

Article

Collaborative Multi-Agent Method for Zero-Shot LLM-Generated Text Detection

Gang Sun ^{1,2}, Bowen Li ³, Ying Zhou ^{3,*}, Yi Zhu ³ and Jipeng Qiang ³ ¹ Department of Anhui Engineering Research Center for Intelligent Computing and Information Innovation, Fuyang Normal University, Fuyang 236037, China² Department of Computer and Information Engineering, Fuyang Normal University, Fuyang 236037, China³ Department of Information and Artificial Intelligence, Yangzhou University, Yangzhou 225127, China

* Correspondence: mz120241055@stu.yzu.edu.cn

Abstract

With the rapid proliferation of large language models (LLMs), distinguishing machine-generated text from human-authored content has become increasingly critical for ensuring content authenticity, academic integrity, and trust in information systems. However, detecting text generated by LLMs remains a challenging problem, particularly in zero-shot settings where labeled data and domain-specific tuning are unavailable. To address this challenge, in this paper, we propose a novel Collaborative Multi-Agent Zero-Shot Detection framework (CMA-ZSD). In contrast to existing methods based on watermarking, statistical heuristics, or neural classifiers, our CMA-ZSD employs three functionally heterogeneous agents that perform differentiated perturbations of the input text. By jointly modeling semantic consistency, grammatical normalization, and feature-level reconstruction, our method captures intrinsic asymmetries between human-authored and LLM-generated text. A semantic similarity evaluation mechanism, combined with majority voting, enables robust and interpretable detection decisions that balance individual agent autonomy with collective consensus. Extensive experiments across 11 domains demonstrate the effectiveness of our method, with its zero-shot detection achieving accuracy comparable to domain-finetuned models in specific domains such as Finance and Reddit-dli5.

Keywords: collaborative multi-agent; LLM-generated text detection; zero-shot

1. Introduction

In recent years, LLMs have demonstrated remarkable capabilities in text generation. The content they produce has reached a level comparable to human creation in terms of natural fluency, logical coherence, contextual richness, and other dimensions [1]. Beyond their generative capabilities, LLMs have been increasingly applied in high-stakes domains such as personalized healthcare, supporting clinical decision-making and patient risk assessment [2]. Recent studies have shown that even state-of-the-art LLMs can produce variable reasoning chains and semantic inconsistencies, highlighting the need for robust evaluation and detection of machine-generated text [3]. While these advances have substantially promoted the development of natural language processing applications, they have also raised growing concerns regarding content authenticity, academic integrity, and copyright protection [4]. The increasing accessibility and widespread deployment of LLMs make it progressively more difficult for ordinary users to reliably distinguish between human-authored and machine-generated text based on subjective judgment alone. As a



Academic Editor: Antony Bryant

Received: 12 January 2026

Revised: 20 February 2026

Accepted: 3 March 2026

Published: 16 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

result, the development of accurate and reliable methods for detecting LLM-generated text has become an urgent and important research problem for both academia and industry.

Existing approaches to LLM-generated text detection can be broadly categorized into three main paradigms: watermarking-based methods, statistics-based methods, and neural model-based methods [5]. Watermarking techniques embed identifiable signals into model outputs during generation, either by manipulating token sampling distributions or by post-processing generated text, enabling reliable detection when the watermarking scheme is known [6]. However, these methods require control over or cooperation from the generation process, making them ineffective for detecting text produced by unmodified or unknown models. Statistics-based methods exploit intrinsic statistical properties of generated text, such as token probability distributions, perplexity, repetition patterns, or sensitivity to perturbations [7], with representative approaches including rank-based metrics and perturbation-based detectors such as DetectGPT [8]. Despite their zero-shot nature, such methods are often sensitive to domain shifts, text length variations, and decoding strategies, leading to unstable performance in real-world scenarios. Neural model-based methods frame detection as a supervised or semi-supervised classification task, leveraging feature-based classifiers or fine-tuned pre-trained language models (e.g., RoBERTa-based detectors [9]) to distinguish between human-authored and machine-generated content. While these approaches can achieve high accuracy within specific domains, existing methods typically depend on large-scale labeled data and domain-specific fine-tuning, resulting in high annotation costs and limited generalization across domains and model variants.

To address the limitations of existing LLM-generated text detection methods, in this paper, we propose a Collaborative Multi-Agent framework for Zero-Shot Detection, termed CMA-ZSD. Unlike watermarking-based methods, CMA-ZSD does not require access to or control over the text generation process, enabling the detection of content produced by arbitrary and previously unseen LLMs. By operating in a zero-shot setting, the proposed framework avoids dependence on labeled data and domain-specific fine-tuning, thereby alleviating the scalability and generalization issues inherent in neural classifier-based methods. Moreover, instead of relying on a single statistical signal, CMA-ZSD integrates multiple complementary perspectives through collaboration among heterogeneous agents, which enhances robustness to domain shifts, decoding strategies, and surface-level paraphrasing. This design allows our CMA-ZSD to achieve a balanced trade-off between detection accuracy, cross-domain robustness, and practical deployability.

Specifically, CMA-ZSD constructs three functionally heterogeneous agents, including a text revision agent, a grammar correction agent, and a feature reconstruction agent, to perform differentiated perturbations of the input text from semantic, grammatical, and structural perspectives. Each agent independently processes the same input text and produces a transformed version, which is subsequently compared with the original text using a semantic similarity scoring mechanism. The individual detection results produced by the agents are then aggregated through a majority voting strategy to determine the final classification. This collaborative design ensures that diverse linguistic cues are jointly considered while reducing the impact of individual agent bias, thereby improving the stability and interpretability of zero-shot LLM-generated text detection. Extensive experiments across 11 diverse domains demonstrate that our CMA-ZSD consistently outperforms existing zero-shot baselines and achieves performance comparable to, or exceeding, that of domain-finetuned methods in several domains, validating its effectiveness and robustness in real-world settings.

The main contributions of this paper can be summarized as follows:

(1) We propose a novel collaborative multi-agent framework for zero-shot LLM-generated text detection, termed CMA-ZSD. To the best of our knowledge, this is the

first study to introduce multi-agent collaboration into zero-shot generated text detection, enabling robust detection without access to the generation process or reliance on labeled training data.

(2) We design a heterogeneous agent collaboration mechanism that integrates semantic revision, grammatical correction, and feature-level reconstruction, together with a similarity-based decision strategy and majority voting. This design effectively captures complementary linguistic cues and enhances robustness against domain shifts, paraphrasing, and decoding variability.

(3) We conduct extensive experiments across 11 diverse domains to systematically evaluate the proposed method. Experimental results demonstrate that CMA-ZSD consistently outperforms existing zero-shot baselines and achieves performance comparable to, or exceeding, that of domain-finetuned methods in several domains, highlighting its strong cross-domain generalization capability and practical applicability.

2. Related Work

With LLMs demonstrating superior performance in natural language processing tasks, the misuse of their generated content has become increasingly prominent, posing significant challenges for detection. Recent studies suggest that machine-generated text exhibits distinct distributional and geometric properties compared to human-authored text. For example, DetectGPT shows that LLM-generated text tends to lie near high-probability regions of the model distribution and demonstrates characteristic curvature behavior under perturbations [8]. Similarly, likelihood-based detection methods observe that generated text often exhibits smoother probability landscapes and greater local consistency compared to human-written text [10,11]. These findings support the hypothesis that perturbation-based analysis can reveal structural differences between human and machine-generated language. Existing approaches can be categorized into three paradigms:

2.1. Watermarking-Based Methods

Watermarking-based methods aim to detect LLM-generated text by embedding identifiable signals into the generation process or the generated content itself, thereby enabling post hoc verification of machine authorship [12]. As a proactive detection paradigm, watermarking has attracted considerable attention due to its theoretical reliability when the watermarking scheme is known and preserved [13].

Early efforts in this domain focused on logit-based watermarking, where the token sampling distribution of an autoregressive language model is adjusted to promote specific patterns in generated text. A foundational example of this strategy is the work by Kirchenbauer et al., who introduced a logits-bias watermarking framework (WLLM) that divides the vocabulary into “green” and “red” token sets and biases generation toward the green set, such that the statistical imbalance in token usage can later be detected with high confidence [14].

Building on this premise, subsequent research explored more adaptive watermarking strategies. For example, Liu et al. proposed an adaptive text watermarking technique that selectively applies watermarking to high-entropy token distributions to maintain text quality while preserving watermark robustness [15]. Meanwhile, Dabiriaghdam et al. presented SimMark, a sentence-level similarity-based watermarking algorithm that embeds imperceptible patterns by leveraging semantic similarity of sentence embeddings combined with rejection sampling, offering enhanced robustness against paraphrasing attacks [16].

Beyond logits-manipulation techniques, recent works like REMARK-LLM explore message-encoding and reparameterization modules to infuse binary signatures into LLM

outputs while preserving semantic integrity. This framework efficiently encodes watermark bits and ensures their recoverability even under certain editing operations [17].

Despite the advances in embedding and detecting watermarks under controlled conditions, a key limitation of watermarking-based methods is their reliance on access to the generation process or the watermark key. If the generation model is unknown, uncooperative, or has not been watermarked, these techniques cannot be applied, and watermarks can be weakened or removed through paraphrasing, translation, or other text transformations. Consequently, while watermarking offers a theoretically sound approach under specific conditions, its applicability in open, uncontrolled environments remains limited.

2.2. Statistics-Based Methods

Statistics-based methods detect LLM-generated text by exploiting statistical regularities in language patterns that differentiate machine outputs from human writing [18].

Early work in this area explored basic likelihood and rank statistics derived from language model outputs. For example, DetectLLM leverages log-rank information from token distributions to perform zero-shot detection without requiring training data [19]. By comparing log-rank statistics with and without perturbations, DetectLLM variants (e.g., DetectLLM-LRR and DetectLLM-NPR) have shown improved detection performance over earlier baselines across multiple datasets and models, highlighting how distributional signals can be used for robust detection.

A related line of research analyzes properties of model probability functions. For example, DetectGPT introduces a probability curvature-based criterion for zero-shot detection, observing that text sampled from an LLM tends to occupy negative curvature regions of the model's log-probability landscape [8]. By generating perturbed versions of a given passage and comparing the resulting log probability differences, DetectGPT demonstrates superior discriminative ability relative to simple likelihood measures. Subsequent extensions such as DetectGPT-SC improve detection by incorporating self-consistency checks with masked predictions, further refining statistical cues to distinguish machine-generated text without supervision [20].

Beyond direct probability metrics, researchers have also investigated information density and other psycholinguistically motivated features. For example, in GPT-who, an information density-based detector inspired by the Uniform Information Density principle, text is evaluated for how evenly information is distributed, a characteristic thought to differ between human and LLM outputs [21]. This UID-based statistical approach achieves strong detection performance across multiple benchmarks and provides interpretable features that supplement traditional probability statistics.

While statistics-based methods benefit from being training-free and domain-agnostic, they can be sensitive to model sampling strategies, text length, and domain shifts. Simple metrics like perplexity or n-gram statistics may vary substantially with prompt design or text genre, potentially leading to unstable detection in real-world applications.

2.3. Neural-Based Methods

Neural model-based methods formulate LLM-generated text detection as a supervised or semi-supervised classification task, leveraging neural networks to learn discriminative features that distinguish human-authored text from machine-generated text [22]. These methods typically fine-tune pre-trained language models (PLMs) or train lightweight classification heads using labeled examples, enabling flexible adaptation to diverse domains and generation styles.

One of the earliest works in this category applied fine-tuned BERT for generated text detection. For example, Ippolito et al. demonstrated that a BERT classifier fine-tuned on

labeled examples of human and machine-generated text can achieve competitive detection performance, particularly when training and evaluation data are drawn from similar distributions [23]. This study highlighted the potential of leveraging deep bidirectional representations to capture subtle linguistic cues indicative of generative processes.

More recent studies have focused on improving robustness and generalization across domains and generation models. For example, Tian et al. proposed RoBERTa-MPU, a neural detection framework that incorporates multi-scale semantic representations and positive-unlabeled learning to enhance detection performance for both short and long texts [24]. By leveraging large-scale pre-training and carefully designed learning objectives, this approach demonstrates strong detection accuracy across multiple datasets and LLM variants. Moreover, Koike et al. introduced Outfox, an LLM-based detection framework that applies in-context learning with adversarially generated examples, enabling the detector to adapt to diverse generation styles without explicit fine-tuning [25].

Despite their strong empirical performance, existing neural model-based methods suffer from several inherent limitations. They typically require large amounts of labeled training data, which are costly to obtain and may not be available for newly emerging domains or generation models. Moreover, existing detectors trained on specific LLM outputs often exhibit degraded performance when applied to text generated by unseen architectures or decoding strategies, limiting their cross-model generalization. The high computational cost associated with training and deploying large neural detectors further constrains their practicality in dynamic, real-world detection scenarios.

3. Methodology

3.1. Overall Framework

In this paper, we propose a collaborative multi-agent framework for zero-shot LLM-generated text detection, termed CMA-ZSD, which identifies machine-generated content without relying on labeled training data or domain-specific fine-tuning. The core idea of the framework is to exploit complementary linguistic perspectives through the coordinated interaction of multiple heterogeneous agents, thereby improving robustness and generalization in open-domain scenarios.

As illustrated in Figure 1, our CMA-ZSD consists of three independently operating yet collaboratively integrated agents, each designed to process the same input text using a distinct transformation strategy. Specifically, the text revision agent performs human-like rewriting while preserving semantic meaning, the grammar correction agent normalizes grammatical structures through error correction, and the feature reconstruction agent regenerates text based on abstracted semantic features. These agents introduce differentiated perturbations to the input text from semantic, grammatical, and structural viewpoints, respectively.

For a given input text, each agent produces a transformed version, which is then compared with the original text using a semantic similarity scoring mechanism based on BARTScore. The resulting similarity scores reflect how sensitive the input text is to each type of perturbation. Since human-authored and LLM-generated texts respond differently to such transformations, these scores provide discriminative signals for detection. Each agent independently applies a predefined decision threshold to generate a binary prediction indicating whether the text is human-written or machine-generated.

To integrate the decisions of individual agents, our CMA-ZSD employs a majority voting strategy that aggregates the three binary predictions into a final classification result. This ensemble-based decision mechanism mitigates the influence of individual agent errors and reduces detection instability caused by domain shifts or surface-level paraphrasing. By combining agent autonomy with collaborative decision-making, the overall framework

achieves a balanced trade-off between accuracy, robustness, and interpretability, making it well suited for practical zero-shot LLM-generated text detection.

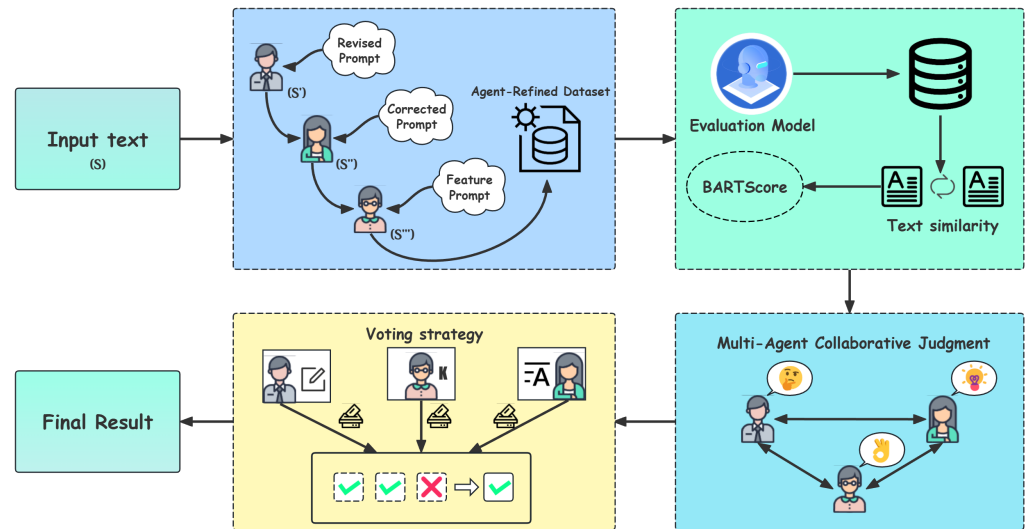


Figure 1. The illustration of our collaborative multi-agent method for zero-shot LLM-generated text detection. Three agents are first designed to process the text. Then the results are evaluated to calculate similarity with BARTScore. The voting mechanism is final conducted to predict the results.

Algorithm 1: Multi-Agent Text Revision

Input: Original human answer set: $\mathcal{H} = \{h_1, h_2, \dots, h_n\}$;

Original LLMs answer set: $\mathcal{C} = \{c_1, c_2, \dots, c_n\}$;

Agent collection: $\mathcal{A} = \{\mathcal{A}_{\text{revise}}, \mathcal{A}_{\text{correct}}, \mathcal{A}_{\text{feature}}\}$

Output: Revised dataset: $D^* = \{\mathcal{H}_k^*, \mathcal{C}_k^*\}$

// (generated by agents $\mathcal{A}_k$, $k \in \{\text{revise}, \text{correct}, \text{feature}\}$)

Dynamic token limit:

Based on original text length $|s|$:

$$\text{max_tokens} = \begin{cases} 150 & \text{if } |s| < 50 \\ 250 & \text{if } 50 \leq |s| < 200 \\ 400 & \text{otherwise} \end{cases}$$

for each sentence index i do

$m = \text{max_tokens}(a_i)$

for each agent $\mathcal{A}_k \in \mathcal{A}$ do

// Generate augmented answer using GPT

$a_{k,i}^* = \text{GPT-3.5-turbo}(\text{prompt}_k, T = 0.7, m)$

// Save to corresponding set **if** $a_{k,i}^* \in \mathcal{H}$ **then**

| $\mathcal{H}_k^* \leftarrow \mathcal{H}_k^* \cup \{a_{k,i}^*\}$

else

| $\mathcal{C}_k^* \leftarrow \mathcal{C}_k^* \cup \{a_{k,i}^*\}$

return Revised dataset D^*

3.2. Agent Design

All three agents are implemented via LLM-based text transformation using gpt-3.5-turbo with domain-specific prompts. For an input text s , each agent generates a perturbed version (s' , s'' , or s''') through conceptually guided rewriting. Notably, the following

equations describe the operational implementation of the proposed framework rather than abstract conceptual relationships. All quantities are computed directly using fixed pre-trained models without parameter optimization.

Text Revision Agent: The core task of the text revision agent is to modify input text and generate a new version directly that maintains semantic consistency while altering the expression form [1], thereby simulating human adjustments to linguistic expression. The optimization objective of the intelligent agent can be formalized as:

$$A_{\text{revise}}(s) = \arg \max_{s' \in \mathcal{T}(s)} [\lambda \cdot S_{\text{sem}}(s, s') - (1 - \lambda) \cdot D_{\text{lex}}(s, s')] \quad (1)$$

where s denotes the original text, s' is the revised version, $\mathcal{T}(s)$ represents the candidate text space constrained by dynamic generation length, $S_{\text{sem}}(\cdot)$ measures semantic consistency, and $D_{\text{lex}}(\cdot)$ quantifies lexical divergence. $\lambda \in [0, 1]$ balances fidelity to semantics against surface-level diversity. This objective explicitly balances semantic preservation and lexical variation to reveal stylistic rigidity in LLM outputs.

LLM-generated texts typically follow fixed linguistic patterns, with revised versions showing high lexical and syntactic similarity to originals. Human-generated texts exhibit inherent expressive diversity [26], producing significantly lower similarity between revisions and originals. Quantifying this discrepancy enables effective detection.

Grammar Correction Agent: The grammar correction agent focuses on grammatical error correction to produce a standardized version of the input text [27]. Importantly, this component does not assume that LLM-generated text contains more grammatical errors than human-authored text. Rather, it examines how each text type responds to standardized normalization. LLM-generated text is typically produced under probabilistic fluency constraints, often resulting in grammatically well-formed and distributionally smooth outputs. Human-authored text, particularly in informal domains, may exhibit stylistic variability, non-standard constructions, or domain-specific phrasing. As a result, grammatical normalization may induce different degrees of semantic and structural shift across text types. The correction process can be modeled as:

$$A_{\text{correct}}(s) = \arg \min_{s'' \in \mathcal{T}(s)} [L_{\text{syn}}(s'') + \gamma \cdot C_{\text{struct}}(s, s'')] \quad (2)$$

where s'' is the corrected output, $L_{\text{syn}}(\cdot)$ calculates syntactic compliance loss based on a dependency parser, and $C_{\text{struct}}(\cdot)$ quantifies syntactic structural changes using tree edit distance. $\gamma > 0$ controls the trade-off between grammatical normalization and preservation of original structure. This objective enforces syntactic normalization while constraining structural drift, highlighting implicit grammatical biases in LLMs.

Although LLM-generated texts demonstrate high grammatical correctness, they may contain implicit syntactic biases (e.g., excessive nested clauses, function word redundancy). Post-correction, these texts show minimal syntactic divergence from the original. Conversely, human writing often incorporates personalized grammatical habits (e.g., short sentence interleaving, flexible punctuation), leading to more pronounced syntactic modifications during correction.

Feature Reconstruction Agent: The feature reconstruction agent employs an abstract text processing strategy, which extracts core semantic features from input text and then generates new text based on these features.

$$A_{\text{feature}}(s) = \arg \max_{s''' \in \mathcal{T}(s)} [\alpha \cdot F_{\text{core}}(s, s''') - \beta \cdot R_{\text{surf}}(s, s''')] \quad (3)$$

where s''' is the reconstructed text, $F_{\text{core}}(\cdot)$ represents core semantic feature matching degree, and $R_{\text{surf}}(\cdot)$ measures surface-form reconstruction difference. $\alpha, \beta > 0$ weight semantic retention versus surface variation. This objective prioritizes high-level semantic fidelity over surface form, amplifying coherence gaps between human and LLM texts.

This approach exploits inherent differences between LLM-generated and human-written texts [28]: LLM outputs often show limited logical coherence and semantic depth, whereas human texts exhibit strong intrinsic consistency. Through feature extraction and reconstruction, similarity gaps emerge, with human-authored texts showing reduced similarity to reconstructed outputs and LLM-generated texts demonstrating heightened similarity, forming a discriminative pattern.

To illustrate the distinct behaviors of the three agents, Table 1 presents a concrete example using a human-written input from Reddit_dli5 subset in the HC3 dataset. The outputs highlight how each agent transforms the same source text according to its designated objective.

Table 1. Example outputs of the three agents on a human-written input.

Type	Text
Original	Depends when you start work. If you start at 7 am then 12–8 is a bad idea.
Text Revision	It is important to consider the timing of when you begin your workday. If your workday starts at 7 am, following a schedule of working from 12 pm to 8 pm may not be the most ideal choice.
Grammar Correction	It depends on when you start work. If you start at 7 am, then 12–8 is a bad idea.
Feature Reconstruction	Starting work at 7 am means that the time frame of 12–8 would not be ideal.

3.3. Similarity Scoring Mechanism

For a source text x_i and its revised version $\hat{x}_i$, we adopt BARTScore [29] as the core scoring mechanism. BARTScore is a metric based on a pre-trained sequence-to-sequence model, capable of effectively measuring semantic coverage and generation quality between the source and target texts.

Parameterized by θ , the core idea of BARTScore is to compute the log-probability of the target text x_i given the revised text $\hat{x}_i$ as the source. The specific steps are as follows:

1. Tokenization: The target text is tokenized into a sequence of tokens: $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$, where n is the sequence length.
2. Log-Probability Calculation: For each token x_{it} , the model computes its generation probability. The log-likelihood of all tokens is summed to derive the BARTScore. A higher score indicates greater similarity between the source and target texts, suggesting closer alignment with human writing patterns or logic. The BARTScore is formally defined as:

$$\text{BARTScore} = \sum_{t=1}^n \log p(x_{it} \mid x_{i(j < t)}, \hat{x}_i, \theta) \quad (4)$$

3. Normalization: To facilitate threshold selection, the score is normalized to the $[0, 1]$ interval using the expit function (i.e., Sigmoid function):

$$s_{\text{expit}} = \sigma(\text{BARTScore}) = \frac{1}{1 + e^{-\text{BARTScore}}} \quad (5)$$

3.4. Voting Decision Mechanism

Each agent independently computes a normalized similarity score $s_{\text{expit}}^{(k)} \in \{0, 1\}$ and converts it into a binary prediction using a threshold τ^* :

$$\hat{y}_i^{(k)} = \begin{cases} 1, & \text{if } s_{\text{expit}}^{(k)} \geq \tau^*, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

where 1 denotes LLM-generated text and 0 denotes human-written text. Notably, in this work, “zero-shot” refers to the absence of any supervised parameter fine-tuning on labeled training data. The backbone language models and perturbation modules remain fully frozen throughout all experiments. The decision threshold τ^* is applied only to convert continuous similarity scores into binary predictions and does not involve updating model parameters. Therefore, threshold calibration does not constitute supervised training and does not affect the zero-shot nature of the proposed framework.

The final prediction is obtained through a voting strategy. In the default Conservative (majority-voting) strategy, a text is classified as LLM-generated if at least two of the three agents predict it as such. Otherwise, it is classified as human-written. Formally,

$$\hat{y}_i = \begin{cases} 1, & \text{if } \sum_{k=1}^3 \hat{y}_i^{(k)} \geq 2, \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

This formulation clarifies that the decision mechanism is a standard majority-voting ensemble, reducing the influence of single-agent errors.

4. Experiments

4.1. Datasets

HC3 Dataset: : The HC3 corpus [30], collected by the SimpleAI team, consists of human-ChatGPT answer pairs for given questions across five distinct domains: Finance, Open_qa, Medicine, wiki_csai, and reddit_dli5. For the Finance and reddit_dli5 domains, we extracted 1000 human and LLM-generated texts each. The remaining domains were used in their entirety. This dataset enables evaluation across diverse content types, including financial reports, general knowledge queries, medical answers, encyclopedic articles, and online forum conversations, providing a comprehensive benchmark for LLM-generated text detection.

DetectGPT-Processed Datasets (DGPT): We also utilized datasets curated by the DetectGPT study [8], which include XSum, Reddit WritingPrompts, and PubMedQA. XSum consists of news articles and summaries, commonly used for tasks involving content generation and summarization. Reddit WritingPrompts contains stories generated in response to user prompts, reflecting creative writing scenarios. PubMedQA consists of long-form answers written by human experts in the medical domain. For all three datasets, machine-generated texts were obtained by prompting a large language model with the first 30 tokens of the human-written text, ensuring semantic alignment between human and generated samples.

CHEAT Datasets: The CHEAT dataset [31] includes three domains: Algorithm (A), Medicine (M), and Subject (S), drawn from an English summarization corpus. Each domain contains an equal number of human-authored and LLM-generated summaries, with 200 texts per category in our experiments. This dataset provides a controlled setting for evaluating detection methods in summarization tasks, where machine-generated texts often exhibit high fluency and semantic similarity to human-written texts. The

multi-domain design of CHEAT allows testing of cross-domain robustness and zero-shot generalization capabilities.

4.2. Baseline Methods

To validate our CMA-ZSD, we benchmark against baseline models and the SOTA methods spanning diverse paradigms from traditional machine learning to deep learning, establishing comprehensive evaluation benchmarks.

SDA [32]: A classical unsupervised deep network that learns feature representations for both domains via an autoencoder, followed by a classifier trained on source-domain labels to enable target-domain classification.

TextCNN [33]: A deep neural network that uses convolutional layers to extract contextual information from labeled source-domain data. The trained model is then applied to classification tasks in the target domain.

Bert [34]: Bert adapts to specific tasks through supervised fine-tuning. Specifically, it refines the pre-trained Bert model using source-domain data and applies the model for classification in the target domain.

Prompt [35]: Inserts predictions into template prompts with a label word space containing only class names, yielding classification outputs.

Spot [36]: This paper introduces a novel soft-prompt transfer method. Prompts are trained on one or more source tasks and then used to initialize prompts for the target task, enabling the classification of target-domain data.

ItSPT [37]: Proposes an iterative soft prompt tuning method for unsupervised domain adaptation, enhancing classification performance through iterative prompt refinement.

DetectGPT [8]: A zero-shot detector exploiting inherent curvature in LLM probability functions to detect machine-generated text.

RoBERTa-MPU [24]: A RoBERTa-based framework for LLM-generated text detection, enhancing short-text classification accuracy while maintaining long-text performance.

4.3. Implementation Details

In the main experiments, the perturbation operations of the three agents are implemented based on GPT-3.5-turbo. Specifically, text revision, grammar correction, and feature reconstruction are conducted through prompt-based interactions with GPT-3.5-turbo to generate corresponding text variants. The generation process of each agent is constrained by its objective function to ensure semantic preservation, syntactic correctness, and the retention of high-level semantic features during perturbation. All experiments are performed in a zero-shot setting, without any domain-specific fine-tuning of the generative model.

For each dataset, the decision threshold τ^* is determined using a held-out validation subset by maximizing the F1-score. The same threshold is applied across all agents within a dataset to ensure consistency. Thresholding is necessary because BARTScore-based similarity values are continuous, converting them into binary decisions requires selecting a cutoff that balances precision and recall. This validation-based selection ensures fair comparison across different voting strategies.

In addition to the default Conservative (majority-voting) strategy, we evaluate two alternative aggregation mechanisms to analyze the sensitivity of ensemble decisions. The first is Single-Agent Trigger (formerly “Extremely Lenient”), which refers to a text is classified as LLM-generated if any agent predicts it as such. This strategy maximizes recall but may increase false positives. The second is Unanimous Consensus (formerly “Strict Consensus”), which refers to a text is classified as LLM-generated only if all agents agree. This strategy maximizes precision but may increase false negatives. These variants are

evaluated to examine the bias-variance trade-off induced by different aggregation rules, the results are presented in following Section 4.7.

4.4. Main Experimental Results

In the experiments, we adopt Accuracy and F1-score as evaluation metrics. The experimental results presented in Table 2 show that our CMA-ZSD achieves higher accuracy than several domain-specific fine-tuned baselines on the CHEAT and DGPT datasets under out-of-distribution settings. This empirical observation suggests that the proposed perturbation-based collaborative framework exhibits strong cross-domain robustness.

We hypothesize that this advantage may stem from two factors. First, unlike fine-tuned detectors that may implicitly learn domain-specific artifacts, the proposed method relies on perturbation consistency across semantic, grammatical, and structural transformations, which may capture more generalizable linguistic signals. Second, the collaborative multi-agent design may reduce reliance on single-view features, thereby improving robustness to domain shifts. However, these explanations are inferential and should be interpreted as plausible mechanisms rather than experimentally isolated causal factors.

Traditional methods generally suffer from insufficient feature extraction capabilities. While SDA acquires cross-domain features through unsupervised pretraining, its auto-encoder architecture can only capture local statistical patterns and struggles to model the complex semantic discontinuity patterns in generated text. The convolutional kernel design of TextCNN is inherently limited to local context awareness, making it susceptible to being misled by the surface-level fluency of highly realistic text generated by large language models. Fine-tuning methods like BERT, although equipped with semantic understanding capabilities, rely heavily on domain-specific annotated data due to their supervised learning paradigm, resulting in a sharp decline in performance when applied to cross-domain generalization. In contrast, our method fully leverages the prior knowledge of language generation internalized during the pretraining of large language models. Through their self-attention mechanisms and deep semantic understanding capabilities, they can capture multi-dimensional features in generated text. This interpretation is consistent with the numerical results but requires further targeted analysis to establish causality.

The decision-making architectures of existing methods exhibit systemic deficiencies. Prompt and Spot employ fixed templates or static prompts, while DetectGPT's reliance on single-dimensional probability curvature analysis leads to limitations in semantically ambiguous scenarios. In contrast, this study innovatively constructs a multi-agent collaborative framework, achieving performance improvements through a three-tiered dynamic decision-making mechanism. This collaborative mechanism enables the system to maintain high accuracy while effectively reducing the risk of extreme decisions that traditional hard voting may cause.

Unsupervised domain adaptation methods like ItSPT require source domain data support, and the similarity of data distributions between the source and target domains constrains their performance. Supervised fine-tuning methods such as RoBERTa-MPU rely on domain-specific annotated data for parameter optimization, which not only increases the cost of data collection and annotation but also results in significant performance degradation in cross-domain scenarios. In contrast, our method employs a zero-shot learning framework that leverages the cross-domain knowledge transfer capabilities of large language models to achieve stable detection without any domain-specific data.

Table 2. Experimental results of our method against baseline methods. Bold indicates the best performance.

Method	CHEAT						DGPT						HC3									
	A		M		S		Xsum		Writing		PubMed		Wiki		Open		Medicine		Finance		Reddit	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SDA	70.44	72.33	67.74	67.73	67.81	66.15	62.89	67.47	67.78	60.11	58.67	62.32	68.62	70.68	67.82	62.24	68.56	68.13	71.80	67.42	67.80	64.66
TextCNN	75.75	72.51	82.33	79.30	81.05	79.82	63.41	62.37	68.09	65.14	65.79	69.11	72.57	70.60	79.58	75.83	86.52	88.42	82.76	80.78	84.77	83.19
Bert	72.05	70.35	74.21	76.28	68.94	63.78	57.14	60.16	56.67	52.15	59.26	58.76	66.41	68.39	68.75	66.82	82.50	80.12	82.67	82.01	82.03	80.51
Prompt	60.80	58.93	54.17	57.11	55.47	53.15	58.67	53.58	59.84	56.16	53.29	51.17	65.60	63.22	62.17	65.65	72.63	73.85	68.92	66.89	58.65	55.93
Spot	56.61	55.36	66.18	65.38	57.47	55.68	48.67	42.67	52.63	54.78	56.81	58.29	63.44	65.89	71.52	70.43	70.89	73.56	66.87	69.24	68.09	66.09
ItSPT	73.57	71.38	72.25	73.56	63.35	65.89	65.47	68.46	62.38	64.58	64.33	63.78	68.74	64.27	78.56	79.33	76.87	74.63	70.43	72.58	70.34	68.57
DetectGPT	61.50	61.68	59.50	60.15	59.00	57.16	72.00	74.68	75.00	74.58	55.00	59.57	85.50	85.38	61.50	62.79	91.00	90.27	82.00	80.33	64.50	62.17
RoBERTa-MPU	89.92	90.12	87.83	83.76	89.95	87.28	86.89	89.16	53.00	53.11	73.48	72.14	94.66	95.68	96.55	93.10	98.64	95.58	99.05	92.35	99.94	96.48
Our method	91.50	91.98	89.50	89.95	91.50	91.71	91.76	91.91	82.67	83.95	79.17	80.14	77.73	93.64	76.24	78.09	89.42	89.53	93.55	79.70	99.30	99.30

Although Table 2 reports results separately for each dataset, we further analyze performance differences across domains to better understand domain-dependent behavior. We observe that detection performance varies across domain types. In structured academic domains such as Medicine and Law, the proposed method achieves relatively higher accuracy. These domains typically exhibit standardized terminology and formal writing conventions, which may lead to more stable perturbation-based similarity patterns. In contrast, performance decreases in more stylistically diverse domains such as Writing and Blog text. Creative and informal writing often includes narrative variability, rhetorical devices, and flexible sentence structures, which may reduce the contrast between human-authored and LLM-generated responses under controlled transformations. Dialogue and QA-style datasets show intermediate performance. Although conversational text is less formally structured, LLM-generated responses often maintain high fluency and internal consistency, which still produces detectable perturbation signatures. We emphasize that these explanations are hypothesis-based interpretations derived from empirical trends rather than controlled causal experiments. Nonetheless, the results indicate that linguistic regularity and stylistic variability may influence the effectiveness of perturbation-based detection.

Experiments demonstrate that in out-of-distribution testing, our method achieves higher accuracy on both the CHEAT and DGPT datasets compared to domain-specific fine-tuning approaches, fully validating its generalization advantages. To further analyze the underlying reasons for this performance improvement, we identify several key contributing factors.

(1) Firstly, the core advantage of CMA-ZSD lies in its perturbation–response asymmetry modeling. Rather than learning surface-level distributional features tied to a specific domain, our method probes how a text responds to controlled semantic, grammatical, and structural transformations. LLM-generated texts tend to exhibit higher internal consistency and stylistic regularity due to their probabilistic decoding mechanisms, making them more stable under revision and reconstruction operations. In contrast, human-authored texts often contain individualized stylistic variations, implicit assumptions, and irregular syntactic constructions, which lead to larger semantic or structural shifts after perturbation. This asymmetric sensitivity forms a domain-invariant discriminative signal that remains effective under distribution shifts.

(2) Secondly, the zero-shot and model-agnostic nature of CMA-ZSD mitigates overfitting to generation-specific artifacts. Fine-tuned detectors often implicitly exploit generation fingerprints from specific models or sampling strategies. While this can yield high in-domain accuracy, it may reduce generalization to texts generated by unseen models. By avoiding parameter updates and relying instead on transformation-based analysis, CMA-ZSD leverages the intrinsic knowledge embedded in large language models (e.g., GPT-3.5-turbo) without specializing to any particular dataset, thereby preserving cross-domain transferability.

(3) Finally, the majority voting mechanism further amplifies generalization gains by reducing error correlation across agents. As shown in the ablation study, individual agents may exhibit domain-specific weaknesses; however, their error patterns are only partially overlapping. The ensemble decision process therefore reduces variance and prevents extreme misclassifications caused by single-agent bias. This collaborative structure contributes significantly to the stable performance observed on the CHEAT and DGPT benchmarks.

4.5. Ablation Study

To systematically analyze the contribution of individual agents and their synergistic interactions, we conducted ablation experiments comparing three dual-agent configurations against the complete three-agent ensemble.

In the Open_qa domain, both the Rev+Correct (76.87%) and Feat+Correct (76.75%) combinations slightly outperform the Triple-Agent ensemble (76.24%). This phenomenon stems from domain-specific error correlations among agents. The Feature agent exhibits the lowest individual accuracy (73.17%) in this domain, its error patterns show low overlap with the Correct agent. Their complementary errors enable effective mutual correction in the dual-agent setting. However, when all three agents participate in majority voting, the Revise agent (75.11%) occasionally aligns its misjudgments with the Feature agent on a small subset of samples, forming an unfavorable 2:1 voting outcome that overrides the Correct agent's accurate prediction. A similar pattern occurs in the Wiki_csai domain, where the underperforming Correct agent (71.38%) causes the Feat+Rev combination (78.15%) to surpass the full ensemble (77.73%) by 0.42%.

The ablation results indicate that individual agents exhibit domain-dependent sensitivity. In highly formal domains (e.g., Medicine and Law), the Grammar-Correcting Agent provides limited additional signal, as both human and LLM texts are already well-structured. In creative or blog-style domains, the Semantic Revision Agent may show reduced discriminative power due to stylistic variability. For short dialogue texts, the Feature Reconstruction Agent contributes less because structural transformation is limited. However, the full collaborative configuration consistently outperforms single-agent variants, suggesting that majority voting compensates for domain-specific weaknesses.

Although there are cases of local superiority, as shown in Table 3, the overall trends remain clear. The three-agent ensemble achieved the best performance in 9 out of 11 domains, surpassing the highest individual performance by approximately 2.18% on average. In the remaining two domains, it trailed the best dual-agent combination by only a narrow margin. This demonstrates that the three-agent ensemble strategy not only maintains competitive accuracy but also exhibits higher reliability and adaptability in practical applications.

Table 3. Dual-Agent Combinations vs. Triple-Agent Ensemble. Arrows indicate whether the performance is higher or lower compared to the Triple-Agent ensemble.

Dataset	Agent Combinations			
	Feature+Revise	Feature+Correct	Revise+Correct	Triple-Agent
CHEAT				
A	85.75 ↓−5.75	89.00 ↓−2.50	87.25 ↓−4.25	91.50
M	85.50 ↓−4.00	87.50 ↓−2.00	87.50 ↓−2.00	89.50
S	87.00 ↓−4.50	85.75 ↓−5.75	84.75 ↓−6.75	91.50
DGPT				
Xsum	86.52 ↓−5.24	83.15 ↓−8.61	81.65 ↓−10.11	91.76
Writing	74.67 ↓−8.00	81.67 ↓−1.00	75.67 ↓−7.00	82.67
PubMed	76.89 ↓−2.28	65.15 ↓−14.02	62.12 ↓−17.05	79.17
HC3				
Wiki_csai	78.15 ↑+0.42	77.43 ↓−0.30	76.54 ↓−1.19	77.73
Open_qa	74.52 ↓−1.72	76.75 ↑+0.51	76.87 ↑+0.63	76.24
Medicine	84.33 ↓−5.09	84.66 ↓−4.76	81.65 ↓−7.77	89.42
Finance	91.85 ↓−1.70	90.10 ↓−3.45	86.80 ↓−6.75	93.55
Reddit_dli5	98.20 ↓−1.10	97.25 ↓−2.05	96.85 ↓−2.45	99.30

4.6. Influence of LLMs

To evaluate the influence of different model architectures of LLMs in agent configurations on detection performance, we compared the cloud-based closed-source models gpt-3.5-turbo, claude-3.5-haiku, and gemini-2.0-flash, accessed via API, with the open-source model Llama-2-13b-chat-ms running in a locally deployed environment. The results shown in Figure 2 indicate that:

Our primary experimental model, gpt-3.5-turbo, achieved optimal performance across most datasets. For instance, on the Xsum dataset, its Voting accuracy (91.76%) was significantly higher than that of Llama (82.02%), Claude (87.27%), and Gemini (77.9%). Compared to the other three models, it demonstrated relatively stable detection accuracy.

Cloud-based closed-source models (gpt-3.5-turbo, claude-3.5-haiku, and gemini-2.0-flash) demonstrate superior detection performance, as they leverage large-scale training data and robust computational resources to generate text with higher semantic coherence and naturalness. Their outputs often retain distinct detectable features (e.g., specific linguistic patterns or statistical deviations), reducing false classification rates in detection models. While these models outperform Llama-2-13b-chat-ms in accuracy, our methodology maintains relatively small accuracy gaps across different LLMs.

Our voting mechanism preserved high accuracy regardless of the model architecture (e.g., Llama achieved 93.95% on the Medicine domain), demonstrating robust adaptability to diverse model structures.

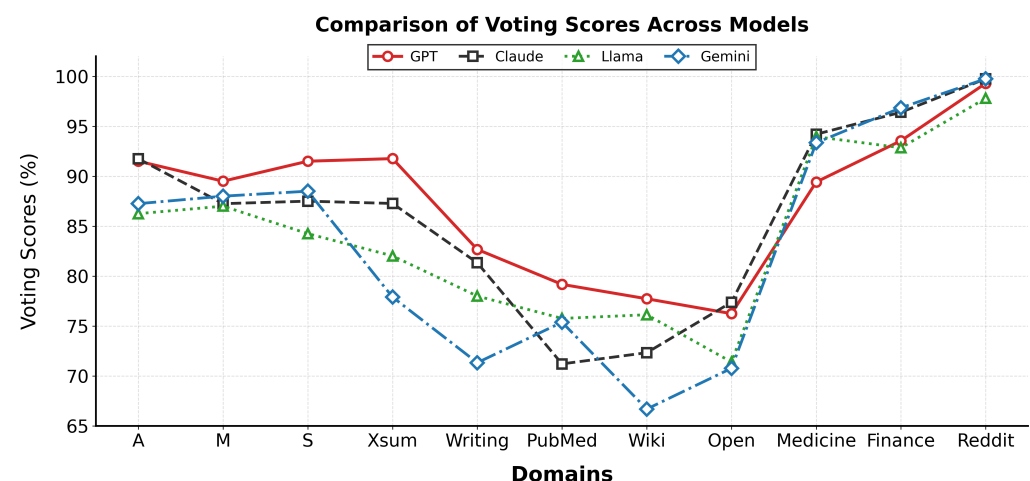


Figure 2. Voting Scores Under Various LLMs.

4.7. Different Voting Strategies

In the main experiment, we adopted the “conservative” strategy (i.e., when two or more agents determine the text to be generated by a large model, the final determination is that a large model generates it). To validate the effectiveness of this strategy, we further tested the performance of the “extremely lenient” and “strict consensus” strategies. The experimental results in Table 4 show that the conservative strategy performs best in balancing accuracy and robustness. In contrast, extreme strategies (such as extremely lenient and strict consensus) exhibit their respective limitations in different scenarios.

The “extremely lenient” strategy is characterized by lenient judgment. As long as one agent determines the text to be generated by a large model, the final result is considered to be generated by a large model. This strategy performs well on some datasets, for example, achieving an accuracy of 84.54% on the Medicine domain. However, its performance significantly drops on the Writing domain (only 50.67%).

The “strict consensus” strategy is characterized by strict judgment: all agents must unanimously determine the text to be generated by a large model for the final result to be

considered as such. This strategy shows stable performance on the Reddit_dli5 domain (96.50%), but its accuracy declines on the PubMed domain due to overly strict judgment (only 62.50%).

Table 4. Performance comparison (Accuracy %) of voting strategies.

Dataset	Domain	Strategy		
		Extremely Lenient	Strict Consensus	Conservative
CHEAT	A	75.75	85.25	91.50
	M	83.75	85.50	89.50
	S	84.25	83.00	91.50
DGPT	Xsum	76.78	79.78	91.76
	Writing	50.67	74.67	82.67
	PubMed	65.53	62.50	79.17
HC3	Wiki_csai	68.71	77.20	77.73
	Open_qa	70.72	75.95	76.24
	Medicine	84.54	80.61	89.42
	Finance	83.00	87.60	93.55
	Reddit_dli5	95.15	96.50	99.30

To further analyze the reasons behind the performance differences across datasets under different voting strategies, we examine the interaction between agent-level error patterns and domain-specific textual characteristics.

(1) Firstly, the “Extremely Lenient” strategy is highly sensitive to false positives. This strategy effectively maximizes recall for detecting generated text but may substantially reduce precision when individual agents exhibit domain-specific over-detection tendencies. In structured and fact-oriented domains such as Medicine, LLM-generated texts often display stronger stylistic regularity and semantic smoothness compared to human-authored answers. As a result, at least one agent can reliably detect perturbation stability patterns characteristic of machine-generated text, leading to improved detection performance (84.54%). However, in creative domains such as Writing, human-authored texts frequently resemble LLM outputs in terms of fluency, narrative coherence, and stylistic uniformity. In this case, individual agents are more likely to produce false positives. Under the “Extremely Lenient” rule, even a single erroneous prediction is sufficient to misclassify the sample, causing the accuracy to drop significantly (50.67%).

(2) Secondly, the “Strict Consensus” strategy is highly sensitive to false negatives. This approach improves precision but sacrifices recall. In domains where agent predictions are moderately correlated but not perfectly aligned (e.g., PubMed), small disagreements among agents prevent positive classification, leading to under-detection and lower accuracy. Conversely, in domains where LLM-generated texts exhibit strong and consistent perturbation stability, the three agents tend to agree more frequently, resulting in stable performance.

(3) Thirdly, the “Conservative” strategy achieves a balance between these two extremes by mitigating both single-agent overconfidence and unanimous under-detection. From an ensemble learning perspective, this strategy reduces variance without excessively increasing bias. In domains with heterogeneous linguistic structures, such as Writing or PubMed, majority voting effectively offsets domain-specific weaknesses of individual agents by leveraging partially decorrelated error patterns. This explains why the Conservative strategy consistently achieves the most stable performance across all datasets.

(4) Finally, the differences across datasets can also be attributed to textual genre characteristics. Creative writing domains contain higher lexical diversity, narrative flexibility, and stylistic overlap between human and machine texts, increasing ambiguity in

perturbation-based signals. In contrast, technical or knowledge-intensive domains exhibit more constrained discourse structures, making perturbation-response asymmetry more discriminative. Voting strategies that amplify or suppress individual agent signals therefore interact differently with each domain's intrinsic linguistic properties.

4.8. Comparison of Individual Agent Accuracy with Voting Strategy Results

In order to further validate the efficacy of the majority voting approach [38], Table 5 shows the performance differences between the individual detection accuracy of three independent agents and the ensemble voting strategy.

Table 5. Individual agent accuracy and ensemble voting results (Accuracy %).

Datasets	Domain	Agent			
		Feature	Revised	Corrected	CMA-ZSD
CHEAT	A	85.25	84.00	86.25	91.50
	M	84.50	86.50	87.75	89.50
	S	87.25	85.25	86.25	91.50
DGPT	Xsum	88.76	82.02	77.53	91.76
	Writing	81.33	75.67	51.00	82.67
	PubMed	75.00	68.94	63.26	79.17
HC3	Wiki_csai	75.71	76.54	71.38	77.73
	Open_qa	73.17	75.11	74.64	76.24
	Medicine	85.66	84.62	84.29	89.42
	Finance	92.70	88.70	82.75	93.55
	Reddit_dli5	97.00	96.45	97.50	99.30

The experimental results indicate that the ensemble voting strategy significantly outperforms the best performance of any single agent across all datasets. For example, in datasets CHEAT, CMA-ZSD consistently achieves high performance across different domains, respectively, which is an improvement of 1.75–5.25 percentage points over the best-performing single agent.

Majority voting effectively reduces the domain bias of a single method by integrating multi-agent features. In the Writing domain, the misjudgment tendency of individual agents (Corrected Accuracy of 51.00% vs. CMA-ZSD's 82.67%) is effectively mitigated through majority decision-making.

5. Conclusions

In this paper, we propose a collaborative multi-agent method for zero-shot LLM-generated text detection, which demonstrates superior performance across 11 domains. Through coordinated decision-making among three heterogeneous agents combined with a majority voting mechanism, the system effectively captures fundamental discrepancies in rewritten consistency, grammatical sensitivity, and semantic stability of generated texts, demonstrating the robustness of multi-agent frameworks against complex semantic features. Through zero-shot multi-agent collaboration without requiring annotated training data, this framework achieves a balanced trade-off between detection accuracy and cross-domain generalization capability, establishing a novel and practical paradigm for generated text detection.

Experimental results demonstrate that CMA-ZSD consistently outperforms traditional baselines and zero-shot statistical methods such as DetectGPT, and achieves competitive or superior performance compared to fine-tuned neural detectors such as RoBERTa-MPU in several out-of-distribution scenarios. Ablation studies confirm that the triple-agent collaboration (semantic revision, grammatical correction, and feature reconstruction) provides

complementary signals and achieves the best performance in most domains. Furthermore, comparative analysis of voting strategies shows that the Conservative majority-voting mechanism offers the most stable and balanced results, outperforming both the Extremely Lenient and Strict Consensus strategies.

In the future, we plan to extend our work in two directions. Firstly, we aim to explore the automatic feature representation learning method for LLM-generated text detection with a smaller model, the high quality of representations will help the constructed model beyond the hand-crafted model in detection tasks. Secondly, we will explore better methods for introducing PLMs, which will consider both the requirement of prior knowledge in PLMs and detection performance improvement.

Author Contributions: Conceptualization, G.S. and B.L.; methodology, B.L.; software, B.L. and Y.Z. (Yi Zhu); validation, Y.Z. (Yi Zhu) and J.Q.; formal analysis, G.S.; investigation, B.L.; resources, G.S.; data curation, Y.Z. (Ying Zhou); writing—original draft preparation, B.L. and Y.Z. (Ying Zhou); writing—review and editing, G.S., Y.Z. (Yi Zhu) and J.Q.; visualization, G.S.; supervision, Y.Z. (Yi Zhu); project administration, G.S.; funding acquisition, G.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is partially supported by the Open Project of Anhui Engineering Research Center for Intelligent Computing and Information Innovation (FYKFKT24049, ICII202508), the National Natural Science Foundation of China (62403412), the National Language Commission of China (ZDI145-71), the Key Research and Development Program of Jiangsu Province in China (BE2023315), Anhui Provincial Science and Technology Fortification Plan (Grant No. 202423k09020015), and the Open Project Program of Key Laboratory of Knowledge.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings will be available at <https://github.com/zhuyiYZU/CMA-ZSD> (accessed on 13 January 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhu, B.; Yuan, L.; Cui, G.; Chen, Y.; Fu, C.; He, B.; Deng, Y.; Liu, Z.; Sun, M.; Gu, M. Beat llms at their own game: Zero-shot llm-generated text detection via querying chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 7470–7483.
2. Souza, G.d.P.; Melo, G.; Schneider, D. Tailoring Treatment in the Age of AI: A Systematic Review of Large Language Models in Personalized Healthcare. *Informatics* **2025**, *12*, 113. [\[CrossRef\]](#)
3. Anghel, C.; Anghel, A.A.; Pecheanu, E.; Susnea, I.; Cocu, A.; Istrate, A. Multi-Model Dialectical Evaluation of LLM Reasoning Chains: A Structured Framework with Dual Scoring Agents. *Informatics* **2025**, *12*, 76. [\[CrossRef\]](#)
4. Khurana, D.; Koli, A.; Khatter, K.; Singh, S. Natural language processing: State of the art, current trends and challenges. *Multimed. Tools Appl.* **2023**, *82*, 3713–3744. [\[CrossRef\]](#)
5. Lamsiyah, S.; Ezzini, S.; El Mahdaouy, A.; Alami, H.; Benlahbib, A.; El Amrany, S.; Chafik, S.; Hammouchi, H. M-DAIGT: A Shared Task on Multi-Domain Detection of AI-Generated Text. *arXiv* **2025**, arXiv:2511.11340.
6. Gu, C.; Huang, C.; Zheng, X.; Chang, K.W.; Hsieh, C.J. Watermarking Pre-trained Language Models with Backdoor. *arXiv* **2022**, arXiv:2210.07543. [\[CrossRef\]](#)
7. Zhou, H.; Zhu, J.; Su, P.; Ye, K.; Yang, Y.; Gavioli-Akilagun, S.A.; Shi, C. Adadetctgpt: Adaptive detection of llm-generated text with statistical guarantees. *arXiv* **2025**, arXiv:2510.01268.
8. Mitchell, E.; Lee, Y.; Khazatsky, A.; Manning, C.D.; Finn, C. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the International Conference on Machine Learning, Honolulu, HI, USA, 23–29 July 2023*; JMLR: Norfolk, MA, USA, 2023; pp. 24950–24962.
9. Wang, S.; Cui, S.; Zhang, C.; Zhang, Z.; Wang, J.; Liu, T. Towards Persona-Oriented LLM-Generated Text Detection: Benchmark Dataset and Method. In *Proceedings of the International Conference on Artificial Neural Networks, Kaunas, Lithuania, 9–12 September 2025*; Springer: Berlin, Germany, 2024; pp. 352–367.

10. Gehrmann, S.; Strobelt, H.; Rush, A.M. Gltr: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Florence, Italy, 28 July–2 August 2019*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 111–116.
11. Adam, G.A.; Cui, A.; Thomas, E.; Napier, E.; Shmatko, N.; Schnell, J.; Tian, J.J.; Dronavalli, A.; Tian, E.; Lee, D. GPTZero: Robust Detection of LLM-Generated Texts. *arXiv* **2026**, arXiv:2602.13042. [[CrossRef](#)]
12. Yang, Z.; Zhao, G.; Wu, H. Watermarking for large language models: A survey. *Mathematics* **2025**, *13*, 1420. [[CrossRef](#)]
13. Liu, A.; Pan, L.; Hu, X.; Meng, S.; Wen, L. A Semantic Invariant Robust Watermark for Large Language Models. *arXiv* **2023**, arXiv:2310.06356. [[CrossRef](#)]
14. Kirchenbauer, J.; Geiping, J.; Wen, Y.; Katz, J.; Miers, I.; Goldstein, T. A Watermark for Large Language Models. *arXiv* **2023**, arXiv:2301.10226. [[CrossRef](#)]
15. Liu, Y.; Bu, Y. Adaptive text watermark for large language models. *arXiv* **2024**, arXiv:2401.13927. [[CrossRef](#)]
16. Dabiriaghdam, A.; Wang, L. Simmark: A robust sentence-level similarity-based watermarking algorithm for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Suzhou, China, 4–9 November 2025*; Association for Computational Linguistics: Suzhou, China, 2025; pp. 30785–30806.
17. Zhang, R.; Hussain, S.S.; Neekhara, P.; Koushanfar, F. {REMARK-LLM}: A robust and efficient watermarking framework for generative large language models. In *Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24), Philadelphia, PA, USA, 14–16 August 2024*; USENIX Association: Berkeley, CA, USA, 2024; pp. 1813–1830.
18. Mao, C.; Vondrick, C.; Wang, H.; Yang, J. Raidar: geneRative AI Detection viA Rewriting. *arXiv* **2024**, arXiv:2401.12970. [[CrossRef](#)]
19. Su, J.; Zhuo, T.Y.; Wang, D.; Nakov, P. DetectLLM: Leveraging Log Rank Information for Zero-Shot Detection of Machine-Generated Text. *arXiv* **2023**, arXiv:2306.05540. [[CrossRef](#)]
20. Wang, R.; Li, Q.; Xie, S. DetectGPT-SC: Improving detection of text generated by large language models through self-consistency with masked predictions. *arXiv* **2023**, arXiv:2310.14479.
21. Venkatraman, S.; Uchendu, A.; Lee, D. Gpt-who: An information density-based machine-generated text detector. In *Proceedings of the Findings of the Association for Computational Linguistics: NAACL, Mexico City, Mexico, 16–23 June 2024*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 103–115.
22. Aich, A.; Bhattacharya, S.; Parde, N. Demystifying neural fake news via linguistic feature-based interpretation. In *Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022*; International Committee on Computational Linguistics: Gyeongju, Republic of Korea, 2022; pp. 6586–6599.
23. Ippolito, D.; Duckworth, D.; Callison-Burch, C.; Eck, D. Automatic Detection of Generated Text is Easiest when Humans are Fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020*; Jurafsky, D., Chai, J., Schluter, N., Tetreault, J., Eds.; ACL Anthology: Stroudsburg, PA, USA, 2020; pp. 1808–1822. [[CrossRef](#)]
24. Tian, Y.; Chen, H.; Wang, X.; Bai, Z.; Zhang, Q.; Li, R.; Xu, C.; Wang, Y. Multiscale Positive-Unlabeled Detection of AI-Generated Texts. *arXiv* **2023**, arXiv:2305.18149. [[CrossRef](#)]
25. Koike, R.; Kaneko, M.; Okazaki, N. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024*; AAAI Press: Washington, DC, USA, 2024; Volume 38, pp. 21258–21266.
26. Zanutto, S.E.; Aroyehun, S. Human Variability vs. Machine Consistency: A Linguistic Analysis of Texts Generated by Humans and Large Language Models. *arXiv* **2024**, arXiv:2412.03025. [[CrossRef](#)]
27. Wu, J.; Zhan, R.; Wong, D.F.; Yang, S.; Liu, X.; Chao, L.S.; Zhang, M. Who Wrote This? The Key to Zero-Shot LLM-Generated Text Detection Is GECsScore. *arXiv* **2024**, arXiv:2405.04286. [[CrossRef](#)]
28. Fraser, K.C.; Dawkins, H.; Kiritchenko, S. Detecting AI-Generated Text: Factors Influencing Detectability with Current Methods. *arXiv* **2024**, arXiv:2406.15583. [[CrossRef](#)]
29. Yuan, W.; Neubig, G.; Liu, P. BARTScore: Evaluating Generated Text as Text Generation. *arXiv* **2021**, arXiv:2106.11520. [[CrossRef](#)]
30. Guo, B.; Zhang, X.; Wang, Z.; Jiang, M.; Nie, J.; Ding, Y.; Yue, J.; Wu, Y. How Close is ChatGPT to Human Experts? Comparison Corpus, Evaluation, and Detection. *arXiv* **2023**, arXiv:2301.07597. [[CrossRef](#)]
31. Yu, P.; Chen, J.; Feng, X.; Xia, Z. CHEAT: A Large-scale Dataset for Detecting ChatGPT-writtEn AbsTracts. *arXiv* **2023**, arXiv:2304.12008. [[CrossRef](#)]
32. Zhu, Y.; Wu, X.; Li, P.; Zhang, Y.; Hu, X. Transfer learning with deep manifold regularized auto-encoders. *Neurocomputing* **2019**, *369*, 145–154. [[CrossRef](#)]
33. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arXiv* **2014**, arXiv:1408.5882. [[CrossRef](#)]
34. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805. [[CrossRef](#)]
35. Ding, N.; Hu, S.; Zhao, W.; Chen, Y.; Liu, Z.; Zheng, H.T.; Sun, M. OpenPrompt: An Open-source Framework for Prompt-learning. *arXiv* **2021**, arXiv:2111.01998. [[CrossRef](#)]

36. Vu, T.; Lester, B.; Constant, N.; Al-Rfou, R.; Cer, D. SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer. *arXiv* **2021**, arXiv:2110.07904. [[CrossRef](#)]
37. Zhu, Y.; Wang, S.; Qiang, J.; Wu, X. Iterative Soft Prompt-Tuning for Unsupervised Domain Adaptation. *IEEE Trans. Knowl. Data Eng.* **2024**, *36*, 8580–8592. [[CrossRef](#)]
38. Chih-Yao Chen, J.; Saha, S.; Bansal, M. ReConcile: Round-Table Conference Improves Reasoning via Consensus among Diverse LLMs. *arXiv* **2023**, arXiv:2309.13007. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.