loss.

Optimization and hyperparameters. Batch size is **64**; we train up to **5** epochs (many comparisons reported with **1** epoch to standardize the experimental window). We use Adam (lr= 10^{-3} , weight decay= 10^{-4}), ReduceLROnPlateau (patience=1, factor=0.5), and gradient clipping at 1.0. Class-weighted cross-entropy is used when needed. Seeds are fixed and CuDNN is set to deterministic mode.

Metrics and robustness notation. We report **Macro-F1** and **ROC-AUC**. Robustness is quantified by:

$$\text{Gap}_A = F1_{\text{std}} - F1_{\text{shuffle}}, \quad \text{Gap}_A^{\text{ban}} = F1_{\text{std}} - F1_{\text{topK}}$$

for Protocol A, and by the relative drop under distribution shift:

$$\text{Drop}_B = F1_{\text{clean}} - F1_{\text{clean+shuffle}}.$$

Sanity checks (random labels, TF-IDF+LR) and “validation under additional top- K banning” are included as diagnostics when applicable.

4.5 Results and Analysis

We analyze three dimensions: (i) **standard accuracy** (Protocol A), (ii) **robustness to word-order shuffling** (Protocol A, shuffled validation), and (iii) **stability under distribution shift** (Protocol B, train-ban $\rightarrow$ test-clean). Self-Attention achieves the highest standard accuracy; CNN is most stable under shuffling; **MCSA** strikes a favorable balance, remaining close to Self-Attention on standard accuracy while narrowing robustness gaps toward CNN under both shuffling and distribution shift.

Table 4.1: Protocol-A/B comparison (Macro-F1). Relative Drop is computed on Protocol-B validation: $F1_{\text{clean}} - F1_{\text{clean+shuffle}}$. Best per column in **bold**.

Model	A/Std Test	A/Shuffle Val	B/Clean Val	B/Clean Test	B/Clean+Shuf Val	B/Rel. Drop
CNN	0.9500	0.8563	0.9431	0.9443	0.8563	0.0868
Self-Attention	0.9781	0.7656	0.9274	0.9280	0.7175	0.2099
MCSA (ours)	0.9749	0.7950	0.9516	0.9523	0.7937	0.1579

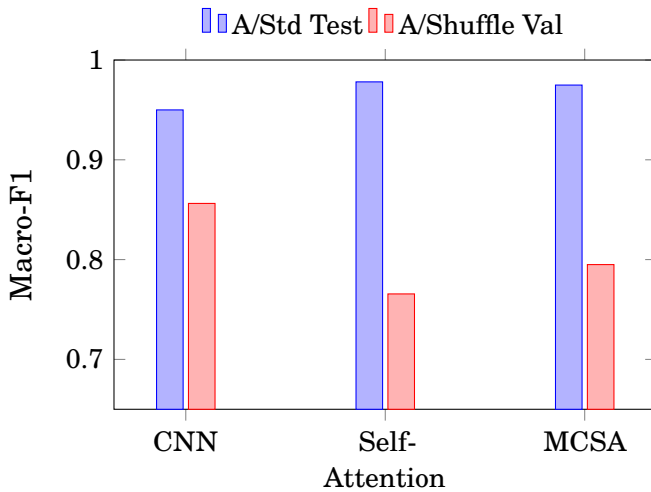


Figure 4.2: Protocol-A: Std Test vs. Shuffled Val (Macro-F1)

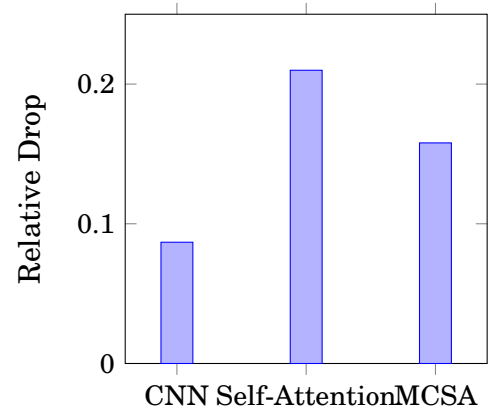


Figure 4.3: Protocol-B: Relative Drop (Val)

Takeaways. (i) Self-Attention delivers the strongest standard accuracy; (ii) CNN provides the best invariance to word-order perturbations; (iii) **MCSA** offers a better accuracy–robustness trade-off, staying competitive on standard metrics while substantially reducing robustness gaps relative to pure self-attention and approaching CNN-like stability under distribution shift.

Practical Meaning of Robustness Gains. Although the absolute improvements in some robustness metrics appear moderate, they are practically meaningful in the context of AI-generated text detection. For example, under Protocol B, MCSA reduces the relative drop from 0.2099 for Self-Attention to 0.1579, indicating a substantially smaller degradation under distribution shift. Under Protocol A, MCSA also improves shuffled-input performance over Self-Attention (0.7950 vs. 0.7656), suggesting greater resilience when sequential structure is perturbed. In deployment scenarios where detectors are exposed to rewritten, perturbed, or distribution-mismatched inputs, such reductions are valuable because they improve reliability and reduce the risk of unstable predictions. Therefore, the robustness gains of MCSA should be interpreted not only as numerical improvements, but also as evidence of more consistent behaviour under stress conditions.

Implicit Ablation Analysis. The comparison between CNN, Self-Attention, and MCSA can be interpreted as a form of implicit ablation analysis. CNN-only models emphasize local features and demonstrate stronger robustness to word-order perturbations, while self-attention-only models capture global dependencies and achieve higher standard accuracy. MCSA integrates both components and achieves a more balanced accuracy–robustness trade-off, suggesting that the combination of local convolutional features and global attention contributes to improved robustness without sacrificing competitive accuracy.

Although an explicit gated-versus-ungated fusion ablation is not included due to computational constraints, the performance improvements observed with MCSA over both CNN-only and attention-only baselines provide indirect evidence that the fused design, including the gated fusion mechanism, contributes to robustness under perturbation and distribution shift settings.

Qualitative Error Analysis. A qualitative inspection of the results suggests that errors are concentrated in more challenging input conditions, particularly perturbed inputs and cases with reduced contextual support. Shuffled inputs weaken sequential structure and therefore affect models that depend more heavily on global ordering, especially Self-Attention. By contrast, CNN remains more stable because its predictions depend more strongly on local patterns. MCSA preserves part of this robustness through its convolutional component while still benefiting from attention-based global modeling. In addition, shorter or less information-dense texts are likely to remain difficult for all models because they provide fewer lexical and contextual cues for reliable attribution. These observations suggest that the remaining errors are not uniformly distributed, but are associated with specific categories of difficult inputs.

4.6 Discussion and Limitations

Accuracy–robustness trade-off. Our results show a clear pattern: Self-Attention (SA) is slightly superior on standard accuracy (Protocol A), CNN is the most stable under word-order perturbations and distribution shift (Protocol B), and **MCSA** strikes a better balance by narrowing robustness gaps while remaining competitive on standard metrics. The combination of multi-scale convolutions and position-wise gated fusion allows MCSA to buffer local perturbations and preserve global alignment.

Impact of sanitization. Unified top- K banning and word dropout effectively suppress lexical shortcuts. Under Protocol A, applying an additional top- K ban on validation barely changes F1 for all models, suggesting that when train/val/test share the same rule, banning itself is not the main separator; *shuffling* and *train-ban* $\rightarrow$ *test-clean* are. This implies that, for deployment with potential train–test mismatch, mild augmentation or input standardization is desirable.

Threats to validity and scope. (i) **Long inputs:** vanilla self-attention is quadratic in sequence length, which can make inference slower and more memory-intensive for long documents. In deployment settings such as essay screening, report analysis, or large-scale content moderation, this may reduce throughput and increase computational cost, potentially limiting real-time applicability. Sparse/linear attention or chunked inference could be substituted to improve scalability.

(ii) **Vocabulary and K :** the choice of `max_vocab` and top- K controls sanitization strength; extreme values may under- or over-suppress useful cues.

(iii) **Data bias:** residual domain/model signatures may persist; cross-domain and temporal evaluations are needed.

(iv) **Adversarial paraphrase/mixed languages:** substantial rephrasing or code-switching can still confuse detectors by weakening the lexical and structural cues used for attribution. In real-world deployment, this means that users may intentionally rewrite AI-generated text to evade detection, increasing the risk of false negatives and reducing trust in automated screening systems. MCSA mitigates but does not eliminate this risk.

(v) **Ethics and usage boundaries:** AI-text detection bears both false positives and false negatives; detectors should be combined with watermarking, provenance metadata, and human review, and not be the sole decision-maker.

Prioritized Future Work. Future work can be prioritized according to both feasibility and expected impact.

High feasibility, high impact: the most immediate next step is to extend evaluation to cross-domain, short-text, and adversarial paraphrasing settings, since these directly

test whether the robustness gains observed in this thesis transfer to more realistic deployment scenarios.

Moderate feasibility, high impact: a second priority is to improve scalability and stability for longer inputs by introducing sparse-attention, linear-attention, or chunked-inference variants. This would make the approach more suitable for deployment in long-document screening and moderation settings.

Moderate feasibility, moderate impact: another useful direction is to perform component-level ablation studies and explore alternative hybrid backbones such as Conformer- or ConvBERT-style variants, which would further clarify the contribution of local–global fusion.

Lower feasibility, potentially high long-term impact: more ambitious directions include multilingual generalization, compression/distillation for cost-aware deployment, and multi-signal fusion with watermarking or provenance signals. These extensions could substantially improve practical applicability, but require broader datasets, additional engineering effort, and more complex evaluation protocols.

4.7 Summary

We proposed **MCSA**, a hybrid architecture that combines multi-scale convolutions for robust local cues with a self-attention encoder for global consistency, fused via position-wise gating. Under the dual protocols aligned with Chapter 3, MCSA remains close to Self-Attention on standard accuracy while substantially *reducing robustness gaps* under shuffling and distribution shift, thereby offering a stronger accuracy–robustness trade-off. These findings support the hypothesis of local–global complementarity and provide a practical starting point for cross-domain and cost-sensitive deployment in AI-text detection.

CONCLUSION & FUTURE WORK

5.1 Conclusion

This thesis developed a rigorous and reproducible framework for evaluating AI-generated content (AIGC) detection and advanced it with a new hybrid architecture. We began by establishing a unified, apples-to-apples comparison across six representative baselines-**CNN**, **RNN**, **LSTM**, **Attention-LSTM**, **Self-Attention**, and **RCNN**-under standardized preprocessing, vocabulary handling, and *sanitization* steps that suppress lexical shortcuts (top- K banning and word dropout). Two complementary evaluation regimes were defined: **Protocol A** (sanitization on train/val/test) and **Protocol B** (sanitization on training only; validation/test clean), enabling both controlled testing and robustness under train-test mismatch.

Baseline results revealed a consistent pattern: Self-Attention attained the highest standard accuracy, while CNN exhibited the strongest invariance to word-order perturbations and smaller degradation under distribution shift. These complementary strengths motivate a hybrid approach. Accordingly, Chapter 4 introduced the **Masked Convolutional Self-Attention (MCSA)** model, which fuses multi-scale convolutions for robust local cues with a self-attention encoder for global dependencies, combined via position-wise gated fusion and masked pooling.

Empirically, **MCSA** remained competitive with Self-Attention on standard metrics while *narrowing robustness gaps* toward CNN under shuffling and train-ban $\rightarrow$ test-clean shift. In other words, MCSA provides a stronger *accuracy-robustness trade-off* than either component family alone. Together with the unified protocols and diagnostics (including shuffled-order tests and top- K stress), our findings show that reporting standard accuracy alone is insufficient for AIGC detection; robustness to lexical cue removal, word-order

perturbations, and distribution shift is essential for real-world reliability.

Contributions.

- **Unified, reproducible evaluation protocol.** We standardize preprocessing, sanitization, training, and reporting (Macro-F1, ROC-AUC, robustness gaps/drops) for six architectures under Protocols A/B.
- **Comprehensive baseline study.** We clarify how architectural inductive biases (local n -gram vs. global dependency modeling) shape behavior under shortcut suppression and shuffling.
- **A new hybrid detector (MCSA).** We propose MCSA and demonstrate that it achieves a more favorable accuracy–robustness trade-off: close to Self-Attention on standard accuracy while approaching CNN-like stability under perturbations and distribution shift.

5.2 Future Work

Building on this foundation, several directions are promising:

- **Cross-generator and cross-domain generalization.** Expand evaluation to broader families of LLMs (unseen architectures, domain-specialized variants) and to out-of-domain corpora; adopt rolling/temporal splits to track drift.
- **Robustness beyond shuffling.** Systematically probe paraphrase/back-translation, synonym and style transfer, code-switching, and obfuscation; incorporate contrastive or invariance objectives.
- **Architectural extensions.** Replace vanilla attention with sparse/linear attention, Conformer/ConvBERT-style conv-attn hybrids, or low-rank adapters; explore multi-branch gating variants.
- **Efficiency and deployment.** Apply knowledge distillation, quantization, and pruning for real-time settings; study batchless/streaming inference and memory-bound optimizations.
- **Calibration and decision policy.** Improve confidence calibration, threshold tuning under class/shift imbalance, and cost-sensitive operating points for educational and media platforms.
- **Integration with provenance signals.** Combine classification with watermarking and cryptographic/provenance metadata; develop principled fusion and conflict resolution under partial evidence.

- **Interpretability and governance.** Provide token-/span-level rationales (attribution maps, counterfactuals), and document ethical boundaries and failure modes to reduce over-reliance on a single signal.

Pursuing these directions will help future AIGC detectors achieve stronger generalization, robustness, transparency, and deployability across diverse and evolving real-world environments.

BIBLIOGRAPHY

- [1] Q. Kang, E. J. Chen, Z.-C. Li, H.-B. Luo, and Y. Liu, “Attention-based lstm predictive model for the attitude and position of shield machine in tunneling,” *Underground Space*, vol. 13, pp. 335–350, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2467967423000880>
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [3] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” 2020.
- [4] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, “A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt,” 2023. [Online]. Available: <https://arxiv.org/abs/2303.04226>
- [5] C. Chen, J. Fu, and L. Lyu, “A pathway towards responsible ai generated content,” 2023. [Online]. Available: <https://arxiv.org/abs/2303.01325>
- [6] M. R. DeLisi, “Why is chatgpt making waves in the ai market?” Dec 2022. [Online]. Available: <https://www.gartner.com/en/newsroom/press-releases/2022-12-08-why-is-chatgpt-making-waves-in-the-ai-market>
- [7] T. Foltýnek, N. Meuschke, and B. Gipp, “Academic plagiarism detection,” *ACM Computing Surveys*, vol. 52, no. 6, pp. 1–42, 2020.
- [8] A. Boutadjine, F. Harrag, and K. Shaalan, “Human vs. machine: A comparative study on the detection of ai-generated content,” *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 24, no. 2, Feb. 2025. [Online]. Available: <https://doi.org/10.1145/3708889>

- [9] D. GhiurfeŃu and D. E. Popescu, “Distinguishing reality from ai: Approaches for detecting synthetic content,” *Computers*, vol. 14, no. 1, 2025. [Online]. Available: <https://www.mdpi.com/2073-431X/14/1/1>
- [10] P. Wang, L. Li, K. Ren, B. Jiang, D. Zhang, and X. Qiu, “Seqxgpt: Sentence-level ai-generated text detection,” 2023. [Online]. Available: <https://arxiv.org/abs/2310.08903>
- [11] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, “Recurrent neural network based language model.” in *Interspeech*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.
- [12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2019. [Online]. Available: <https://arxiv.org/abs/1810.04805>
- [13] A. Liu, L. Pan, Y. Lu, J. Li, X. Hu, X. Zhang, L. Wen, I. King, H. Xiong, and P. S. Yu, “A survey of text watermarking in the era of large language models,” 2024.
- [14] X. Yin, Z. Liu, D. Liu, and X. Ren, “A novel cnn-based bi-lstm parallel model with attention mechanism for human activity recognition with noisy data,” *Scientific Reports*, vol. 12, no. 1, p. 7878, May 2022. [Online]. Available: <https://doi.org/10.1038/s41598-022-11880-8>
- [15] Y. Zhang, Y. Ma, J. Liu, X. Liu, X. Wang, and W. Lu, “Detection vs. anti-detection: Is text generated by ai detectable?” in *Wisdom, Well-Being, Win-Win*, I. Sserwanga, H. Joho, J. Ma, P. Hansen, D. Wu, M. Koizumi, and A. J. Gilliland, Eds. Cham: Springer Nature Switzerland, 2024, pp. 209–222.
- [16] J. Wu, W. Gan, Z. Chen, S. Wan, and H. Lin, “Ai-generated content (aigc): A survey,” 2023. [Online]. Available: <https://arxiv.org/abs/2304.06632>
- [17] J. P. Wahle, T. Ruas, F. Kirstein, and B. Gipp, “How large language models are transforming machine-paraphrased plagiarism,” 2022.
- [18] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, and Y. Wu, “How close is chatgpt to human experts? comparison corpus, evaluation, and detection,” 2023.
- [19] D. Ippolito, D. Duckworth, C. Callison-Burch, and D. Eck, “Automatic detection of generated text is easiest when humans are fooled,” 2020. [Online]. Available: <https://arxiv.org/abs/1911.00650>
- [20] G. Jawahar, B. Sagot, and D. Seddah, “What does BERT learn about the structure of language?” in *ACL 2019 - 57th Annual Meeting of the Association*

- for Computational Linguistics*, Florence, Italy, Jul. 2019. [Online]. Available: <https://inria.hal.science/hal-02131630>
- [21] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, “Defending against neural fake news,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/3e9f0fc9b2f89e043bc6233994dfcf76-Paper.pdf
- [22] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn, “DetectGPT: Zero-shot machine-generated text detection using probability curvature,” in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 24 950–24 962. [Online]. Available: <https://proceedings.mlr.press/v202/mitchell23a.html>
- [23] W. Liang, M. Yuksekgonul, Y. Mao, E. Wu, and J. Zou, “Gpt detectors are biased against non-native english writers,” *Patterns*, vol. 4, no. 7, Jul 2023. [Online]. Available: <https://doi.org/10.1016/j.patter.2023.100779>
- [24] A. Uchendu, Z. Ma, T. Le, R. Zhang, and D. Lee, “Turingbench: A benchmark environment for turing test in the age of neural text generation,” 2021. [Online]. Available: <https://arxiv.org/abs/2109.13296>
- [25] S. Gehrmann, H. Strobelt, and A. M. Rush, “Gltr: Statistical detection and visualization of generated text,” 2019. [Online]. Available: <https://arxiv.org/abs/1906.04043>
- [26] A. Uchendu, L. Thai, S. Kai, and L. Dongwon, “Authorship attribution for neural text generation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, W. Bonnie, C. Trevor, H. Yulan, and L. Yang, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 8384–8395. [Online]. Available: <https://aclanthology.org/2020.emnlp-main.673/>
- [27] I. Solaiman, M. Brundage, J. Clark, A. Askeel, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, M. McCain, A. Newhouse, J. Blazakis, K. McGuffie, and J. Wang, “Release strategies and the social impacts of language models,” 2019. [Online]. Available: <https://arxiv.org/abs/1908.09203>
- [28] E. Tian, “Gptzero: Ai-generated text detection tool,” <https://gptzero.me>, 2023, accessed July 2025.

- [29] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, “A watermark for large language models,” in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 17 061–17 084. [Online]. Available: <https://proceedings.mlr.press/v202/kirchenbauer23a.html>
- [30] R. Zhang, S. S. Hussain, P. Neekhara, and F. Koushanfar, “Remark-llm: A robust and efficient watermarking framework for generative large language models,” 2024.
- [31] J. Weizenbaum, “Eliza-a computer program for the study of natural language communication between man and machine,” *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.
- [32] T. Winograd, “Procedures as a representation for data in a computer program for understanding natural language,” Tech. Rep., 1971.
- [33] M. A. Covington, “Building natural language generation systems,” *Language*, vol. 77, no. 3, pp. 611–612, 2001.
- [34] R. Dale, H. Moisl, and H. Somers, *Handbook of natural language processing*. CRC press, 2000.
- [35] S. F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” *Computer Speech & Language*, vol. 13, no. 4, pp. 359–394, 1999.
- [36] F. Jelinek, *Statistical methods for speech recognition*. MIT press, 1998.
- [37] P. Koehn, F. J. Och, and D. Marcu, “Statistical phrase-based translation,” 2003.
- [38] F. J. Och and H. Ney, “The alignment template approach to statistical machine translation,” *Computational linguistics*, vol. 30, no. 4, pp. 417–449, 2004.
- [39] A. Gatt and E. Reiter, “Simplenlg: A realisation engine for practical applications,” in *Proceedings of the 12th European workshop on natural language generation (ENLG 2009)*, 2009, pp. 90–93.
- [40] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: <https://doi-org.ezproxy.lib.uts.edu.au/10.1162/neco.1997.9.8.1735>
- [41] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” 2014. [Online]. Available: <https://arxiv.org/abs/1406.1078>

-
- [42] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Advances in neural information processing systems*, vol. 27, 2014.
 - [43] M. Sundermeyer, R. Schlüter, and H. Ney, “Lstm neural networks for language modeling,” in *Interspeech*, vol. 2012, 2012, pp. 194–197.
 - [44] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” 2016. [Online]. Available: <https://arxiv.org/abs/1409.0473>
 - [45] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” 2015. [Online]. Available: <https://arxiv.org/abs/1508.04025>
 - [46] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
 - [47] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.
 - [48] T. Kumarage, G. Agrawal, P. Sheth, R. Moraffah, A. Chadha, J. Garland, and H. Liu, “A survey of ai-generated text forensic systems: Detection, attribution, and characterization,” 2024.
 - [49] E. Stamatatos, “A survey of modern authorship attribution methods,” *Journal of the American Society for information Science and Technology*, vol. 60, no. 3, pp. 538–556, 2009.
 - [50] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” 2019. [Online]. Available: <https://arxiv.org/abs/1907.11692>
 - [51] P. He, X. Liu, J. Gao, and W. Chen, “Deberta: Decoding-enhanced bert with disentangled attention,” 2021. [Online]. Available: <https://arxiv.org/abs/2006.03654>
 - [52] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” *Advances in neural information processing systems*, vol. 32, 2019.
 - [53] G. Bao, Y. Zhao, Z. Teng, L. Yang, and Y. Zhang, “Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature,” 2024. [Online]. Available: <https://arxiv.org/abs/2310.05130>

- [54] Z. Guo and S. Yu, “Authentigtpt: Detecting machine-generated text via black-box language models denoising,” 2023. [Online]. Available: <https://arxiv.org/abs/2311.07700>
- [55] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature machine intelligence*, vol. 1, no. 5, pp. 206–215, 2019.
- [56] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, “On the opportunities and risks of foundation models,” 2022. [Online]. Available: <https://arxiv.org/abs/2108.07258>
- [57] Ahmadreza, “Human vs ai generated dataset,” 2023. [Online]. Available: <https://huggingface.co/datasets/ahmadreza13/human-vs-Ai-generated-dataset>
- [58] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [59] R. M. Schmidt, “Recurrent neural networks (rnns): A gentle introduction and overview,” 2019. [Online]. Available: <https://arxiv.org/abs/1912.05911>
- [60] S. Lai, L. Xu, K. Liu, and J. Zhao, “Recurrent convolutional neural networks for text classification,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.
- [61] A. W. Yu, D. Dohan, M. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, “Qanet: Combining local convolution with global self-attention for

- reading comprehension,” *CoRR*, vol. abs/1804.09541, 2018. [Online]. Available: <http://arxiv.org/abs/1804.09541>
- [62] F. Wu, A. Fan, A. Baevski, Y. N. Dauphin, and M. Auli, “Pay less attention with lightweight and dynamic convolutions,” 2019. [Online]. Available: <https://arxiv.org/abs/1901.10430>
- [63] Z. Jiang, W. Yu, D. Zhou, Y. Chen, J. Feng, and S. Yan, “Convbert: Improving bert with span-based dynamic convolution,” 2021. [Online]. Available: <https://arxiv.org/abs/2008.02496>
- [64] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The long-document transformer,” 2020. [Online]. Available: <https://arxiv.org/abs/2004.05150>
- [65] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented transformer for speech recognition,” 2020. [Online]. Available: <https://arxiv.org/abs/2005.08100>
- [66] Z. Dai, H. Liu, Q. V. Le, and M. Tan, “Coatnet: Marrying convolution and attention for all data sizes,” 2021. [Online]. Available: <https://arxiv.org/abs/2106.04803>
- [67] S. d’Ascoli, H. Touvron, M. L. Leavitt, A. S. Morcos, G. Biroli, and L. Sagun, “Convit: improving vision transformers with soft convolutional inductive biases*,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2022, no. 11, p. 114005, Nov. 2022. [Online]. Available: <http://dx.doi.org/10.1088/1742-5468/ac9830>