Back to Basics, Powered by Data: Character-Level Transformers for Robust Chinese Machine-Generated Text Detection

Shengjie Xu[1,2,3] and Zhenghao Gao[4,3] and Rujie Dai[1,3]
Zijing Li[1,3] and Xiaoyang Xie[5] and Zehua Lyu[1,*]
[1]School of Software Engineering, Huazhong University of Science and Technology
[2]School of Biomedical Engineering, Tsinghua Medicine, Tsinghua University
[3]Wuhan Ripple Effect Technology Co., Ltd.
[4]School of Artificial Intelligence and Automation,
Huazhong University of Science and Technology
[5]School of Future Technology, Huazhong University of Science and Technology
[*]Corresponding author: lvzehua@hust.edu.cn

Abstract

Machine-generated text detection remains difficult in Chinese, where systematic benchmarks and large paired corpora are still scarce. We revisit this problem with a compact 4-layer character-level Transformer, trained on a 363.4-million-character Chinese detection corpus—to our knowledge, the largest AI–human paired Chinese resource for this task. The resulting **data-centric** pipeline attains strong scores on Chinese benchmarks and, for the independently trained English variant, state-of-the-art RAID leaderboard numbers that remain stable under 11 RAID adversarial attack types. Ablations expose a "double-edged sword" role for punctuation: it eases optimization yet encourages overfitting to surface cues; symbol cleaning shifts the model toward deeper syntactic and discourse patterns and improves generalization. Minimal architecture plus rich supervision thus yields compact detectors suitable as a document-level **authorship authenticity** screen before downstream Chinese text processing; on a laptop RTX 3070 Ti we measure 5.90×10^5 tokens/s at under 100 MB VRAM. We present **GiantMelon-ZH** (Chinese) and **GiantMelon-EN** (English for RAID)—same backbone, disjoint training corpora, transparent data descriptions.

1 Introduction

Large language models have intensified the need for reliable machine-generated text detection, yet Chinese remains under-served relative to English. Many detectors that excel on in-distribution English data degrade under adversarial perturbations or domain shift.

The practical cost of that gap is not confined to leaderboard scores. Multi-source Chinese pipelines—news aggregation, encyclopedic curation, and downstream extraction or risk analysis—typically treat incoming documents as human-authored records. Fluently generated text can enter those pipelines without the telltale emptiness of placeholders or the broken character sequences that length and frozen-language-model perplexity filters already catch. Once accepted as authentic, such text contaminates whatever runs next. We therefore treat detection as **authorship authenticity screening**: a compact, document-level gate that should sit **in front of subsequent Chinese text processing**, so later stages operate only on text that has been screened (or explicitly flagged). The gate scores generation provenance—human-authored versus machine-written—not communicative sincerity or speaker intent.

The Chinese task is consequently **human-authored versus AI-written** text. "AI-rewritten" names our construction of the positive class, not a claim that lightly edited human prose is machine-generated. Each source document is abridged to a topic seed of at most 50 words; the generator then plans new details and expands them (Appendix D). The resulting wording is model-authored rather than rearranged from the source. This controlled procedure keeps the two classes on similar topics, blocking a trivial topic cue, and differs from RAID-style prompt completion.

We **return to basics** with two compact Transformer detectors—**GiantMelon-ZH** (character-level, Chinese) and **GiantMelon-EN** (word-level, English for RAID)—built on the same four-layer backbone (∼2.1M parameters) but trained on **disjoint** corpora without cross-lingual weight sharing. Instead of architectural scale, we invest in data construction and symbol cleaning: a 363.4-million-character Chinese corpus (the largest AI–human paired Chinese detection resource we are aware of) and a GPT-4o English parallel corpus for RAID fine-tuning. GiantMelon-ZH is accurate on the paired Chinese corpus, on HC3-Chinese, and on MultiSocial-Chinese. **Adversarial robustness evidence is for GiantMelon-EN on RAID**; we do

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 5879–5903
November 6-10, 2026 ©2026 Association for Computational Linguistics

not claim a Chinese adversarial benchmark. This design supports the view that **scaling curated data improves per-parameter effectiveness** of compact detectors (Devlin et al., 2019; Hoffmann et al., 2022), while inference stays practical on consumer hardware (Table 4).

Systematic ablations reveal a "double-edged sword" for symbolic cues: retaining punctuation and Western symbols lowers test accuracy from 98.1% to 95.2% (Table 6), consistent with overfitting to formatting; symbol cleaning redirects the model toward deeper linguistic structure. We further validate unmasked mean pooling, importance-weighted aggregation, and bidirectional attention (Table 12). Gradient-based character importance is an **exploratory diagnostic** of which characters move a prediction, not a causal "AI fingerprint"; domain and style remain confounds (Section 4.6).

Our principal contributions include: (1) **GiantMelon-ZH** and **GiantMelon-EN**, two compact variants achieving **SOTA on RAID (EN)** and strong Chinese benchmark scores (ZH) with transparent training data descriptions, intended as a front-end authorship screen for Chinese text pipelines; (2) a **363.4-million-character Chinese corpus** and its **GPT-4o English parallel translation** for reproducible data-centric MGTD research; (3) **systematic ablations** on symbol cleaning, pooling, and aggregation, plus deployment benchmarks showing that small specialized detectors remain viable alongside large PLM-based systems.

2 Related Work

2.1 Detection Methods

Early detectors exploit statistical features or probability curvature (e.g., GLTR (Gehrmann et al., 2019), DetectGPT (Mitchell et al., 2023)). Fast-DetectGPT (Bao et al., 2024) accelerates this paradigm via conditional probability curvature, substantially reducing inference cost, yet all curvature-based methods presuppose white-box access to the generator and typically reach only 60–75% accuracy against modern obfuscated outputs. Binoculars (Hans et al., 2024) extends the zero-shot family by cross-referencing perplexity scores from two pre-trained LMs, attaining strong performance on English benchmarks without labeled training data, though it inherits the heavy-inference limitation at deployment scale.

A complementary line embeds covert signals directly in the generated tokens: Kirchenbauer et al. (2023) partitions the vocabulary into a green/red list at each generation step, enabling provably detectable watermarks under a hypothesis-testing framework. Watermarking, however, requires generator cooperation and provides no protection against non-watermarked or paraphrased outputs—precisely the adversarial regime central to RAID.

Classifier-based approaches fine-tune large pre-trained encoders (Solaiman et al., 2019; Liu et al., 2019) or cascade multiple models (Verma et al., 2024); adversarial training (RADAR (Hu et al., 2023)) still assumes English-centric resources and heavy inference. We instead pair a lightweight character-level encoder with symbol cleaning.

2.2 Evaluation Benchmarks

TuringBench (Uchendu et al., 2021) constructed an early multi-model benchmark; MULTITuDE (Macko et al., 2023) extends to 11 languages, revealing that existing detectors achieve only 54–72% accuracy in multilingual scenarios. The RAID benchmark (Dugan et al., 2024) encompasses ∼6 million samples across 11 generator models, 8 domains, and 11 adversarial attacks, systematically testing robustness under cross-domain transfer and diverse attacks. MAGE (Li et al., 2024) focuses on real-world scenarios with 27 LLMs, emphasizing natural distribution shifts. At the framework level, MGTBench (He et al., 2024) and IMGTB (Spiegel and Macko, 2024) unify evaluation pipelines. GiantMelon-EN achieves near-perfect ROC-AUC on RAID, outperforming commercial detectors on the leaderboard and open-source baselines.

2.3 Chinese Detection

Systematic investigation of Chinese scenarios remains scarce. LLM-Detector (Wang et al., 2024) uses instruction tuning but depends on large-scale resources. HC3-Chinese (Guo et al., 2023) and MultiSocial-Chinese (Macko et al., 2025) are among the few public Chinese sets, yet target short social or QA text. EnsemJudge (Wang et al., 2026) ensembles diverse judges at a high inference cost. XLM-R (Conneau et al., 2020) offers cross-lingual transfer but is not character-native. We instead train a compact backbone on a large paired Chinese corpus and a GPT-4o English parallel for RAID, with no cross-lingual weight sharing.

Data Source	Human (M)	AI (M)
Chinese Wikipedia	12.0	4.3
THUCNews	103.8	243.3
Total	**115.8**	**247.6**
Grand Total	**363.4**	

Table 1: Dataset scale by source (millions of Chinese characters).

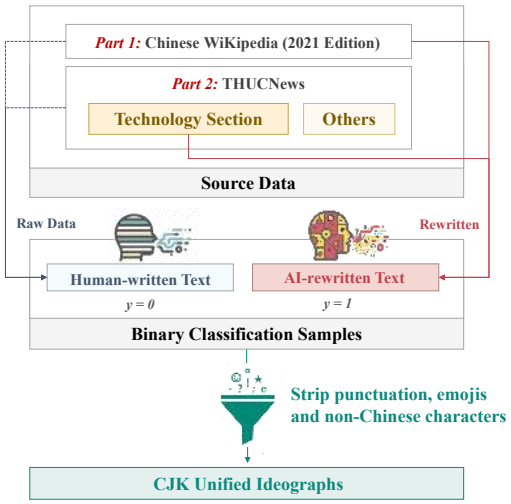


Figure 1: Dataset construction pipeline: from source data (Chinese Wikipedia and THUCNews) to binary classification samples (human-written vs. AI-rewritten), with preprocessing to retain only CJK Unified Ideographs.

3 Method

3.1 Data and Preprocessing

Corpus sources and annotation: Our Chinese detection corpus comprises human-authored texts and AI-written texts. As illustrated in Figure 1, we collect source data from two primary sources: (1) the 2021 edition of Chinese Wikipedia, and (2) the THUCNews news corpus, which is further divided into the Technology Section and other sections. The pipeline then branches into two distinct paths: **human-written text** ($y = 0$), which directly originates from the raw data (Chinese Wikipedia and the non-Technology sections of THUCNews), and **AI-written text** ($y = 1$), constructed by a three-stage rewriting process applied to the source: abridge to at most 50 words, plan details, then expand (verbatim prompts in Appendix D). After abridging, only a topic seed remains; the model authors the subsequent details and wording rather than rearranging the source. Specifically, the AI texts are divided into two parts: the 2021 Wikipedia-**Claude 4.5 Opus** version and the THUCNews Technology Section **Grok 4.1 Fast** version. These two types of text are combined to form binary classification training samples after chunking. The label records **who authored the semantic content**, not whether an LLM later touched the string.

Dataset scale: After preprocessing to retain only Chinese characters, our corpus contains a total of 363.4 million characters. Table 1 presents the detailed breakdown by data source and label type. We release the Chinese detection texts, the Wiki–Claude pairs, and a paired Chinese–English Wikipedia slice at `https://huggingface.co/datasets/layce/giantmelon-zh`.

Symbol cleaning: As shown in the preprocessing stage of Figure 1, prior to feeding text into the character-level tokenizer, we uniformly remove Chinese punctuation, Western punctuation, emojis, and other non-Chinese symbols, retaining only CJK Unified Ideographs. This design deliberately weakens non-semantic patterns that may exist in punctuation and formatting, compelling the model to rely more on syntactic and discourse-level generation patterns.

3.2 English Parallel Corpus for GiantMelon-EN

RAID provides an **English-only** official training set; it does not include Chinese data. To transfer our Chinese data-construction pipeline to English without generating new labels, we create a **segment-aligned parallel corpus**: for each human/AI pair in the GiantMelon-ZH sources (Wikipedia–Claude and THUCNews-Technology–Grok), we translate both sides with **GPT-4o** into English, preserving binary labels (y=0 human, y=1 AI). Translation applies identically to both classes and changes language, not content authorship: a translated human document remains human-written, and a translated AI-written document remains AI-written. Otherwise every machine-translated human document would have to be relabeled as AI-written. The translation prompt, length-ratio filter, and quality checks are in Appendix D. Translation is used only for language transfer, not for synthesizing additional machine text. The English text undergoes the same symbol-stripping policy as Chinese (all punctuation and non-alphanumeric symbols removed) and is tokenized with a natural word dictionary (any learner-grade English dictionary suffices; we use a licensed dictionary that we do not redistribute).

Component	Description
Parallel sources	Wiki (Claude) and THUCNews Tech (Grok), as in Table 1
Translation	GPT-4o, segment-aligned; labels unchanged
Tokenization	Dictionary word tokens after symbol stripping
RAID fine-tuning	Official RAID train + parallel English corpus

Table 2: English training data for GiantMelon-EN (RAID). RAID benchmark data are English-only.

GiantMelon-EN is trained independently from GiantMelon-ZH on the **same four-layer backbone with no shared weights or initialization**: we first train on the parallel English corpus, then fine-tune jointly on the RAID official training set plus this translated data for leaderboard submission. We report corpus composition in Table 2 and RAID training ablations in Table 8.

3.3 Model Architecture

Figure 2 illustrates our character-level Transformer architecture. Given a Chinese long text $T = (c_1, \ldots, c_L)$, we first map each character to an index using a character-level vocabulary $\mathcal{V}$ (with <pad>=0 and <unk>=1), obtaining the index sequence $\mathbf{x} = (x_1, \ldots, x_L)$.

To handle long texts, we employ a fixed-window sliding-stride chunking strategy with window length $S = 512$ and stride $\Delta = 256$, extracting overlapping subsequences $\mathbf{x}^{(k)} = (x_b, x_{b+1}, \ldots, x_{b+S-1})$ for each chunk k.

For each chunk, we pass it through token and positional embeddings (dimension $d = 256$), sum them element-wise, and apply dropout to obtain $\mathbf{H}^{(0)} \in \mathbb{R}^{S \times d}$. The input is then processed by a Transformer encoder (Vaswani et al., 2017) with $L = 4$ layers, each containing multi-head self-attention (4 heads) and feed-forward networks (hidden dimension 512), following standard Transformer architecture with residual connections and layer normalization.

After the encoder, we apply unmasked mean pooling to aggregate token-level representations into chunk-level vectors:

$$\mathbf{z} = \frac{1}{S} \sum_{i=1}^{S} \mathbf{H}_i^{(L)} \in \mathbb{R}^d, \tag{1}$$

where padding positions are automatically suppressed through attention mechanisms. The chunk-level vector $\mathbf{z}$ is fed into a linear classifier followed by softmax to obtain chunk-level probabilities $\mathbf{p} = (p_{\text{Human}}, p_{\text{AI}})$.

For document-level prediction, we aggregate chunk-level probabilities using importance-weighted averaging (as detailed in Section 3.6), where importance scores are derived from gradient-based character sensitivity analysis of character-level contributions to the AI class prediction.

Setting	ZH	EN
Optimizer	AdamW	AdamW
Learning rate (pre-train)	10^{-4}	10^{-4}
Learning rate (RAID fine-tune)	–	10^{-5}
Batch size	16	16
Training epochs	8	8
Gradient clip (max norm)	1.0	1.0
Window / stride	512 / 256	512 / 256
HC3 short-text fine-tune	10^{-6}	–

Table 3: Training hyperparameters for GiantMelon-ZH (ZH) and GiantMelon-EN (EN). HC3-FT is a separate ZH branch (Section 4.4).

3.4 Training

We employ cross-entropy loss for binary classification and optimize using AdamW (Loshchilov and Hutter, 2019) with gradient clipping (max norm 1.0). Following BERT-like training practices (Devlin et al., 2019), we use stride Δ=256 for window S=512 (50% overlap). Table 3 lists hyperparameters; GiantMelon-ZH training on the full Chinese corpus consumed one NVIDIA A100 for approximately 84 hours. GiantMelon-EN uses the same architecture and optimizer settings for pre-training on the translated corpus, then a lower learning rate for RAID fine-tuning (details in Section 4).

3.5 UnMasked Mean Pooling

We employ unmasked mean pooling to aggregate token-level representations into chunk-level features. Compared to max pooling or single-position pooling (e.g., [CLS] token in BERT (Devlin et al., 2019)), mean pooling provides greater robustness against noisy tokens and adversarial perturbations, as it averages over all contextualized token representations rather than relying on local maxima or a single position. Our ablation studies (Section 4.5) validate that unmasked mean pooling achieves superior performance compared to alternatives.

3.6 Character-Level Interpretability

To analyze which characters move a prediction, we compute gradient-based importance scores as

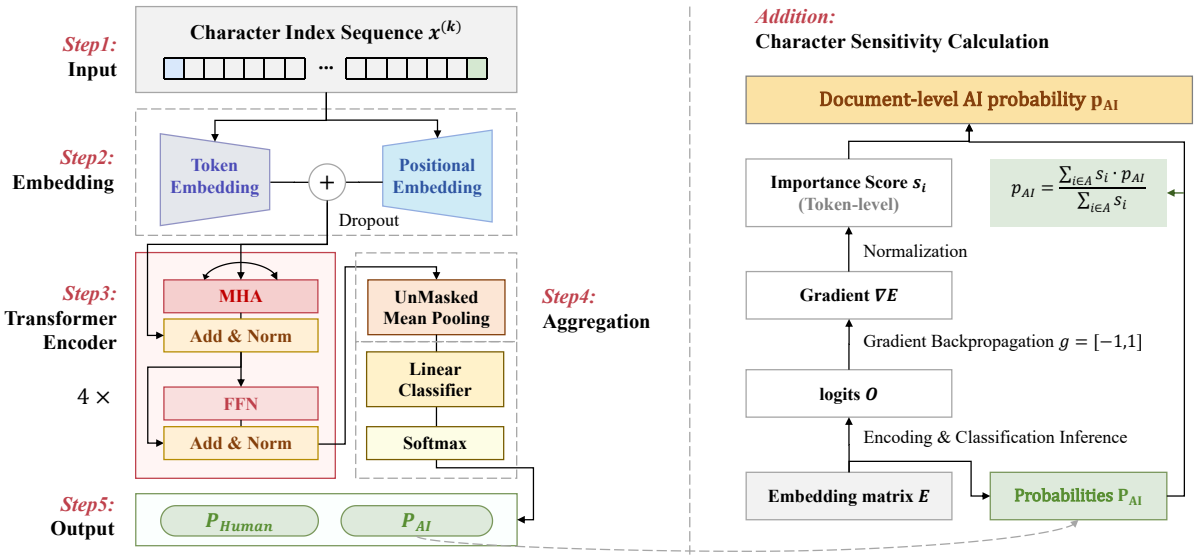


Figure 2: Model architecture: character-level Transformer encoder with unmasked mean pooling for chunk-level classification, and document-level aggregation with importance-weighted averaging.

a **post-hoc diagnostic**, not as evidence of a causal generation fingerprint. Source, domain, and style can explain part of the pattern (Section 4.6). For each chunk, we set embeddings $\mathbf{E}$ to require gradients, perform a forward pass to obtain logits $\mathbf{o}$, then backpropagate with target direction $\mathbf{g} = [-1, 1]$ (increasing AI class score, decreasing human class score), where logits are ordered as [human, AI] following our label convention (y=0 human, y=1 AI). The gradient magnitude $g_i = \|\nabla_{\mathbf{E}_i}\|_1$ for each character position i is normalized through a multi-step process: (1) shift to non-negative: $\tilde{g}_i = g_i - \min_j g_j$; (2) truncate upper bound: $\tilde{g}_i \leftarrow \min(\tilde{g}_i, 2 \cdot \bar{g})$ where $\bar{g}$ is the mean; (3) normalize to $[0, 1]$: $\hat{g}_i = \tilde{g}_i / \max_j \tilde{g}_j$; (4) logarithmic compression: $\hat{g}_i \leftarrow \hat{g}_i + 1$, $s_i = \log \hat{g}_i$, then renormalize to $[0, 1]$. This process keeps importance scores $s_i \in [0, 1]$ bounded and avoids extreme outliers while maintaining relative relationships.

For document-level prediction, we aggregate chunk-level probabilities using importance-weighted averaging:

$$\hat{p}_{\text{AI}} = \frac{\sum_{i \in \mathcal{A}} s_i \cdot p_{\text{AI}}}{\sum_{i \in \mathcal{A}} s_i}, \tag{2}$$

where $\mathcal{A}$ denotes analyzed character positions. This enables visualization of character-level contributions to AI detection, as shown in Figure 2.

4 Experiments

4.1 Experimental Setup

4.1.1 Model Variants and Datasets

We evaluate two **independently trained** detectors (no shared checkpoints): **GiantMelon-ZH** (character-level, Chinese corpus, Section 3.1) and **GiantMelon-EN** (word-level, RAID training, Section 3.2). Unless stated otherwise, "GiantMelon" in RAID tables refers to GiantMelon-EN; Chinese benchmark tables refer to GiantMelon-ZH.

Self-constructed Chinese dataset: GiantMelon-ZH is trained on 363.4 million characters with a 9:1 train/test split (fixed seed).

RAID benchmark (Dugan et al., 2024): English-only training and evaluation (∼6M samples, 11 generators, 8 domains, 11 attacks). GiantMelon-EN is fine-tuned on the official RAID training set plus our GPT-4o parallel English corpus (Table 2). We submitted predictions to the blind leaderboard under the name "GiantMelon" (historical submission ID).

HC3-Chinese (Guo et al., 2023): We report results on `open_qa` and `nlpcc_dbqa`. Short inputs use a separate **GiantMelon-ZH-HC3-FT** branch fine-tuned on HC3 with learning rate 10^{-6} so the main ZH model's zero-shot behavior is preserved.

MultiSocial-Chinese (Macko et al., 2025): Chinese social-media subset (short texts); we fine-tune a dedicated ZH branch for fair comparison with domain-adapted baselines.

Hardware	Throughput (tok/s)	Peak VRAM
RTX 3070 Ti Laptop GPU	589,932	<100 MB
Intel i7-12700H CPU	45,648	—

Table 4: Inference efficiency for GiantMelon-ZH (∼2.1M parameters, batch size 1). Benchmarks verified on consumer laptop hardware.

4.1.2 Deployment Efficiency

Table 4 reports inference throughput for GiantMelon-ZH (batch size 1, sliding windows). Training is compute-intensive (84 GPU-hours on A100), but deployment requires less than 100 MB VRAM—orders of magnitude below LLM-based detectors.

4.1.3 Evaluation Metrics

Evaluation metrics vary by dataset to align with standard evaluation protocols:

Self-constructed dataset: We employ Accuracy, ROC-AUC, and F1-score as primary evaluation metrics.

RAID benchmark: Following the official evaluation protocol, we use AUROC (Area Under the Receiver Operating Characteristic curve) as the primary metric, reporting Aggregate AUROC across all models and domains.

HC3-Chinese: Following the original HC3 evaluation protocol, we report F1-Human and F1-ChatGPT metrics, which measure detection performance for human and ChatGPT-generated texts separately.

MultiSocial-Chinese (Macko et al., 2025): Following the MultiSocial evaluation protocol, we report AUC ROC and MacroF1@5%FPR (Macro-averaged F1-score at 5% false positive rate), which is particularly suitable for imbalanced social-media detection scenarios.

Table 5 aligns datasets, metrics, and model variants. We refer to the Chinese random 9:1 partition together with the source-held-out splits in Table 7 as the **GiantMelon-ZH evaluation protocol** for follow-up work on this corpus.

4.2 Performance on Self-Constructed Dataset

With identical architecture, keeping punctuation yields 99.9% training accuracy but only 95.2% test accuracy (98.1% ROC-AUC); removing non-CJK symbols raises test accuracy to 98.1% and ROC-AUC to 99.9% (Table 6). Punctuation and formatting are easy to memorize yet hurt generalization;

Setting	Variant	Primary metric
Self-constructed ZH	GiantMelon-ZH	ACC / AUC / F1
Source-held-out ZH	GiantMelon-ZH	ACC / AUC / F1
RAID (EN, blind)	GiantMelon-EN	Aggregate AUROC
HC3-Chinese	GiantMelon-ZH-HC3-FT	F1-Human / F1-AI
MultiSocial-Chinese	GiantMelon-ZH (FT)	AUC ROC / MacroF1@5%FPR

Table 5: Evaluation matrix: dataset, model variant, and official metric. RAID adversarial robustness is reported only for GiantMelon-EN.

Preprocessing Setting	Train ACC (%)	Test ACC (%)	Test AUC (%)
Remove punct. (default)	**99.9**	**98.1**	**99.9**
Keep punct. (comparison)	**99.9**	95.2	98.1

Table 6: Model performance under different preprocessing settings.

symbol cleaning shifts capacity toward syntactic rhythm and discourse structure. We therefore apply the default "remove punctuation" setting in all later experiments, including the architectural ablations in Table 12.

A random 9:1 split may overestimate in-corpus accuracy. We therefore retrain GiantMelon-ZH with **source-held-out** splits: all News documents in train and Wiki in test, then the reverse, using the same architecture and preprocessing and preventing document chunks from crossing the split (Table 7). News→Wiki yields 97.8 ACC / 99.9 AUC / 96.5 F1; Wiki→News yields 98.2 / 99.9 / 97.2, close to the random split. Source and generator are coupled in this corpus (Wiki/Claude and Tech-News/Grok), so the test also approximates a generator shift and cannot isolate the two factors. We treat the result as evidence that the main conclusion does not depend only on a random partition, not as a strict generator-held-out or time-held-out protocol—those require cross-generator pairs and dated slices that the current corpus does not contain.

4.3 Performance on RAID Benchmark

We evaluate **GiantMelon-EN** on the official RAID leaderboard[1] (submitted as "GiantMelon"). RAID uses blind evaluation: organizers return aggregate

[1] https://raid-bench.xyz/leaderboard

Split	ACC (%)	AUC (%)	F1 (%)
Random 9:1 (paper)	98.1	99.9	97.0
Train News, test Wiki	97.8	99.9	96.5
Train Wiki, test News	98.2	99.9	97.2

Table 7: Source-held-out evaluation of GiantMelon-ZH. Source and generator are coupled; the shift is therefore combined.

Training configuration	Agg. AUROC
RAID official train only	0.996
RAID official + GPT-4o parallel corpus	1.000

Table 8: RAID training ablation for GiantMelon-EN (`attack=none`).

AUROC only, so we cannot audit per-sample errors; reported scores may be rounded at three decimals.

Table 8 isolates data contributions: RAID-only training yields 0.996 AUROC, while adding our translated parallel corpus reaches 1.000—extra curated English data, not architectural complexity, closes the remaining gap. Table 9 compares GiantMelon-EN with public detectors under `adversarial attack=none, domain=all, decoding strategy=all`. We show mainstream generators; remaining models appear in Aggregate (full table in Appendix A).

GiantMelon-EN achieves an AUROC of 1.000 on the Aggregate metric under RAID's blind evaluation protocol. Among the seven mainstream generation models shown, four reach 1.000 (chatgpt, gpt4, mistral-chat, llama-chat), and the remainder are ≥ 0.997. Compared to SpeedAI, GPTZero, e5-small-lora (Wang et al., 2022), and BERT-tiny-4M (Jiao et al., 2020), GiantMelon-EN matches or exceeds commercial and open-source detectors while using only four Transformer layers (Table 4).

These results indicate that a compact, data-scaled English detector reaches SOTA on the public RAID leaderboard. The gain from RAID-only (0.996) to full training (1.000) corroborates that curated additional English data—not deeper architectures—closes the remaining error (Table 8).

4.3.1 Robustness Under Adversarial Attacks

RAID supplies the standardized adversarial evaluation; **no authoritative Chinese adversarial MGTD benchmark is available**, so we do not read RAID numbers as Chinese robustness. Figure 3 presents GiantMelon-EN's Aggregate AUROC across 11 attack types; details are in Appendix B.

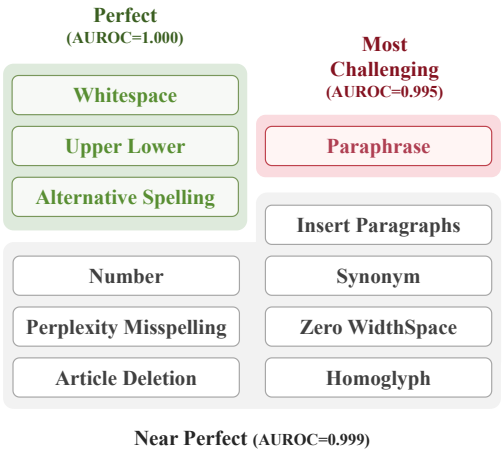


Figure 3: Categorization of Aggregate AUROC performance across adversarial attack types on the RAID benchmark.

GiantMelon-EN achieves Aggregate AUROC ≥ 0.995 on every RAID attack type (0.999 aggregated). Three attacks reach 1.000 on Aggregate (Whitespace, Upper Lower, Alternative Spelling); Paraphrase is lowest at 0.995.

Formatting versus semantic robustness. Whitespace and Upper Lower perturb surface formatting. GiantMelon-EN's symbol-stripping preprocessor removes spaces and case before tokenization, so these attacks are **system-level** effects of preprocessing rather than architectural adversarial robustness of the Transformer. Semantic attacks—Paraphrase (aggregate AUROC 0.995) and Synonym—are the meaningful model-level stress tests. Under those attacks GiantMelon-EN remains competitive with SpeedAI and well above GPTZero and BERT-tiny-4M (Appendix B).

4.4 Performance on Chinese Public Benchmarks

We evaluate **GiantMelon-ZH** on HC3-Chinese (Guo et al., 2023) and MultiSocial-Chinese (Macko et al., 2025). Because both benchmarks use shorter texts than our training corpus, we fine-tune dedicated branches: **GiantMelon-ZH-HC3-FT** (learning rate 10^{-6} on HC3) and a MultiSocial-adapted ZH model, kept separate from the main ZH checkpoint to avoid overfitting to short ChatGPT-style QA. Tables 10 and 11 report these results.

On HC3-Chinese, GiantMelon-ZH-HC3-FT matches the RoBERTa-based best detector (Solaiman et al., 2019; Liu et al., 2019) (F1-Human 98.90% and 97.90% on `open_qa` and `nlpcc_dbqa`), outperforming GLTR (Gehrmann et al., 2019). On

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**1.000**	**1.000**	**1.000**	**0.999**	**1.000**	**0.999**	**0.997**	**1.000**
SpeedAI	0.999	1.000	1.000	0.997	1.000	0.997	0.987	1.000
GPTZero	0.984	0.998	0.998	0.975	0.997	0.938	0.951	0.999
e5-small-lora	0.939	0.992	0.993	0.888	0.983	0.740	0.876	0.994
DivEye	0.936	0.994	0.901	0.864	0.943	0.899	0.926	0.975
BERT-tiny-4M	0.911	0.978	0.966	0.883	0.962	0.627	0.818	0.977
Desklib	0.924	0.996	0.939	0.858	0.994	0.685	0.850	0.998
Desklib AI Text Detector v1.01	0.949	0.996	0.957	0.918	0.996	0.654	0.863	0.998

Table 9: AUROC performance comparison of detectors on the RAID benchmark (no adversarial attack setting). SpeedAI, GPTZero, DivEye, and Desklib are commercial detectors on the RAID leaderboard. e5-small-lora (Wang et al., 2022) and BERT-tiny-4M (Jiao et al., 2020) are open-source baselines.

Dataset	Detector	F1-H	F1-AI
HC3 zh-open_qa	GiantMelon-ZH-HC3-FT	**98.90**	99.28
	RoBERTa-best	98.75	**99.33**
	GLTR	85.75	93.15
HC3 zh-nlpcc_dbqa	GiantMelon-ZH-HC3-FT	**97.90**	95.32
	RoBERTa-best	97.42	**95.42**
	GLTR	77.25	69.78

Table 10: Performance on HC3-Chinese (F1-Human / F1-ChatGPT).

Detector	AUC ROC	MacroF1 @5%FPR
GiantMelon-ZH (FT)	**0.98**	**0.89**
Llama-3-8b-MultiSocial	**0.98**	0.87
Mistral-7b-v0.1-MultiSocial	**0.98**	0.87
Aya-101-MultiSocial	0.97	0.85
XLM-RoBERTa-large-MultiSocial	0.93	–
BLOOMZ-3b-mixed-Detector	0.69	0.30
ChatGPT-Detector-RoBERTa-Chinese	0.81	0.34
Statistical methods	0.74–0.79	–

Table 11: Performance on MultiSocial-Chinese.

MultiSocial-Chinese, the fine-tuned ZH variant matches Llama-3-8b-MultiSocial and Mistral-7b-MultiSocial (AUC 0.98, MacroF1@5%FPR 0.89) and exceeds zero-shot baselines. The same compact backbone therefore transfers across Chinese domains and English RAID when the data pipeline is matched to each setting. Fine-tuning at 10^{-6} preserves symbol-cleaning invariances while adapting to short QA and social text.

4.5 Ablation Studies

To validate the design choices in Section 3, we ablate pooling (`pool`), aggregation (`agg`), and attention (`attn`) on the self-constructed test set under the default symbol-cleaning policy (Table 6). Table 12 highlights the default against mean aggregation, causal attention, and max pooling; the full 18-row grid is Appendix C.

Setting	AUC	ACC	F1
bi / unmasked_mean / weighted (def.)	**99.44**	**98.1**	97.02
bi / unmasked_mean / mean	99.42	96.95	96.99
causal / unmasked_mean / weighted	99.09	96.99	97.02
bi / max / weighted_mean	76.45	50.46	66.74

Table 12: Selected architectural ablations (%). Default: bidirectional, unmasked mean, importance-weighted aggregation. Full grid: Appendix C.

Pattern	Typical error	Freq.
Finance news	FN: human text w/ dense figures; FP: templated AI after numeral strip	High
Expert prose	FP: regular terminology → AI template	Med.
Fact-rich AI	FN: anchored generations look human	Med.
HC3 social	Mitigated by HC3-FT (lr=10^{-6})	Low

Table 13: Qualitative error patterns (ZH test set; RAID aggregate only).

`Unmasked_mean` + `weighted_mean` is best. `Max` pooling degrades sharply, consistent with Section 3.5.

4.6 Error Analysis

We analyze failure modes on the GiantMelon-ZH held-out test set and RAID blind submissions (aggregate only for RAID). Table 13 summarizes dominant error patterns.

Finance errors follow from symbol stripping (numerals removed); semantic RAID attacks remain the meaningful stress test. Preserving numeric placeholders without punctuation overfitting is future work.

4.7 Case Study

Through gradient-based importance analysis (Section 3.6), AI-written texts often concentrate importance on standardized terminology and formulaic collocations, whereas human texts more of-

ten spread importance across idiosyncratic details. These patterns are **correlational** and may partly reflect source or style; Table 13 already lists expert prose as a medium-frequency false positive when dense professional terminology resembles AI templates. Appendix F reports vignettes aligned with that table.

5 Discussion

Our results support a **data-centric** view of MGTD (Devlin et al., 2019; Hoffmann et al., 2022): curated corpora and symbol cleaning let a four-layer backbone match RAID and Chinese benchmarks (Table 12) at over 589k tokens/s on a laptop (Table 4). GiantMelon-ZH and GiantMelon-EN are separate models; the RAID-only ablation (0.996 vs. 1.000) isolates the English pipeline. Formatting RAID attacks are largely preprocessing; semantic paraphrase and finance-domain numerals (Table 13) remain open. Unmasked mean pooling is more stable than `max` (Table 12).

That efficiency is what makes a **front-end** screen feasible. Downstream Chinese NLP—extraction, quality scoring, retrieval, or risk analysis—almost always assumes that the string it receives is a human record. Length checks and frozen character-level perplexity catch empty shells and broken sequences; they do not reliably catch fluent machine-written prose that already sits inside the perplexity band of ordinary reports. GiantMelon-ZH targets that residue. The intended stack is therefore ordered: run authorship authenticity screening first, pass or flag the document, and only then invoke later processors. The screen does not certify that a speaker is sincere, that a record is factually true, or that AI-assisted editing is illegitimate. It estimates whether the document's wording is human-authored or machine-written, which is the quantity the training labels encode.

Two failure modes show why the gate cannot be unattended. Expert human prose with dense terminology can resemble AI templates (Table 13); fact-rich machine text can look human. Both argue for human review of borderline scores rather than automatic deletion. Source-held-out numbers (Table 7) remain close to the random split, but source and generator are coupled, so the protocol is not a substitute for future cross-generator pairs. On RAID, we continue to separate preprocessing-driven formatting invariance from model behavior under paraphrase.

6 Conclusion

We present GiantMelon-ZH and GiantMelon-EN, compact bilingual detectors trained on a 363.4-million-character Chinese corpus and its GPT-4o English parallel. GiantMelon-EN reaches state-of-the-art RAID results—the setting in which we use the word **robust**—while GiantMelon-ZH performs strongly on HC3 and MultiSocial with targeted domain fine-tuning. The same compact Chinese model is meant to hang at the head of subsequent text processing as a document-level authorship authenticity screen, complementary to length and perplexity filters that miss fluent machine-written text.

The Chinese label is human-authored versus AI-written content. The positive class is built by abridging a source to a topic seed and letting the generator plan and expand, not by treating light edits or RAID-style prompt completion as the same phenomenon (Appendix D). Translation into English for GiantMelon-EN does not flip those labels. Source-held-out splits on News and Wiki stay close to the random 9:1 protocol (Table 7), with the caveat that source and generator remain coupled. Formatting attacks on RAID are largely absorbed by symbol stripping; paraphrase is the stress test that still belongs to the model.

What the gate does not do matters as much as what it does. It does not score whether a speaker is sincere, whether a record is true, or whether AI-assisted drafting should be forbidden. It estimates generation provenance so that later extractors, quality checkers, and analysts are not fed fluent machine text as if it were an attested human report. Expert prose and fact-rich generations still require a human in the loop (Table 13). The compact character-level detector is already cheap enough to run first at intake.

Limitations

We do not introduce new architectural primitives; our contribution is empirical—data construction, preprocessing, and transparent reporting. The Chinese corpus uses only two generators (Claude 4.5 Opus and Grok 4.1 Fast). Source-held-out evaluation cannot separate source from generator. English parallel text introduces translationese and may not match native English RAID domains. RAID evaluation is blind and aggregate-only, limiting per-sample leakage audits and error analysis on English test data. Finance-domain errors (Table 13) show that removing numerals harms some human–AI distinctions. Formatting RAID attacks (Whitespace, Upper Lower) are largely neutralized by preprocessing and should not be interpreted as architectural adversarial robustness. Authorship screening does not measure factual truth or speaker intent, and expert human writing can still be flagged.

Ethics Statement

Our dataset is constructed from publicly available sources: the 2021 edition of Chinese Wikipedia and the THUCNews news corpus. These sources consist of encyclopedia articles and news articles that do not contain personally identifying information or offensive content. We apply symbol cleaning to remove non-Chinese characters, further reducing any potential privacy concerns. The dataset consists of general-domain text without personal identifiers, ensuring that our research does not raise privacy or ethical concerns related to personal data.

Use of generative AI. We used GPT-4o to translate paired Chinese training segments into English for GiantMelon-EN (Section 3.2); labels were not generated by translation. Non-native English-speaking authors also used AI assistants (e.g., ChatGPT and Claude) for **manuscript** translation and polishing only; research design, experiments, and conclusions are the authors' own.

While machine-generated text detection can help maintain content authenticity in multi-source pipelines, we acknowledge risks of misuse. Detection systems could be employed for content censorship or to unfairly discriminate against AI-assisted writing, which may affect legitimate uses of AI tools in education, research, and creative writing. **We do not infer communicative sincerity** or the speaker's intent; scores estimate generation provenance only. The model is designed to assist rather than replace human judgment. Users should consider context and false positives, especially on expert prose, before acting on a front-end flag.

Acknowledgments

This work is supported by the National Key Research and Development Program of China under grants 2023YFC3304501 (project "Data Fusion–Driven Dynamic Perception and Early Warning of Social Governance Risks," topic "Quality Assurance of Multi-Source Fusion Data for Social Governance"). The released corpus is hosted at https://huggingface.co/datasets/layce/giantmelon-zh.

References

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

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

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

Liam Dugan, Alyssa Hwang, Filip Trhlik, Josh Magnus Ludan, Andrew Zhu, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. Raid: A shared benchmark for robust evaluation of machine-generated text detectors. *arXiv preprint arXiv:2405.07940*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. *Preprint*, arXiv:1906.04043.

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

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha,

Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting LLMs with binoculars: Zero-shot detection of machine-generated text. In *International Conference on Machine Learning*, pages 17519–17537. PMLR.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, pages 2251–2265, New York, NY, USA. Association for Computing Machinery.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Damien Riquelme, Alastair Norman, Saffron Henderson, Liang Li, and 5 others. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

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

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. Tinybert: Distilling bert for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR.

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

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

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Dominik Macko, Jakub Kopál, Robert Moro, and Ivan Srba. 2025. Multisocial: Multilingual benchmark of machine-generated text detection of social-media texts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 727–752, Vienna, Austria. Association for Computational Linguistics.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. Multitude: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987.

Eric Mitchell, Yoonkyung Lee, Alex Khazatsky, Kai Krishna, Mohit Iyyer, Hal Daumé III, and Shashank Singh. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *Preprint*, arXiv:1908.09203.

Michal Spiegel and Dominik Macko. 2024. Imgtb: A framework for machine-generated text detection benchmarking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 169–179.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

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

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Rongsheng Wang, Haoming Chen, Ruizhe Zhou, Han Ma, Yaofei Duan, Yanlan Kang, Songhua Yang, Baoyu Fan, and Tao Tan. 2024. Llm-detector: Improving ai-generated chinese text detection with open-source llm instruction tuning. *arXiv preprint arXiv:2402.01158*.

Zhuoshang Wang, Yubing Ren, Guoyu Zhao, Xiaowei Zhu, Hao Li, and Yanan Cao. 2026. EnsemJudge: Enhancing reliability in chinese LLM-generated text detection through diverse model ensembles. *arXiv preprint arXiv:2603.27949*.

Appendix

A RAID Benchmark Main Results (Complete Table)

This appendix provides complete AUROC performance data for GiantMelon-EN and major baseline detectors on the RAID benchmark under the no-adversarial-attack setting (`adversarial attack=none`, `domain=all`, `decoding strategy=all`). Figure 4 includes all generator models (chatgpt, gpt4, gpt3, gpt2, mistral, mistral-chat, cohere, cohere-chat, llama-chat, mpt, mpt-chat) and the Aggregate metric.

B RAID Benchmark Adversarial Attack Detailed Results

The RAID benchmark provides 11 adversarial attack strategies to evaluate detector robustness under adversarial scenarios. This appendix presents GiantMelon-EN's performance across various adversarial attacks, comparing against major baselines. For presentation clarity, we display only current mainstream generation models (chatgpt, gpt4, mistral, mistral-chat, cohere, cohere-chat, llama-chat); results for other models (e.g., gpt2, gpt3, mpt, mpt-chat) are included in the Aggregate metric.

B.1 All Attacks Summary

Table 14 presents performance comparison under all adversarial attack types aggregated.

B.2 Detailed Results by Attack Type

The following subsections present detailed AUROC performance under each of the 11 adversarial attack types.

B.2.1 Zero Width Space

B.2.2 Alternative Spelling

B.2.3 Article Deletion

B.2.4 Homoglyph

B.2.5 Insert Paragraphs

B.2.6 Number

B.2.7 Paraphrase

B.2.8 Perplexity Misspelling

B.2.9 Synonym

B.2.10 Upper Lower

B.2.11 Whitespace

B.3 Key Findings

We summarize the key findings from the detailed adversarial attack results:

1. **Overall robustness**: Under all adversarial attacks aggregated, GiantMelon-EN achieves an Aggregate AUROC of 0.999, matching or slightly outperforming SpeedAI (0.998) and significantly outperforming other public detectors.

2. **Performance across different attack types**:

   - **Whitespace attack**: GiantMelon-EN performs best, achieving perfect Aggregate 1.000, with 1.000 on chatgpt, gpt4, mistral-chat, llama-chat columns.
   - **Upper Lower attack**: GiantMelon-EN achieves perfect Aggregate 1.000, with 1.000 on chatgpt, gpt4, mistral-chat, llama-chat columns.
   - **Alternative Spelling attack**: GiantMelon-EN achieves Aggregate 1.000, with perfect 1.000 on chatgpt, gpt4, mistral-chat, llama-chat columns.
   - **Number attack**: GiantMelon-EN achieves 0.999 on Aggregate, with perfect 1.000 on chatgpt, gpt4, mistral-chat, llama-chat columns.
   - **Perplexity Misspelling attack**: GiantMelon-EN achieves 0.999 on Aggregate, with perfect 1.000 on chatgpt, gpt4, mistral-chat, llama-chat columns.

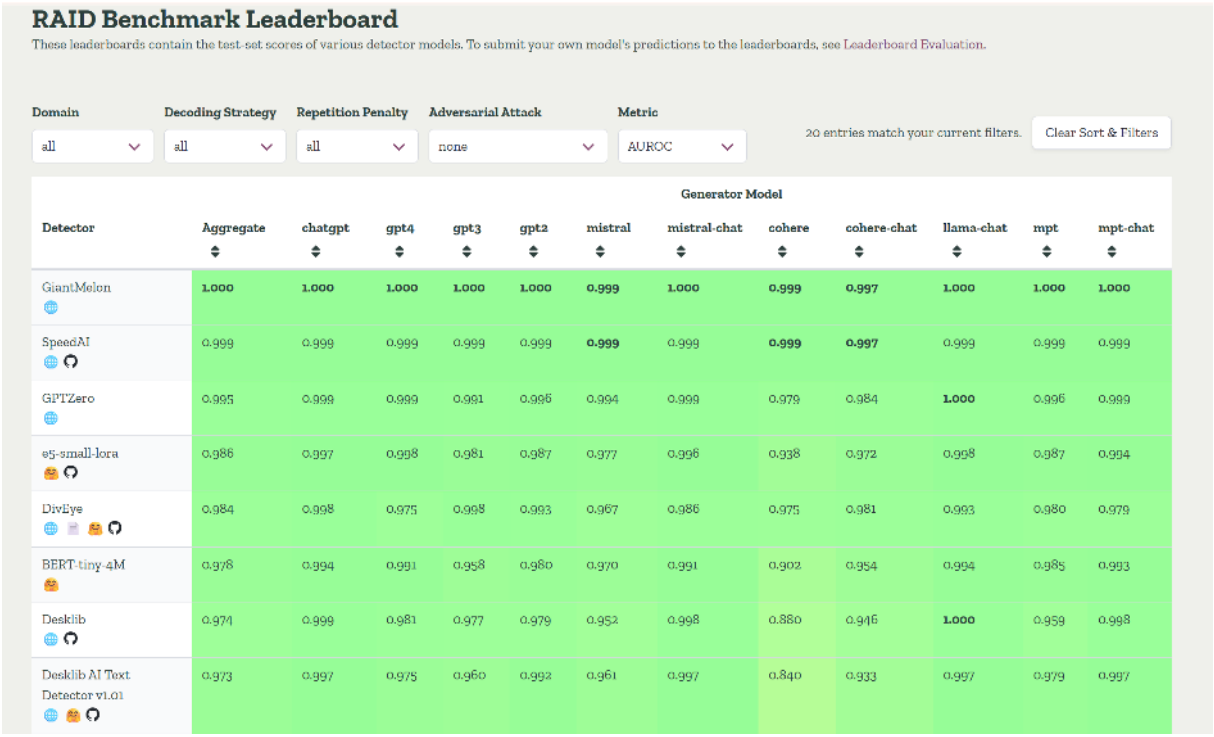


Figure 4: Complete AUROC performance comparison of detectors on the RAID benchmark (no adversarial attack setting), including all generator models.

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**0.999**	**1.000**	**1.000**	**0.998**	**1.000**	**0.996**	**0.996**	**1.000**
SpeedAI	0.998	0.999	0.999	0.998	0.999	0.996	0.995	0.999
GPTZero	0.993	0.999	0.998	0.989	0.997	0.968	0.978	0.999
e5-small-lora	0.968	0.982	0.982	0.958	0.980	0.919	0.955	0.981
BERT-tiny-4M	0.967	0.984	0.980	0.959	0.980	0.894	0.945	0.982
DivEye	0.951	0.964	0.927	0.932	0.950	0.920	0.942	0.965
Desklib	0.949	0.982	0.959	0.925	0.979	0.847	0.919	0.981
Desklib AI Text Detector v1.01	0.948	0.978	0.947	0.934	0.974	0.804	0.904	0.975

Table 14: AUROC performance comparison under all adversarial attack types aggregated.

- **Article Deletion attack**: GiantMelon-EN achieves 0.999 on Aggregate, with perfect 1.000 on chatgpt, gpt4, mistral-chat, llama-chat columns.
- **Insert Paragraphs attack**: GiantMelon-EN achieves 0.999 on Aggregate, with perfect 1.000 on chatgpt, gpt4, mistral-chat, llama-chat columns.
- **Synonym attack**: GiantMelon-EN achieves 0.999 on Aggregate, with perfect 1.000 on chatgpt, gpt4, mistral-chat, llama-chat columns.
- **Zero Width Space attack**: GiantMelon-EN achieves 0.999 on Aggregate, with perfect 1.000 on chatgpt, gpt4, mistral-chat, llama-chat columns.
- **Homoglyph attack**: GiantMelon-EN achieves 0.999 on Aggregate, with 1.000 on chatgpt, gpt4, mistral, cohere-chat columns, but 0.998 on mistral-chat and 0.996 on cohere.
- **Paraphrase attack**: This is the most challenging attack type. GiantMelon-EN achieves 0.995 on Aggregate, still superior to SpeedAI (0.995) and other baselines, but shows slight drops on cohere series (cohere: 0.981, cohere-chat: 0.991).

3. **Comparison with baselines**:

- GiantMelon-EN achieves or outperforms SpeedAI's performance across all attack types.
- Compared to public systems such as

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**0.999**	**1.000**	**1.000**	**0.999**	**1.000**	**0.997**	**0.996**	**1.000**
SpeedAI	0.998	0.999	0.999	0.998	0.999	0.997	0.994	0.999
GPTZero	0.995	0.999	0.999	0.991	0.994	0.979	0.984	1.000
e5-small-lora	0.986	0.997	0.998	0.981	0.977	0.938	0.972	0.998
BERT-tiny-4M	0.978	0.994	0.991	0.958	0.970	0.902	0.954	0.994

Table 15: AUROC performance under Zero Width Space attack.

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**1.000**	**1.000**	**1.000**	**0.999**	**1.000**	**0.999**	**0.997**	**1.000**
SpeedAI	0.999	0.999	0.999	0.999	0.999	0.999	0.996	0.999
GPTZero	0.995	0.999	0.999	0.991	0.996	0.976	0.983	1.000
e5-small-lora	0.985	0.997	0.997	0.980	0.986	0.932	0.969	0.998
BERT-tiny-4M	0.976	0.993	0.990	0.955	0.978	0.894	0.950	0.994

Table 16: AUROC performance under Alternative Spelling attack.

GPTZero, e5-small-lora, and BERT-tiny-4M, GiantMelon-EN achieves significant advantages across all attack types and all generation models.

- Under the most challenging Paraphrase attack, GiantMelon-EN maintains an Aggregate AUROC of 0.995, while GPTZero drops to 0.976 and BERT-tiny-4M drops to 0.938.

4. **Advantages of word-level modeling with symbol stripping**: GiantMelon-EN's word-level encoder combined with aggressive symbol stripping performs excellently when facing character-based adversarial attacks such as homoglyphs, zero-width spaces, and whitespace characters, validating the benefits of normalization before tokenization. In particular, under Whitespace and Upper Lower attacks, GiantMelon-EN achieves perfect Aggregate 1.000, demonstrating that formatting perturbations are neutralized by preprocessing rather than by novel architectural robustness alone.

C Full Architectural Ablation

D Construction Prompts, Translation QC, and Tokenization

Chinese AI-written class. Each source document is processed in three stages. The generator receives, in order:

1. *abridge the above text to no more than 50 words*;
2. *plan some details for the abridged text*;
3. *expand the text according to the details*.

After stage (1), almost all source detail is gone and only a topic seed remains. Stages (2)–(3) are therefore model-authored content, not a light paraphrase of the human original. Human-class documents are the unrewritten Wikipedia and non-Technology THUCNews sources.

English parallel translation. Both classes are translated with GPT-4o under the instruction *translate the text material strictly into English*. Segments are aligned to the Chinese source spans. We drop pairs whose English/Chinese length ratio is extreme relative to the batch distribution, then spot-check remaining pairs by hand for omitted clauses and invented facts. Qwen3 perplexity on abridged-then-expanded Chinese is 8.372 and 12.519 after translation; cross-language perplexities are not comparable, so these figures are a sanity check only. We switched from Qwen3 translation to GPT-4o after observing substantial factual errors in the former. GPT-4o perplexity is not directly available.

Chunking, split, and vocabularies. Training windows use length $S{=}512$ and stride $\Delta{=}256$ (50% overlap). The Chinese random split is 9:1 at **document** level so that chunks of one document do not appear on both sides; the same document-level rule is used for source-held-out splits. The Chinese character vocabulary maps CJK Unified Ideographs with <pad> $= 0$ and <unk> $= 1$. English GiantMelon-EN uses dictionary word tokens after symbol stripping; any learner-grade English dictionary suffices. We used a licensed dictionary that we do not redistribute.

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**0.999**	**1.000**	**1.000**	**0.999**	**1.000**	**0.999**	**0.996**	**1.000**
SpeedAI	0.999	0.999	0.999	0.999	0.999	0.998	0.996	0.999
GPTZero	0.993	0.999	1.000	0.988	0.991	0.967	0.977	1.000
e5-small-lora	0.982	0.996	0.996	0.973	0.971	0.915	0.961	0.998
BERT-tiny-4M	0.977	0.995	0.992	0.953	0.967	0.903	0.954	0.994

Table 17: AUROC performance under Article Deletion attack.

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**0.999**	**1.000**	**1.000**	**1.000**	**0.998**	**0.996**	**1.000**	**0.999**
SpeedAI	0.999	0.999	0.999	0.999	0.997	0.995	0.999	0.999
GPTZero	0.995	0.999	0.999	0.991	0.994	0.979	0.984	1.000
e5-small-lora	0.809	0.827	0.825	0.805	0.819	0.779	0.818	0.805
BERT-tiny-4M	0.897	0.897	0.896	0.912	0.900	0.868	0.894	0.894

Table 18: AUROC performance under Homoglyph attack.

E Dataset Submission

We provide a subset of our Chinese machine-generated text detection dataset as supplementary material for this submission. The dataset archive (`datasubmission.zip`) contains the **Wiki-Claude** subset, which consists of 1,982 paired samples. Each pair contains:

- A human-written text from Wikipedia (Chinese)
- An AI-rewritten text generated by Claude 4.5 Opus

Dataset structure: The dataset is organized as `Wiki-Claude/human/` and `Wiki-Claude/ai/`, where files are named `wiki{ID}` and `claude{ID}` respectively. Files with the same numeric ID form a pair (e.g., `wiki206678` ↔ `claude206678`).

Release: The camera-ready files are at https://huggingface.co/datasets/layce/giantmelon-zh. They include the cleaned Chinese news documents, these 1,982 Wiki–Claude pairs, a paired Chinese–English Wikipedia slice, and the case-study passages in Appendix F. The repository is gated for research use. Model weights and the licensed English dictionary are not included.

F Case Study Details

F.1 Experimental Setup

To validate our model's detection patterns and interpretability, we selected several text samples from outside our training set and applied gradient-based importance analysis (Section 3.6) to visualize word-level contributions to detection decisions. For visualization purposes, we performed word segmentation on the character-level importance scores to aggregate character-level importance into word-level importance, enabling clearer interpretation of which types of words the model focuses on when making detection decisions.

F.2 Case Descriptions

The following cases demonstrate GiantMelon's detection performance across diverse text types. For AI-generated texts, high-importance words are visualized in red; for human-written texts, high-importance words are visualized in blue. Each case includes the detection confidence score, word importance visualization, and source attribution.

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**0.999**	**1.000**	**1.000**	**0.999**	**1.000**	**0.996**	**0.997**	**1.000**
SpeedAI	0.999	0.999	0.999	0.999	0.999	0.996	0.995	0.999
GPTZero	0.995	0.999	0.999	0.991	0.993	0.979	0.983	1.000
e5-small-lora	0.983	0.996	0.997	0.977	0.973	0.924	0.965	0.998
BERT-tiny-4M	0.976	0.994	0.990	0.955	0.968	0.902	0.949	0.994

Table 19: AUROC performance under Insert Paragraphs attack.

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**0.999**	**1.000**	**1.000**	**0.999**	**1.000**	**0.999**	**0.997**	**1.000**
SpeedAI	0.999	0.999	0.999	0.999	0.999	0.997	0.996	0.999
GPTZero	0.995	0.999	0.999	0.991	0.996	0.978	0.983	1.000
e5-small-lora	0.983	0.996	0.997	0.977	0.995	0.936	0.971	0.998
BERT-tiny-4M	0.976	0.994	0.990	0.968	0.990	0.900	0.949	0.994

Table 20: AUROC performance under Number attack.

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**0.995**	**0.999**	**0.998**	**0.995**	**0.991**	**0.981**	**0.991**	**0.999**
SpeedAI	0.995	0.999	0.998	0.995	0.991	0.980	0.991	0.998
GPTZero	0.976	0.996	0.992	0.970	0.953	0.941	0.965	0.994
e5-small-lora	0.968	0.992	0.990	0.972	0.938	0.933	0.963	0.990
BERT-tiny-4M	0.938	0.969	0.955	0.936	0.902	0.871	0.923	0.955
RoBERTa (ChatGPT)	0.847	0.900	0.858	0.873	0.822	0.804	0.835	0.872

Table 21: AUROC performance under Paraphrase attack.

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**0.999**	**1.000**	**1.000**	**0.999**	**1.000**	**0.996**	**0.997**	**1.000**
SpeedAI	0.999	0.999	0.999	0.999	0.999	0.996	0.996	0.999
GPTZero	0.995	0.999	0.999	0.991	0.995	0.977	0.983	1.000
e5-small-lora	0.983	0.996	0.997	0.977	0.983	0.924	0.965	0.998
BERT-tiny-4M	0.976	0.994	0.990	0.968	0.990	0.892	0.949	0.994

Table 22: AUROC performance under Perplexity Misspelling attack.

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**0.999**	**1.000**	**1.000**	**0.999**	**1.000**	**0.991**	**0.993**	**1.000**
SpeedAI	0.998	0.999	0.999	0.999	0.998	0.990	0.993	0.999
GPTZero	0.986	0.998	0.995	0.967	0.990	0.921	0.950	0.998
e5-small-lora	0.980	0.995	0.996	0.971	0.968	0.915	0.958	0.997
BERT-tiny-4M	0.974	0.991	0.987	0.946	0.968	0.896	0.947	0.990

Table 23: AUROC performance under Synonym attack.

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**1.000**	**1.000**	**1.000**	**0.999**	**1.000**	**0.999**	**0.997**	**1.000**
SpeedAI	0.999	0.999	0.999	0.999	0.999	0.998	0.996	0.999
GPTZero	0.995	0.999	0.999	0.991	0.996	0.978	0.983	1.000
e5-small-lora	0.983	0.996	0.997	0.977	0.995	0.936	0.971	0.998
BERT-tiny-4M	0.976	0.994	0.990	0.968	0.990	0.900	0.949	0.994

Table 24: AUROC performance under Upper Lower attack.

Detector	Aggregate	chatgpt	gpt4	mistral	mistral-chat	cohere	cohere-chat	llama-chat
GiantMelon-EN (ours)	**1.000**	**1.000**	**1.000**	**0.999**	**1.000**	**0.999**	**0.997**	**1.000**
SpeedAI	0.999	0.999	0.999	0.998	0.999	0.998	0.996	0.999
GPTZero	0.995	0.999	0.999	0.991	0.994	0.979	0.984	1.000
e5-small-lora	0.986	0.997	0.998	0.977	0.996	0.938	0.972	0.998
BERT-tiny-4M	0.978	0.994	0.991	0.970	0.991	0.902	0.954	0.994

Table 25: AUROC performance under Whitespace attack.

attn	pool	agg	AUC (%)	ACC (%)	F1 (%)
bi	unmasked_mean	weighted_mean	**99.44**	**98.1**	97.02
bi	unmasked_mean	mean	99.42	96.95	96.99
bi	masked_mean	weighted_mean	99.21	96.21	96.29
bi	masked_mean	mean	99.20	96.17	96.26
causal	unmasked_mean	weighted_mean	99.09	96.99	97.02
causal	unmasked_mean	mean	98.95	97.15	97.16
causal	masked_mean	weighted_mean	98.88	96.50	96.56
causal	masked_mean	mean	98.80	96.55	96.60
causal	unmasked_mean	max	98.68	96.98	96.99
causal	masked_mean	max	98.00	94.72	94.82
bi	unmasked_mean	max	96.06	92.02	92.55
bi	masked_mean	max	94.35	88.11	89.31
causal	max	max	50.86	53.94	67.75
bi	max	max	56.82	51.82	67.29
causal	max	mean	69.35	50.56	66.78
bi	max	mean	70.89	50.41	66.73
causal	max	weighted_mean	71.34	50.54	66.77
bi	max	weighted_mean	76.45	50.46	66.74

Table 26: Full architectural ablation on the self-constructed test set (symbol cleaning fixed). Default: `bi`, `unmasked_mean`, `weighted_mean`.



Figure 5: Case Study 1: AI-generated news report detection analysis. Three AI-generated texts (news report, encyclopedia entry, literature) from Doubao AI, all correctly identified with high confidence (96.7%–99.8%). The model focuses on standardized policy terminology, technical vocabulary, and formulaic emotional expressions (visualized in red).



Figure 6: Case Study 2: AI-generated text detection analysis (Deepseek AI). Three AI-generated texts (news report, encyclopedia entry, literature) from Deepseek AI, correctly identified with varying confidence (52.5%–99.8%). The encyclopedia entry containing real historical facts achieves lower confidence (52.5%), while the model focuses on technical terminology, scientific terms, and literary descriptions (visualized in red).



AI Text Detection Analysis

Cannot rule out AI assistance (18.3%) Incorrect

Word Importance Visualization

近日，中国科学技术大学宣布，我国首个量子计算原型机“九章三号”在光子数和算力方面实现重大突破，其处理特定问题的速度比全球最快的超级计算机快一千万倍以上，标志着我国量子计算研究领域继续处于国际领先地位。据介绍，“九章三号”是基于光量子计算路线的量子计算原型机，由中国科学技术大学潘建伟院士团队研发。与前一代“九章二号”相比，“九章三号”的光子数从113个提升至255个，量子比特数和量子门操作精度也得到了大幅提升。在处理“高斯玻色取样”问题时，“九章三号”的算力达到了新的高度，完成同样的任务，“九章三号”仅需0.0001秒，而全球最快的超级计算机则需要1000多秒，算力提升了一千万倍以上。量子计算具有算力强大、处理速度快等优势，在密码学、材料科学、药物研发、人工智能等多个领域具有广阔的应用前景。“九章三号”的成功研发，进一步验证了光量子计算路线的可行性和优越性，为我国量子计算技术的产业化发展奠定了坚实基础。潘建伟院士表示，下一步，团队将继续优化“九章三号”的性能，推动量子计算原型机向实用化、商业化方向发展，同时加强与产业界的合作，探索量子计算在各个领域的应用场景。国家科技部相关负责人表示，我国高度重视量子计算技术的研发，将继续加大对量子科技领域的科研投入，支持科研团队开展关键核心技术攻关，推动我国量子科技产业的快速发展。业内专家指出，“九章三号”的算力突破，不仅提升了我国在全球量子计算领域的竞争力，也为全球量子计算技术的发展作出了重要贡献，将加速量子计算从理论研究走向实际应用的进程。

Generated by Doubao AI (News Report)

AI Text Detection Analysis

Suspected AI-assisted content detected (36.2%) Incorrect

Word Importance Visualization

景泰蓝，学名“铜胎掐丝珐琅”，是我国传统工艺美术品中的瑰宝，因其在明代景泰年间（1450—1457年）工艺最为成熟，且以蓝色釉料为主色调，故得名“景泰蓝”。景泰蓝的历史可追溯至元代，当时由波斯传入我国，经过明清两代的发展，逐渐形成了独具特色的艺术风格和精湛的制作工艺，成为宫廷御用工艺品，深受皇室贵族的喜爱。景泰蓝的制作工艺复杂繁琐，主要包括制胎、掐丝、点蓝、烧蓝、磨光、镀金等六道核心工序。制胎是将铜片裁剪、敲打、焊接成所需的器物形状，如瓶、罐、盘、盒等；掐丝是用细铜丝按照设计图案的轮廓弯曲成型，然后用白芨浆粘贴在铜胎表面；点蓝是将各种颜色的釉料填充到掐丝形成的花纹空隙中；烧蓝是将填充好釉料的铜胎放入窑中烧制，使釉料熔化并与铜胎牢固结合。这一工序需要反复进行多次，以确保釉料的平整和色泽的均匀；磨光是用砂纸、磨石等工具将烧制好的器物表面打磨光滑；镀金是在器物的表面镀上一层金，使其更加华丽精美，同时也起到防锈保护的作用。景泰蓝的艺术特点主要体现在造型、纹样和色彩三个方面。造型上，景泰蓝器物造型多样，既有传统的古典造型，也有现代的创新造型，线条流畅优美，比例协调匀称；纹样上，景泰蓝的纹样丰富多样，涵盖了花卉、动物、人物、山水、吉祥图案等多个领域，纹样设计精美细腻，富有层次感；色彩上，景泰蓝的色彩鲜艳明快，对比强烈，以蓝色为主色调，搭配红、黄、绿、白等多种颜色，形成了独特的色彩风格。近年来，随着传统工艺美术的复兴，景泰蓝艺术得到了广泛的传承和发展。相关部门加大了对景泰蓝技艺传承人的培养力度，鼓励技艺创新，推动景泰蓝艺术与现代生活相结合，开发出了一系列兼具艺术价值和实用价值的景泰蓝产品，如茶具、餐具、装饰品等。同时，景泰蓝也成为我国对外文化交流的重要载体，多次在国际展览中展出，受到了全球观众的喜爱和赞誉。

Generated by Doubao AI (Encyclopedia Entry)

AI Text Detection Analysis

Cannot rule out AI assistance (4.6%) Incorrect

Word Importance Visualization

快节奏的生活像一列呼啸而过的列车，载着我们匆匆前行，很多人只顾着追赶前方的目标，却忘了抬头看看沿途的风景。我们习惯了边走路边刷手机，习惯了吃快餐式的饭菜，习惯了用碎片化的时间获取信息，却渐渐失去了静下心来感受生活的能力。其实，生活的美好，往往藏在慢下来的时光里。慢下来，才能读懂一本书的深意。不是走马观花地浏览文字，而是沉下心来，跟着作者的笔触走进另一个世界，感受人物的喜怒哀乐，体会文字背后的思考与感悟。当我们放下浮躁，一字一句地品读，才能在墨香中找到心灵的慰藉，在文字的海洋里汲取成长的力量。慢下来，才能品味一顿饭的香甜。不是狼吞虎咽地填饱肚子，而是用心感受食材的本味，体会烹饪的乐趣。看着食材在锅中慢慢蜕变，闻着空气中弥漫的香气，和家人朋友围坐在一起，聊着家常，分享美食，这份温馨与惬意，是快餐永远无法给予的。慢下来，才能发现身边的风景。春日里抽芽的新绿，夏日里聒噪的蝉鸣，秋日里飘落的黄叶，冬日里纷飞的雪花，这些细碎的美好，都需要我们停下脚步，用心去感受。生活不是一场急于求成的竞赛，而是一段需要细细品味的旅程。我们不必总是步履匆匆，偶尔慢下来，整理好心情，才能更好地出发。学会慢下来，才能读懂生活的诗意，才能在平凡的日子里，找到属于自己的幸福与安宁。

Generated by Doubao AI (Literature)

Figure 7: Case Study 3: AI-generated text detection analysis (misclassified cases). Three AI-generated texts (news report, encyclopedia entry, literature) from Doubao AI, incorrectly classified with low confidence (4.6%–36.2%). When AI-generated content incorporates substantial real-world facts (scientific data, historical information) or mimics human thought patterns (philosophical reflections), the model's confidence decreases, as these authentic elements reduce the prominence of AI generation signatures (visualized in red).



Figure 8: Case Study 4: Human-written text detection analysis (classical literature). Three human-written texts (essay by Zhu Ziqing, newspaper editorial from Southern Weekend, novel by Wang Xiaobo), all correctly identified with very low AI confidence (0.0%–36.0%). The model focuses on unique literary expressions, authentic details, and genuine emotional nuances (visualized in blue), reflecting the idiosyncratic nature of human writing.



Figure 9: Case Study 5: Human-written text detection analysis (translated literature). Two translated literary works (Germinal by Zola, Don Quixote by Cervantes), correctly identified with low AI confidence (2.8%–23.7%). The model focuses on rich detail descriptions, authentic environmental depictions, and unique literary styles (visualized in blue), demonstrating the model's ability to recognize human writing characteristics in translated texts.



Figure 10: Case Study 6: Human-written text detection analysis (modern poetry). A modern Chinese poem (Dayanhe—My Wet Nurse by Ai Qing), correctly identified with very low AI confidence (7.4%). The model focuses on poetic structures (repetition, parallelism), authentic emotional expressions, and specific life details (visualized in blue), reflecting the genuine emotional experience and unique poetic techniques of human writing.



Figure 11: Case Study 7: Human-written text detection analysis (misclassified cases). Two human-written texts (academic paper, news report), incorrectly classified with moderate AI confidence (50.8%–55.4%). When human-written texts contain dense professional terminology or structured policy expressions, the model may misclassify them, as these patterns resemble AI-generated "term stacking" and structured templates (visualized in blue). This reveals the model's limitation in distinguishing expert human writing from AI-generated content in specialized domains.