OSTAR: Optimized Statistical Text-classifier with Adversarial Resistance

Yuhan Yao[1,2], **Feifei Kou**[1,2*], **Lei Shi**[3], **Xiao Yang**[1], **Zhongbao Zhang**[1], **Suguo Zhu**[4]
Jiwei Zhang[1], **Lirong Qiu**[1,2], **Haisheng Li**[5]
[1] School of Computer Science (National Pilot School of Software Engineering), BUPT
[2] Key Laboratory of Trustworthy Distributed Computing and Service, BUPT, Ministry of Education
[3]State Key Laboratory of Media Convergence and Communication, CUC
[4]College of Computer Science and Technology, HDU
[5]School of Computer and Artificial Intelligence, BTBU
*Correspondence: koufeifei000@bupt.edu.cn

Abstract

The advancements in generative models and the real-world attack of machine-generated text(MGT) create a demand for more robust detection methods. The existing MGT detection methods for adversarial environments primarily consist of manually designed statistical-based methods and fine-tuned classifier-based approaches. Statistical-based methods extract intrinsic features but suffer from rigid decision boundaries vulnerable to adaptive attacks, while fine-tuned classifiers achieve outstanding performance at the cost of overfitting to superficial textual feature. We argue that the key to detection in current adversarial environments lies in how to extract intrinsic invariant features and ensure that the classifier possesses dynamic adaptability. In that case, we propose **OSTAR**, a novel MGT detection framework designed for adversarial environments which composed of a statistical enhanced classifier and a Multi-Faceted Contrastive Learning(MFCL). In the classifier aspect, our Multi-Dimensional Statistical Profiling (MDSP) module extracts intrinsic difference between human and machine texts, complementing classifiers with useful stable features. In the model optimization aspect, the MFCL strategy enhances robustness by contrasting feature variations before and after text attacks, jointly optimizing statistical feature mapping and baseline pre-trained models. Experimental results on three public datasets under various adversarial scenarios demonstrate that our framework outperforms existing MGT detection methods, achieving state-of-the-art performance and robust against attacks.The code is available at https://github.com/BUPT-SN/OSTAR.

1 Introduction

With the remarkable and rapid progress in Large Language Models (LLMs) [1, 2], the quality of machine-generated text has gradually achieved a level that is increasingly comparable to human-authored content. However, the widespread proliferation of such text substantially risks amplifying the dissemination of unreliable or misleading information and diminishing [3, 4, 5] the creative motivation of human authors. As clearly shown in Figure 1(a), which illustrates a real-world detection scenario, MGT texts are often attacked to evade detection, critically challenging the robustness of existing detection methods [6, 7, 8].Therefore, developing reliable detection methods that can robustly distinguish MGT from human-authored content has become a crucial task in societal research.

As shown in Figure 1(b1) and Figure 1(b2), the current MGT detection methods for adversarial environments can be categorized into two approaches: statistical-based methods and classifier-based

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

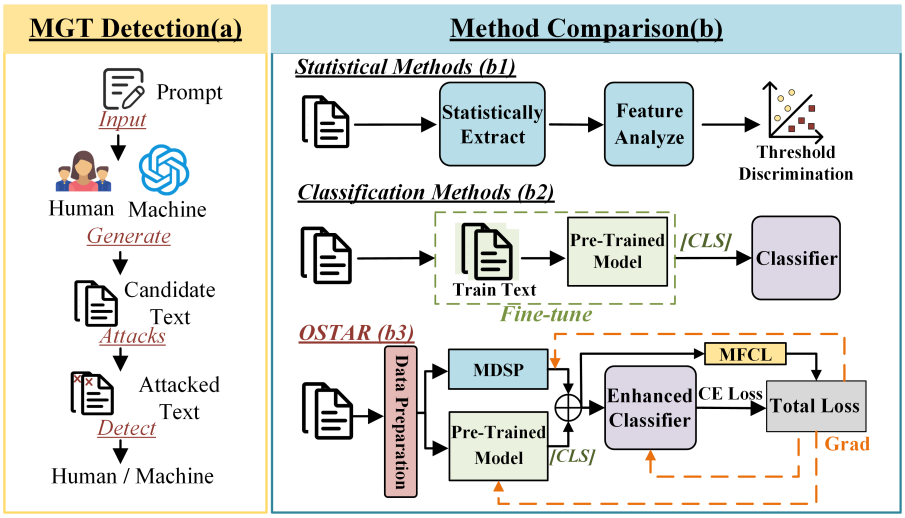

Figure 1: The real-world MGT detection task explanation and the difference between OSTAR and previous detection methods.

methods. While statistical-based methods[9, 10, 11] demonstrate strong zero-shot generalizability across LLMs through intrinsic features (e.g., n-gram distributions, syntactic anomalies), their detection performance is constrained by predefined static thresholds that fail to dynamically adapt to distribution shifts in textual feature patterns under adversarial attacks like lexical substitution or style transfer, leading to significant accuracy degradation. The classifier-based methods[12, 13] rely on pre-trained models (e.g., BERT[14]) to extract deep semantic features through loss function optimization. Although achieving high accuracy on specific datasets, the learned feature representations carry overfitting risks: models tend to capture superficial correlation patterns in training data (e.g., domain-specific sentence structures) rather than the intrinsic differences between human and machine-generated texts. This results in unstable feature representations, leading to a sharp decline in detection performance when distributional discrepancies exist between testing and training data. Collectively, these limitations highlight critical gaps in existing detection paradigms: statistical methods lack adaptation to attacks, classifier-based approaches struggle to capture the intrinsic invariant feature differences between human and machine texts under adversarial attacks.

Given the limitations of current methods in addressing textual attack, single-category detection methods now struggle to achieve high-precision MGT detection in adversarial environments. Our observations reveal that classifier-based methods often demonstrate superior performance on individual datasets, yet exhibit significant performance degradation when confronted with attacked texts. In contrast, although statistics-based detection methods show relatively weaker baseline metrics, certain textual feature statistics (such as Lexical Diversity and Readability) maintain relatively stable deviations between attacked and original texts. This characteristic could serve as an effective mechanism to enhance the robustness of classifier-based methods when operating in adversarial environments. The classifier-based methods can compensate for the shortcomings of statistical methods in feature dynamic adaptation caused by rigid thresholds, as they are capable of developing classification abilities tailored to specific datasets through loss function optimization.

Based on the respective limitations and potential complementarity of the statistical-based methods and classifier-based methods, this paper proposes the **O**ptimized **S**tatistical **T**ext-classifier with **A**dversarial **R**esistance (OSTAR) framework shown in Figure 1(b3). To improve the stability of detection, we design the Statistically Enhanced Classifier, which captures the intrinsic text statistical features thereby enhancing classifier performance. Specifically, we manually design Multi-Dimensional Statistics Profiling(MDSP) to extract and analyze statistical features of texts. These analyzed statistical features are then concatenated with the CLS embeddings generated by pre-trained models to form an enhanced classifier. To enable dynamic feature adaptation in adversarial environments, we categorize attacks into Perturbation and Paraphrases based on whether machine rewriting is involved, distinguishing them by their impact on text characteristics. Guided by this perspective, we design Multi-Feature Contrastive Learning (MFCL). This categorization effectively differentiates attack types by their intrinsic properties, and MFCL significantly enhances the robustness of our method in

adversarial scenarios. Extensive experiments across several datasets and adversarial environments consistently demonstrate that our proposed method outperforms previous approaches, establishing new state-of-the-art performance.

The main contributions of our work are as follows:

- We propose a novel MGT detection framework, OSTAR, which captures the intrinsic differences between machine-generated text and human-authored text and enables dynamic adaptation of detection. To our knowledge, this is the first work to utilize statistical features to guide classifier-based method, enhancing its robustness in adversarial environments.
- We manually designed the MDSP for intrinsic textual characteristics, which effectively captures intrinsic, relatively stable textual multi-dimensional statistical features across diverse scenarios. This mechanism provides a solid foundation for classifier-based methods.
- We categorize attacks into perturbations and paraphrases based on the intrinsic characteristics of their impacts, and design MFCL to comprehensively capture the manifold adversarial effects on text from multiple perspectives, significantly enhancing the overall robustness of our method.
- Through extensive evaluations on three public datasets under diverse adversarial scenarios, OSTAR achieves superior performance and robustness compared to state-of-the-art MGT detection methods.

2 Related Work

2.1 Machine-generated Text Detection

With the rapid development of LLMs, machine-generated texts, also known as AI-generated texts or Deepfake texts, have made it increasingly difficult for people to distinguish them from human-authored texts [15]. Nowadays, common detection methods can be categorized into three types:watermark-based [16], statistics-based and classifier-based. The watermark-based methods achieve detection of machine-generated text by embedding subtle watermark during the text generation phase and subsequently detecting the presence of watermarks in the generated content [17, 18, 19, 20, 21, 22]. Statistics-based methods often focus on identifying the inherent characteristic differences between human-authored and machine-generated texts by establishing thresholds for differentiation, demonstrating applicability across various models[10, 23, 24, 25, 11]. For example, Eduard[25] proposed distinguishing human and machine texts by analyzing differences in the intrinsic dimensionality of their generated embeddings, achieving stable detection performance across multiple cross-domain and cross-model scenarios. Classifier-based methods can be viewed as a binary classification task, where the detector is typically trained on datasets generated by the target LLM to achieve high-performance detection for the target generative model[26, 27, 28, 29, 30]. For example, SimpleAI[29] fine-tunes RoBERTa, removes biased training data for better generalization, and adds sentence-level data to capture local features. Current MGT detection methods excel on clean data but falter against adversarial attacks. However, many studies[31, 8, 32, 33, 6, 34] have indicated that current MGT detectors exhibit vulnerabilities in real-world detection scenarios, facing challenges to their robustness when subjected to various attacks and adversarial paraphrasing from other pre-trained models. Therefore, maintaining detection robustness in adversarial environments remains a major challenge in current MGT detection tasks.

2.2 Contrastive Learning

Currently, in the field of natural language processing, contrastive learning methods can effectively enhance the robustness of frameworks against adversarial attacks[35, 36, 37, 38, 30, 39, 27, 40]. For example, PairCFR[37] enhances the generalization performance of natural language processing tasks when handling counterfactually augmented data by promoting global feature alignment through contrastive learning; DeTeCtive[38] boosts MGT detection generalization through multi-level contrastive learning that identifies cross-sample author style gaps under out-of-distribution task; CoCo[30] tackles sparse training data via contrastive learning, achieving superior performance with minimal training datasets; PESCO[39] addresses the cold-start problem in Zero-Shot text classification by dynamically optimizing document-label matching through a self-training loop of contrastive learning.

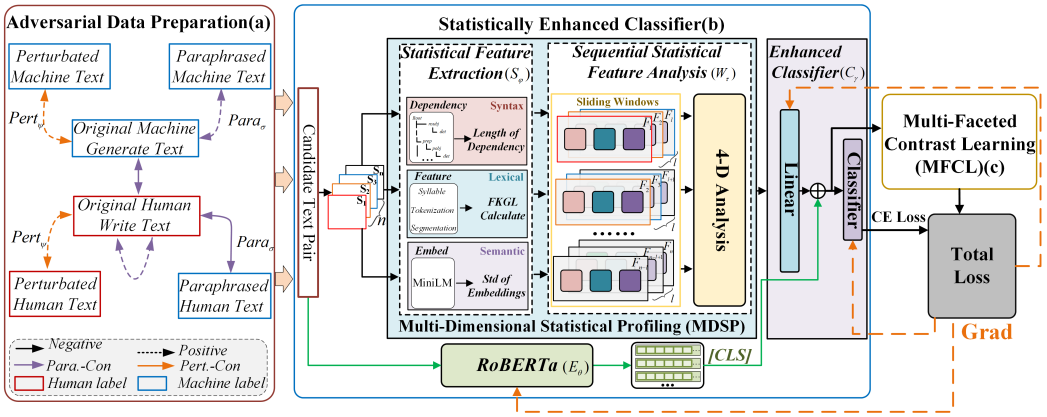

Figure 2: An overview of our OSTAR framework. (a) Adversarial data generation and contrastive learning pair construction (b) Enhancing classifier performance via MDSP (c) Performing contrastive learning and computing loss through MFCL

Existing research demonstrates that contrastive learning exhibits broad applicability and effectively enhances model performance in natural language processing, particularly for MGT classification.

3 OSTAR: Framework and Algorithms

We start this section by giving an overview the framework of OSTAR. Then, we detail the specific methods for each step in Sections 3.1 to 3.3. Finally, we will summarize the entire training process into an algorithmic procedure in Section 3.4.

Abstract Methodology As shown in Figure 2, during the training stage, our OSTAR consists of three parts as follows:

- **Part a (Adversarial Data Preparation):** Prior to training, we pre-process the original dataset O to construct the contrastive pairs. Specifically, we apply Perturbation Source$Pert_\psi$ to generate perturbation pairs and Paraphraser $Para_\sigma$ to produce paraphrase pairs.
- **Part b (Statistically Enhanced Classifier):** During the initial stage of the training phase, statistical feature extraction performed on each sentence using Statistical Feature Extraction S_φ. Once the S_φ has processed the entire text, Sequential Statistical Feature Analysis W_τ will analyze the intrinsic statistical feature with a sliding windows of length l and the 4-D Analysis method. Then, the outputs of W_τ are projected and concatenated with the CLS token from Pre-Trained Model (RoBERTa is used in this paper) E_θ for Enhanced Classifier C_γ to classify.
- **Part c (Multi-Faceted Contrast Learning):** In MFCL, perturbation contrastive learning and paraphrase triplet contrastive learning are designed for adversarial environments, optimized with updating τ and γ.

3.1 Adversarial Data Preparation

In terms of adversarial data preparation, as shown in Figure 2(a), we categorize training texts into two types: original human-authored and original machine-generated texts, and texts that may be encountered during detection (divided into Perturbation and Paraphrase). This design is based on the rationale that paraphrases often lead to changes in text ownership, such as transforming text from being generated by LLM to being written by $Para_\sigma$, or converting human-written text into machine-generated content. In contrast, perturbations do not involve ownership alteration but can affect text recognition accuracy. Therefore, we preprocessed the training set by $Pert_\psi$ and $Para_\sigma$, then assigning positive and negative sample pairs according to Figure 2(a), and dynamically constructing the dataset in each epoch.

3.2 Statistically Enhanced Classifier

The Statistically Enhanced Classifier is composed of Multi-Dimensional Statistical Profiling in section 3.2.1 and the Enhanced Classifier with a Pre-Trained Model in section 3.2.2.

3.2.1 Multi-Dimensional Statistical Profiling (MDSP)

As shown in Figure 2(b), MDSP consists of two components: Statistical Feature Extraction and Sequential Statistical Feature Analysis.

Statistical Feature Extraction(S_φ) With the increasingly powerful generative capabilities, relying solely on surface-level statistical features of text has become insufficient for robust MGT detection. While machine-generated texts exhibit certain statistical similarity with human-authored texts in surface patterns, they still demonstrate discernible differences in intrinsic statistical characteristics such as variations in sentence complexity, lexical diversity shifts, and semantic coherence dynamics within a document.

In this paper, we focus on extracting statistical features from three dimensions—syntactic, lexical, and semantic—by quantifying metrics such as the maximum dependency distance per sentence (syntax), the Flesch-Kincaid Grade Level (lexical complexity), and the variance across dimensions in semantic embedding projections (semantic coherence), thereby capturing multi-level discriminative patterns between human-authored and machine-generated texts, the three-dimensional features extraction can be formally formulated as follows:

$$\Phi(s) = \left[\underbrace{\max_{t \in s} \text{head}(t)}_{\text{Syntax}}, \underbrace{\mathcal{F}_{\text{FK}}(s)}_{\text{Lexical}}, \underbrace{1 - \frac{1}{n-1} \sum_{i=1}^{n-1} \cos(\mathbf{e}_i, \mathbf{e}_{i+1})}_{\text{Semantic}} \right] \tag{1}$$

where, s represents a single sentence in an article, t denotes the tokens in sentence s, $\text{head}(t)$ is the head index of token t, $\max \text{head}(\cdot)$ represents Maximum dependency distance in parse tree, $\mathcal{F}_{\text{FK}}()$ is the Standard Flesch-Kincaid Grade Level formula, $\{\mathbf{e}_1, ..., \mathbf{e}_n\}$ denotes the sentence embeddings from MiniLM, $\cos(\cdot)$ is inter-sentence cosine similarities.

Sequential Statistical Feature Analysis(W_τ) When performing statistical feature analysis, we employ a sliding window of length l with a step size t to segment the text. For each feature dimension $d \in \{\text{Syntax}, \text{Lexical}, \text{Semantic}\}$, we design a 4-D Analysis module to analyze the statical feature and use a 4D-feature vector $\mathbf{F}_d \in \mathbb{R}^4$ to store the outputs, aggregating four key measures: mean value (μ_d), standard deviation (σ_d), autocorrelation coefficient (ρ_d), and range (R_d). The process is formally formulated as follows:

$$\mathbf{F}_d = [\mu_d, \sigma_d, \rho_d, R_d] = \frac{1}{N} \sum_{k=1}^{N} \begin{cases} \mu_d^{(k)} = \frac{1}{|W^k|} \sum_{y \in W^k} y \\ \sigma_d^{(k)} = \sqrt{\frac{1}{|W^k|-1} \sum_{y \in W^k} (y - \mu_d^{(k)})^2} \\ \rho_d^{(k)} = \text{autocorr}(W^k) \\ R_d^{(k)} = \max(W^k) - \min(W^k) \end{cases} \tag{2}$$

where W^k denotes the sequential statistical features of k-th sliding window, y represents the statistical feature of a sentence within the window W^k.

For an input text sequence $S = \{s_1, s_2, ..., s_n\}$ with n sentences, we compute window-based statistics across three linguistic dimensions (syntax, lexical, semantic) as follows:

$$\mathbf{T}_S^{\text{raw}} = \bigoplus_{d \in \{Syn, Lex, Sem\}} \left(\frac{1}{|\mathcal{W}_{(d)}|} \sum_{W^k \in \mathcal{W}_{(d)}} \Phi(W^k) \right) \in \mathbb{R}^{12} \tag{3}$$

where, $\mathbf{T}_S^{\text{raw}} \in \mathbb{R}^{12}$ denotes Raw temporal feature vector formed by concatenating window statistics across three dimensions, $\phi(W^k)$ denotes the k-th sliding window for $\mu^{(k)}, \sigma^{(k)}, \rho^{(k)}, R^{(k)}$, $\mathcal{W}_{(d)} = \{W_{(d)}^1, \ldots, W_{(d)}^m\}$ represents the Sliding windows for dimension d.

Algorithm 1 OSTAR train process

Require: Original dataset $\mathcal{O} = (\mathcal{H}, \mathcal{M})$, Perturbation Source $Pert_\psi$, Paraphraser $Para_\sigma$
Ensure: Trained parameters τ (Analyzer), γ (Classifier), θ(Pre-Trained Model)

Frozen Components:
$S_\varphi, Para_\sigma, Pert_\psi$
Trainable: $W_\tau, C_\gamma, E_\theta$
Adversarial Data Preparation:
Generate $\tilde{\mathcal{O}}_{Pert} \leftarrow Pert_\psi(\mathcal{O})$ ▷ Perturbation pairs
Generate $\tilde{\mathcal{O}}_{Para} \leftarrow Para_\sigma(\mathcal{O})$ ▷ Paraphrase pairs
Build $\mathcal{D} \leftarrow \mathcal{O} \cup \tilde{\mathcal{O}}_R \cup \tilde{\mathcal{O}}_A$
for epoch $= 1$ **to** N **do**
  for batch $(x, \tilde{x}_{Pert}, \tilde{x}_{Para}) \sim \mathcal{D}$ **do**
    Statistical Feature Extraction:
    for each sentence s_i in x **do**
      Extract $\Phi(s_i) = [\max \text{dep}(s_i), \mathcal{F}_{FK}(s_i), 1 - \frac{1}{n-1} \sum \cos(e_i, e_{i+1})]$
    end for
    Sequential Statistical Feature Analysis:
    Apply sliding window ($l = 3$, step=1) on Φ sequence
    Compute window 4-D stats: $\mathbf{T}_x^{\text{raw}} = \frac{1}{m} \sum_{k=1}^{m} [\mu_k, \sigma_k, \rho_k, R_k]$
    Project: $\mathbf{T}_x^{\text{proj}} = \tanh(W_\tau \mathbf{T}_x^{\text{raw}} + b_t)$
    Statistical Enhance:
    $h_x = E_\theta(x)^{[\text{CLS}]}$ ▷ RoBERTa
    $f_x = \text{concat}(h_x, \mathbf{T}_x^{\text{proj}})$ ▷ 832-dim
    Multi-Faceted Learning:
    Compute Para-contrast loss: $\mathcal{L}_{\text{Para}} \leftarrow \log \frac{e^{s_p/\tau}}{e^{s_p/\tau} + \sum_n e^{s_n/\tau}}$
    Compute Pert-contrast loss: $\mathcal{L}_{\text{Pert}} \leftarrow \beta_a r S_a$
    Parameter Update:
    $\tau, \gamma, \theta - \eta \nabla((1 - \lambda) \cdot \mathcal{L}_{\text{CE}} + (\lambda_1 \cdot \mathcal{L}_{\text{Para}} + \lambda_2 \cdot \mathcal{L}_{\text{Pert}}))$
  end for
end for
return $\tau^*, \gamma^*, \theta^*$

3.2.2 Enhanced Classifier(C_γ)

The output $\mathbf{T}_S^{\text{raw}}$ of Multi-Dimensional Statistical Profiling is used to enhance classifier-based method by projection, the formula is as follows:

$$\mathbf{T}_S^{\text{proj}} = \tanh(\mathbf{W}_t \cdot \mathbf{T}_S^{\text{raw}} + \mathbf{b}_t) \in \mathbb{R}^{64} \quad \text{where } \mathbf{W}_t \in \mathbb{R}^{64 \times 12} \tag{4}$$

where, $\mathbf{T}_S^{\text{proj}} \in \mathbb{R}^{64}$ represents the projected features after nonlinear transformation, $\mathbf{W}_t \in \mathbb{R}^{64 \times 12}$ denotes the learnable projection matrix mapping raw features to latent space, $\tanh(\cdot)$ is the hyperbolic tangent activation function constraining values to [-1, 1].

Then the E_θ encodes the input text S , and the resultant 768-dimensional CLS embedding from the last layer is concatenated with the projected statistical features for classification.

3.3 Multi-Faceted Contrast Learning(MFCL)

MFCL can be divided to Paraphrase Contrastive Learning(Para-contrast) and Perturbation Contrastive Learning(Pert-contrast). In Para-contrast, anchors are defined as H(original human write text) and M(original machine generate text), where the positive and negative samples vary based on the anchor type: when the anchor is H, positive samples consist of other original human texts while negative samples are $\tilde{H}_{para}$, whereas for machine-generated anchors (M), positive samples are $\tilde{M}_{para}$ and negative samples include any human H. In Pert-contrast, only positive samples—corresponding to adversarially attacked versions of the specified text ($\tilde{H}_{pert}$ or $\tilde{M}_{pert}$) are utilized. The Multi-Faceted

Contrast can be formulated as follows:

$$\mathcal{L}_{\text{MFCL}} = \lambda_1 \cdot \underbrace{\sum_{i=1}^{M} \sum_{p \in \mathcal{P}(i)} \log \frac{e^{S_{ip}/\tau}}{\sum_{p' \in \mathcal{P}(i)} e^{S_{ip'}/\tau} + \sum_{n \in \mathcal{N}(i)} e^{S_{in}/\tau}}}_{\mathcal{L}_{\text{Para}}} + \lambda_2 \cdot \underbrace{\sum_{a \in \mathcal{A}(i)} \beta_{ia} r S_{ia}}_{\mathcal{L}_{\text{Pert}}} \quad (5)$$

where, $p \in \mathcal{P}(i)$ and $n \in \mathcal{N}(i)$ denote the positive and negative sample sets for anchor i, S_{ip} and S_{in} are similarity scores between anchor i and its positive/negative samples, scaled by temperature τ to sharpen or soften the contrastive probability distribution; $\mathcal{A}(i)$ is adversarially perturbed sample sets, S_{ia} denotes the is the similarity scores between anchor and perturbed samples, weighted by attack impact ratio β_{ia} and scaled by regularization coefficient r, λ_1 and λ_2 denote the weighting coefficients for $\mathcal{L}_{\text{Para}}$ (paraphrase loss) and $\mathcal{L}_{\text{Pert}}$ (perturbation loss), respectively.

The total loss function is composed of the Cross-Entropy $\mathcal{L}_{\text{CE}}$ loss and $\mathcal{L}_{\text{MFCL}}$, formulated as:

$$\mathcal{L}_{\text{total}} = (1 - \lambda) \cdot \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{MFCL}} \quad (6)$$

where λ are weighting coefficients.

3.4 OSTAR Algorithm

The overall training process of OSTAR is summarized in Algorithm 1. For a given text, OSTAR extracts its statistical features as supplementary information and concatenates them with embeddings generated by a pre-trained model. These representations are then aligned with Multi-Faceted Contrastive pairs to enhance the model's robustness against attacks.

4 Experiments

4.1 Experiment Setup

Datasets and Real-world Attacks In this study, we employed three widely-used and moderately challenging datasets: CheckGPT[41], HC3[29], and a cross-domain dataset generated by GLM-130B from the DeepFake[33]. CheckGPT comprises 900,000 multi-domain samples (e.g.news, reviews and articles) generated by ChatGPT[1] using diverse prompts. HC3 is composed of question-answer pairs, where each question includes at least one human-written response and one machine-generated response, focusing on open-ended questions across domains such as finance and medicine. DeepFake contains cross-domain texts generated by various LLMs. To capture real-world MGT diversity, we employ GLM-130B[42] as the representative dataset. Adversarial attacks are categorized into perturbation and paraphrase attacks. For perturbation, the specific 9 perturbations adopted are as follows. For the paraphrase, we employ DIPPER[6] to modify the text. More detailed dataset construction is shown in appendix C.

- **Character-level Perturbations**:
  - Space Insertion: Introduce extraneous whitespace within words (e.g., "hel␣lo")
  - Punctuation Removal: Delete commas, periods, etc. (e.g., "Hello, world!" → "Hello world")
  - Initial Character Case Alteration: Randomize capitalization of word-initial letters (e.g., "apple" → "Apple")
  - Word Merging: Concatenate adjacent words (e.g., "new␣york" → "newyork")
- **Word-level Perturbations**:
  - Keyboard Typos: Simulate typographical errors via adjacent key substitutions (e.g., "house" → "hjuse")
  - Character WordCase: Randomly alter the case of letters within words (e.g., "example" → "ExAmPlE")
  - Spelling Errors: Insert phonetically plausible misspellings (e.g., "because" → "becuz")
  - Adverb Insertion: Add semantically redundant adverbs within sentences (e.g., "He ran quickly" → "He ran *extremely* quickly")

Table 1: Detection performance comparison under original datasets. The best-performing data under each metric has been bolded. Due to dataset balancing, the values of accuracy, recall, and F1 score will be relatively close when the model is well-trained and the architecture is stable.

Methods	DeepFake			CheckGPT			HC3		
	ACC	Recall	F1	ACC	Recall	F1	ACC	Recall	F1
GPT-2	87.29	90.58	88.04	81.92	83.01	80.74	90.86	90.75	89.41
RoBERTa	91.68	91.57	91.66	88.77	87.82	88.78	94.32	94.31	94.32
CoCo	88.03	89.59	87.58	84.55	84.90	85.97	98.42	99.31	98.50
RADAR	55.49	55.49	58.05	63.04	63.26	63.01	89.57	89.57	90.39
Watermark	86.21	90.45	88.91	75.69	**97.06**	72.26	94.88	94.75	95.13
Binoculars	78.22	82.41	76.39	86.90	89.74	87.12	92.44	95.13	91.95
PECOLA	86.29	86.19	86.29	84.58	84.96	84.51	99.23	99.25	99.24
OSTAR	**91.94**	**92.38**	**92.36**	**90.37**	90.12	**90.23**	**99.55**	**99.78**	**99.55**

  - Adverb Append: Attach an additional adverb at the end of sentences (e.g., "The task is done." → "The task is done. *perfectly*")

- **Sentence-level Perturbations**:
  - Sentence Reversal: Invert word order (e.g., "This is a test." → "Test a is this.")
  - Sentence Repetition: Duplicate clauses/phrases (e.g., "I agree. I agree.")

Evaluation Metric To ensure a comprehensive and systematic evaluation of our work, we adopted widely recognized metrics for binary classification task —Accuracy (ACC), Recall, and F1-score (F1)—to evaluate model performance.

Comparison Methods We compare our method with classifier-based state-of-the-art MGT detectors, including: **GPT-2** [43] and **RoBERTa** [44] fine-tuned as binary classifiers (124M/110M parameters); **CoCo** [30] employing contrastive learning with coherence graphs; **RADAR** [26] using adversarial paraphrases for robustness; **Binoculars** [11] leveraging cross-model probability divergence for zero-shot detection; the watermark-based method **Watermark** [20] for reference. Statistic-based methods are excluded due to limited adversarial robustness; **PECOLA** [22] enhancing robustness via core-term perturbation.

To ensure experimental fairness, we utilized data-augmented datasets for training all classifier-based contrastive methods, with the exception of RADAR (whose code was unavailable, necessitating the use of their open-source model). For non-trainable approaches — including statistical-based methods (e.g., Binoculars) and watermark-based techniques (e.g., Watermark) — no training phase was implemented, as their detection mechanisms rely on pre-defined heuristics rather than learnable parameters. All experiments strictly adhered to the original implementations' default training configurations. For methods requiring specialized data preprocessing pipelines (e.g., CoCo), we faithfully executed their prescribed preprocessing steps as outlined in their respective methodologies.

4.2 Performance Evaluation

Evaluation on Original Datasets The detection evaluation on the original dataset is shown in Table 1. OSTAR achieved average Acc, Recall and F1 of 93.95%, 94.09%, and 94.04% across the three datasets, outperforming the fine-tuned RoBERTa baseline by average improvements of 2.81% (ACC), 2.85% (Recall), and 2.97% (F1). Moreover, compared to state-of-the-art methods on the MGT detection task, OSTAR attained at least 1.32%, 1.42%, and 1.05% enhancements in ACC, Recall, and F1, respectively, while achieving the best performance across all three datasets. These results demonstrate that our method outperforms the baseline approaches in average detection quality under non-attack scenarios.

Furthermore, as evidenced in Table 1, the incorporation of statistical feature analysis in our method consistently enhances the detection performance of the baseline RoBERTa model across all three datasets, empirically validating the efficacy of our feature analysis approach. Notably, the DeepFake dataset poses the greatest challenge for all models. We attribute this to the fact that GLM-130B—used to generate DeepFake texts—represents a model not adequately considered by conventional pre-

Table 2: Detection performance of OSTAR and the baselines on datasets with attack. We adopt F1-score as the evaluation metric when facing attack, where we categorize attack into Perturbation(Pert.) and paraphrases(Para.). The best-performing data under each metric has been highlighted in bold.

Methods	DeepFake			CheckGPT			HC3		
	Ori.	Pert.	Para.	Ori.	Pert.	Para.	Ori.	Attack	Para.
GPT-2	88.04	74.23	73.41	80.74	70.58	72.56	89.41	82.72	81.63
RoBERTa	90.12	77.10	79.00	88.78	80.62	81.59	94.32	90.27	90.95
CoCo	87.58	69.54	76.95	85.97	70.38	74.58	98.50	90.09	90.98
RADAR	58.05	48.54	47.11	63.01	60.21	67.42	49.78	47.52	58.47
Watermark	88.91	66.35	47.01	72.26	55.16	50.07	95.13	69.05	68.16
Binculars	76.39	45.42	51.23	87.12	52.32	54.54	91.95	72.34	78.68
PECOLA	86.29	78.13	60.08	84.51	62.64	60.71	98.35	65.09	68.82
OSTAR	**92.36**	**81.27**	**81.46**	**90.23**	**84.48**	**86.04**	**99.55**	**95.72**	**97.52**

Table 3: Results of the ablation study on DeepFake dataset with adversarial attacks . We selected Accuracy and F1-score for evaluating, with the best-performing results highlighted in bold.

Model	Orginal		Pert.		Para.	
	ACC	F1	ACC	F1	ACC	F1
OSTAR (Plain)	90.34	90.12	81.10	77.10	82.61	79.00
OSTAR (Feature Extract)	90.72	90.58	81.67	77.08	82.97	79.27
OSTAR (Feature Extract+Analysis)	**92.17**	**92.87**	82.51	80.25	83.88	80.17
OSTAR	91.94	92.36	**84.34**	**81.27**	**84.75**	**81.46**

trained frameworks, thereby requiring re-adaptation to its unique text generation patterns. Our method achieves significant improvements on GLM-130B, demonstrating that the statistical feature analysis approach we proposed serves as a universal feature analysis method for MGT.

Evaluation on Attacked Datasets The ability of detection methods to maintain robustness across diverse attack types constitutes a fundamental research problem, as this capability directly determines their practical applicability in real-world environments. The changes in the model's F1-score across the three datasets under these two types of attacks serve as the robustness measure for our method. As shown in Table 2, our proposed OSTAR framework achieves state-of-the-art F1-scores across all 9 experimental scenarios, which encompass three diverse public datasets (CheckGPT, HC3, and DeepFake) under three distinct conditions (original, perturbation-attacked, and paraphrase-attacked environments). Under adversarial perturbations, OSTAR exhibits a maximum F1 degradation of only 11.09% on the challenging DeepFake-Perturbation subset, significantly outperforming the statistical method Binoculars, which suffers a 30.97% degradation—this stark contrast underscores the inherent limitations of relying solely on static statistical thresholds for real-world robustness. Moreover, with an average F1 degradation of 6.30% against combined perturbation and paraphrase attacks across all datasets, OSTAR surpasses even the most robust baseline model, RoBERTa (which shows 8.49% degradation), thereby highlighting our framework's superior adversarial resilience. Furthermore, OSTAR achieves an average F1 improvement of 4.43% over the RoBERTa baseline, with a maximum gain of 5.5% observed on the HC3 dataset under adversarial attacks—these results collectively demonstrate substantial robustness enhancement and validate the effectiveness of our approach in practical settings.

4.3 Ablation Study

To better validate the necessity of each module in our model, we conducted ablation studies on DeepFake dataset. The reason why we chose the Deepfake dataset as it showed the sharpest performance drop under adversarial environments, best demonstrating our method's robustness in adversarial scenarios. The ablation study structure designed as follows:

OSTAR (Plain) removes the entire Statistical Feature component, retaining only the RoBERTa part. During model training, it also eliminates MFCL while keeping solely the CE loss.

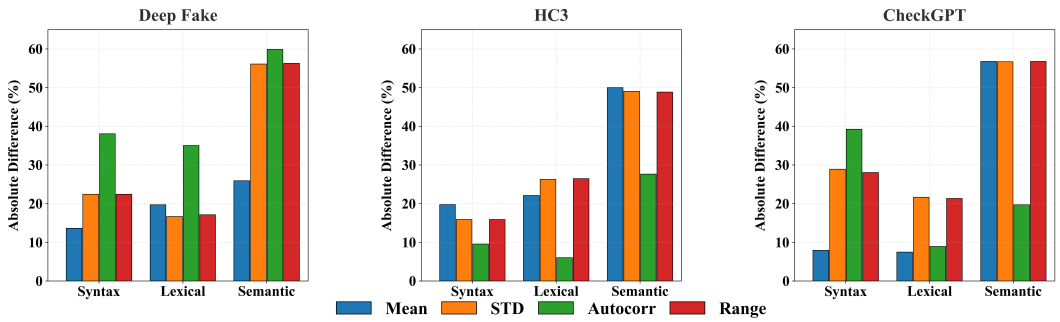

Figure 3: The MDSP performance on three datasets. The vertical axis shows the average absolute difference in MDSP statistical features between human and machine texts across three datasets, demonstrating MDSP's distinct discrimination capability.

OSTAR (Feature Extract) retains the three-dimensional feature extraction component but bypasses analysis processing of these features. Instead, it directly applies global average pooling and projects the results to corresponding dimensions for concatenation with RoBERTa's embeddings. The final classification is performed using a linear classification head with CE loss.

OSTAR (Feature Extract+Analysis) incorporates the complete Statistical Enhancement component but excludes Multi-Faceted Contrastive Learning during training.

Table 3 shows that both Statistical Enhancement and Multi-Faceted Contrast Learning significantly boost the model's detection performance and adversarial robustness. Our OSTAR achieves optimal performance when facing perturbations and paraphrases attacks. Under these attacks, the ACC drops by 7.60% (perturbation) and 7.19% (paraphrases), while F1-score declines by 11.09% and 10.90% respectively — significantly smaller degradation compared to models without MFCL, demonstrating the necessity of our method. The progressive performance improvements from OSTAR (Plain) $\rightarrow$ OSTAR (Feature Extract) $\rightarrow$ OSTAR (Feature Extract+Analysis) validate the effectiveness of MDSP. Notably, using only feature extraction yields marginal gains and even degrades performance under perturbations, likely due to redundant features overlapping with CLS embeddings from pretrained models. While OSTAR with MFCL shows slightly lower performance on the original dataset compared to non-MFCL counterparts (attributed to MFCL training exclusively on attacked data, reducing fidelity to original distributions), this degradation remains within acceptable range (0.23% in acc and 0.51% in F1).

4.4 Statistical Feature Evaluation

As demonstrated in Figure 3, our statistical text analysis method, evaluated on three public datasets(HC3, CheckGPT, DeepFake), reveals an average discrepancy of 30.95% between MGT and human-authored texts, proving its significant discriminative power for MGT detection. Extended analysis of our MDSP framework shows in Appendix A. This suite of stable and quantifiable intrinsic statistical features effectively uncovers systematic biases in the linguistic patterns of machine-generated texts. It serves as a critical anchor point for the OSTAR framework, enhancing detection robustness in adversarial environments by compensating for the tendency of pure neural network features to deviate under attacks.

5 Conclusion

In this paper, we propose OSTAR, a robust MGT detection framework that synergizes the intrinsic invariant feature extraction capability of statistics-based methods with the dynamic adaptability of classifier-based approaches. Specifically, we design the MDSP module to manually extract and analyze statistical features across multiple intrinsic dimensions, enhancing classification through feature fusion. To address adversarial environments, we categorize attacks into Perturbation and Paraphrase based on their impact mechanisms, and accordingly develop MFCL to improve robustness by disentangling adversarial effects through multi-perspective feature alignment. Extensive experiments across three public datasets with 9 kinds of perturbations and a paraphraser, validate the effectiveness of OSTAR and demonstrate its robustness under various attacks.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (62002027, 62472042,62572075,62277001), the Beijing Municipal Natural Science Foundation (L257023, L233034), the Fundamental Research Funds for the Central Universities (No. CUC25SG013) the Fundamental Research Funds for the Beijing University of Posts and Telecommunications(Grant No.2025TSQY01), the National Natural Science Foundation of China Youth Project (Grant No. 62402057) and the State Key Laboratory of Cyberspace Security Defense (No.2025-C08).

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[3] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. *Advances in neural information processing systems*, 32, 2019.

[4] Yuta Yanagi, Ryohei Orihara, Yuichi Sei, Yasuyuki Tahara, and Akihiko Ohsuga. Fake news detection with generated comments for news articles. In *2020 IEEE 24th International Conference on Intelligent Engineering Systems (INES)*, pages 85–90. IEEE, 2020.

[5] Xiaohui Song, Longtao Huang, Hui Xue, and Songlin Hu. Supervised prototypical contrastive learning for emotion recognition in conversation. *arXiv preprint arXiv:2210.08713*, 2022.

[6] Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36:27469–27500, 2023.

[7] Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, pages 1–66, 2025.

[8] Ying Zhou, Ben He, and Le Sun. Navigating the shadows: Unveiling effective disturbances for modern ai content detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10847–10861, 2024.

[9] Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. Llmdet: A third party large language models generated text detection tool. *arXiv preprint arXiv:2305.15004*, 2023.

[10] Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR, 2023.

[11] Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting llms with binoculars: zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning*, pages 17519–17537, 2024.

[12] Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. Eagle: A domain generalization framework for ai-generated text detection. *arXiv preprint arXiv:2403.15690*, 2024.

[13] Suriya Prakash Jambunathan, Ashwath Shankarnarayan, and Parijat Dube. Convnlp: Image-based ai text detection. In *2024 IEEE International Conference on Big Data (BigData)*, pages 6260–6268. IEEE, 2024.

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019.

[15] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, 2020.

[16] Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023.

[17] Abe Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. k-semstamp: A clustering-based semantic watermark for detection of machine-generated text. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1706–1715, 2024.

[18] Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. An entropy-based text watermarking detection method. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11724–11735, 2024.

[19] Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. A robust semantics-based watermark for large language model against paraphrasing. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 613–625, 2024.

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

[21] Eva Giboulot and Teddy Furon. Watermax: breaking the llm watermark detectability-robustness-quality trade-off. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[22] Shengchao Liu, Xiaoming Liu, Yichen Wang, Zehua Cheng, Chengzhengxu Li, Zhaohan Zhang, Yu Lan, and Chao Shen. Does detectgpt fully utilize perturbation? bridging selective perturbation to fine-tuned contrastive learning detector would be better. *arXiv preprint arXiv:2402.00263*, 2024.

[23] Zae Myung Kim, Kwang Lee, Preston Zhu, Vipul Raheja, and Dongyeop Kang. Threads of subtlety: Detecting machine-generated texts through discourse motifs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5449–5474, 2024.

[24] Zhongping Zhang, Wenda Qin, and Bryan Plummer. Machine-generated text localization. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 8357–8371, 2024.

[25] Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. Intrinsic dimension estimation for robust detection of ai-generated texts. *Advances in Neural Information Processing Systems*, 36:39257–39276, 2023.

[26] Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Radar: Robust ai-text detection via adversarial learning. *Advances in neural information processing systems*, 36:15077–15095, 2023.

[27] Guanhua Huang, Yuchen Zhang, Zhe Li, Yongjian You, Mingze Wang, and Zhouwang Yang. Are ai-generated text detectors robust to adversarial perturbations? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6005–6024, 2024.

[28] Bairu Hou, Joe O'connor, Jacob Andreas, Shiyu Chang, and Yang Zhang. Promptboosting: Black-box text classification with ten forward passes. In *International Conference on Machine Learning*, pages 13309–13324. PMLR, 2023.

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

[30] Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. Coco: Coherence-enhanced machine-generated text detection under low resource with contrastive learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16167–16188, 2023.

[31] Junchao Wu, Runzhe Zhan, Derek Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia Chao. Detectrl: Benchmarking llm-generated text detection in real-world scenarios. *Advances in Neural Information Processing Systems*, 37:100369–100401, 2024.

[32] Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. Raid: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, 2024.

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

[34] Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE, 2018.

[35] Junkang Wu, Jiawei Chen, Jiancan Wu, Wentao Shi, Xiang Wang, and Xiangnan He. Understanding contrastive learning via distributionally robust optimization. *Advances in Neural Information Processing Systems*, 36:23297–23320, 2023.

[36] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. Consert: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 2021.

[37] Xiaoqi Qiu, Yongjie Wang, Xu Guo, Zhiwei Zeng, Yu Yue, Yuhong Feng, and Chunyan Miao. Paircfr: Enhancing model training on paired counterfactually augmented data through contrastive learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11955–11971, 2024.

[38] Xun Guo, Yongxin He, Shan Zhang, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. Detective: Detecting ai-generated text via multi-level contrastive learning. *Advances in Neural Information Processing Systems*, 37:88320–88347, 2024.

[39] Yau-Shian Wang, Ta-Chung Chi, Ruohong Zhang, and YIMING YANG. Pesco: Prompt-enhanced self contrastive learning for zero-shot text classification. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

[40] Yichen Wang, Shangbin Feng, Abe Hou, Xiao Pu, Chao Shen, Xiaoming Liu, Yulia Tsvetkov, and Tianxing He. Stumbling blocks: Stress testing the robustness of machine-generated text detectors under attacks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2894–2925, 2024.

[41] Zeyan Liu, Zijun Yao, Fengjun Li, and Bo Luo. On the detectability of chatgpt content: benchmarking, methodology, and evaluation through the lens of academic writing. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2236–2250, 2024.

[42] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. In *The Eleventh International Conference on Learning Representations*.

[43] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

A Statistical Feature Evaluation

To further verify that the features extracted using MDSP maintain a certain level of stability compared to the original text after being subjected to perturbations and paraphrases, we employed kernel density estimation (KDE) plots to evaluate each statistical feature.

A.1 Variations in the Syntax Features Extracted by MDSP

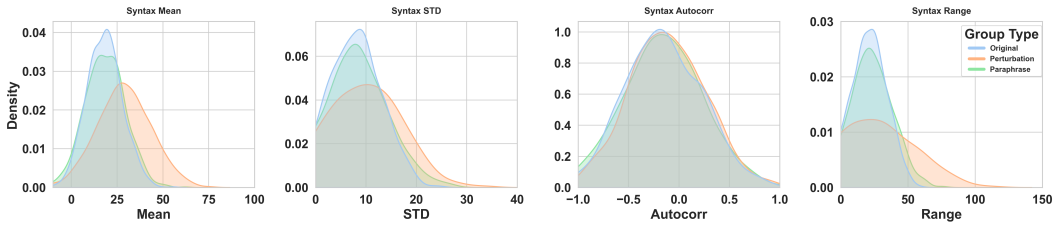

Figure 1: Variations in the Syntax features extracted by MDSP on human texts

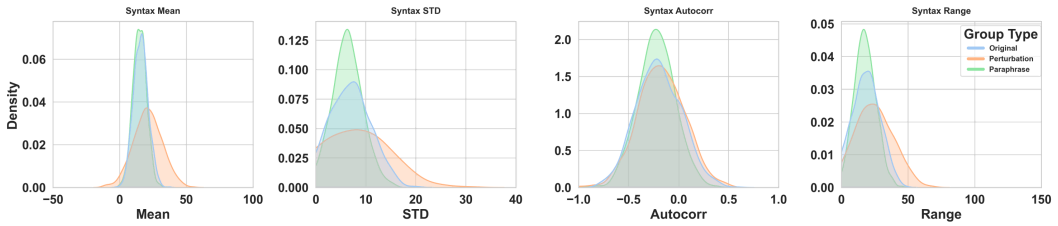

Figure 2: Variations in the Syntax features extracted by MDSP on machine texts

As revealed by the KDE plots, texts subjected to perturbation attacks (e.g., sentence repetition) exhibit substantial divergence from original human texts in the syntactic statistical features extracted by MDSP. This discrepancy likely stems from how repetitive patterns disrupt syntactic regularity, yet they maintain sufficient similarity for discriminative feature learning. In contrast, paraphrased texts demonstrate minimal syntactic deviation from original human-authored content, which may hinder classification accuracy and thus necessitates more sophisticated classifier learning to capture subtle discriminative patterns.

A.2 Variations in the Lexical Features Extracted by MDSP

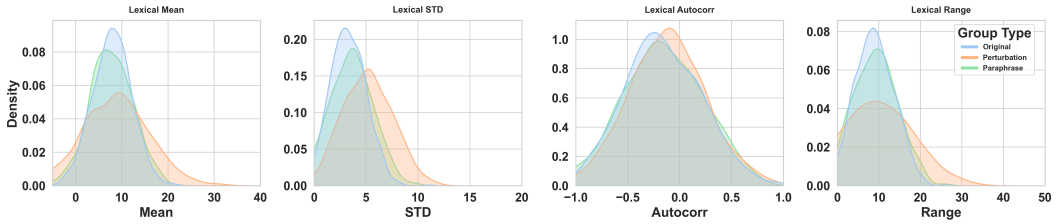

Figure 3: Variations in the Lexical features extracted by MDSP on human texts

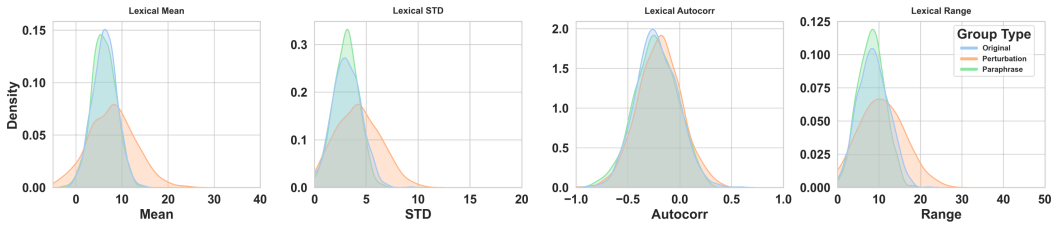

Figure 4: Variations in the Lexical features extracted by MDSP on machine texts

At the lexical level, the three text categories (original, perturbed, and paraphrased) exhibit the closest similarity in auto-corrected (Autocorr) feature representations, which significantly aids in distinguishing perturbation-attacked texts. In contrast, paraphrased texts show minimal divergence across four lexical complexity metrics (e.g., type-token ratio, entropy), yet this subtle variation retains discriminative relevance for identifying machine-generated paraphrased content.

A.3 Variations in the Semantic Features Extracted by MDSP

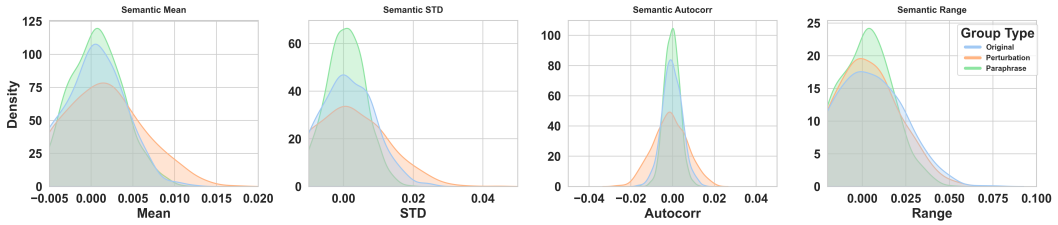

Figure 5: Variations in the Semantic features extracted by MDSP on human texts

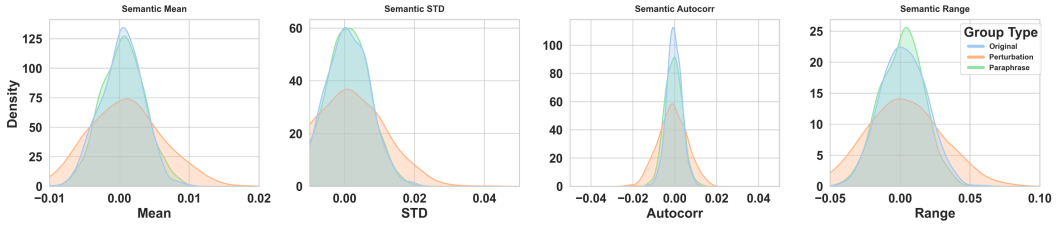

Figure 6: Variations in the Semantic features extracted by MDSP on machine texts

At the semantic level, paraphrased human texts exhibit significant deviations from original human texts across four key metrics (e.g., semantic coherence, entity consistency), which effectively enhances their discriminability. In contrast, paraphrased machine-generated texts maintain close statistical alignment with original machine texts in terms of mean values, standard deviations (STD), and value ranges, thereby enabling robust identification of machine-generated paraphrased content.

B Limitations

Our method achieves generalized detection against various pre-seen attacks in the training set, but struggles to maintain robustness when confronted with unforeseen attacks such as multiple paraphrases, combined attacks, etc. Another limitation is that the sliding window approach for statistical feature analysis imposes requirements on text length (e.g., short single-sentence passages cannot support effective statistical feature extraction).

C Detailed Construction of Dataset

For the three datasets, we constructed both original and adversarially attacked versions. The detailed composition of our original datasets is presented in Table 4, while the configuration of the attacked datasets (generated via perturbation/paraphrases attacks) is summarized in Table 5. The two-tuple (human, machine) in the table represents the number of human texts and the number of AI texts. The symbol "$\times$ " in Table 5 indicates that nine distinct adversarial perturbation methods (shown in appendix E) were applied to each original sample, resulting in a tenfold expansion of the dataset size. To mitigate computational overhead caused by this exponential growth, we selected part of each dataset to include 500 human-authored and 500 machine-generated samples for balanced training and testing.

Table 4: Dataset composition during the Evaluation on Original Datasets experiment.

Dataset	**Train**	**Test**	**Valid**
CheckGPT	(2000, 2000)	(1921, 2078)	(2500, 2500)
HC3	(5000, 5000)	(5000, 5000)	(2000, 2000)
DeepFake	(2000, 2000)	(2000, 2000)	(2000, 1000)

Table 5: Perturbation Methods and Their Intensities

Attack Name	**Intensity**	**Explanation**
Space Insertion	5-10 spaces	Inserts 5-10 spaces randomly in text
Punctuation Removal	Single char	Removes last punctuation character from text
Initial Character Case Alteration	10%	Randomly alter 10% of word initial characters
Word Merging	20%	Randomly merge 20% of adjacent words
Keyboard Typos	10%	Generates typos in 10% characters using adjacent keys
Character WordCase	20%	Randomly changes case for 20% of words
Spelling Errors	3	Introduces 3 spelling errors in each sentence
Adverb Append	1	Appends one adverb to each sentence
Sentence Reversal	10%	Reverses text segments using 3-word pivots in 10% sentence groups
Sentence Repetition	3	Selects 3 sentences to repeat

Table 6 shows the detailed composition of the attacked datasets used in the Evaluation on Attacked Datasets experiment, where the "×" symbol indicates the multiplication factor applied to the original sample counts due to adversarial attacks.

Table 6: Dataset composition during the Evaluation on Attacked Datasets experiment.

Attack Type	**Data Split**		
	Train	Test	Valid
Perturbation	(500×10,500×10)	(500×10,500×10)	(200×10,200×10)
Paraphrase	(500×2,500×2)	(500×2,500×2)	(200×2,200×2)

D Training Details

Our implementation uses RoBERTa as the base pretrained model. Identical attack procedures were applied to both training and test sets. For the MDSP, we set length $l = 3$. The contrastive learning component employs a weight of 0.02 and temperature coefficient of 0.05. Training utilizes the Adam optimizer with learning rate 1×10^{-5} and adam epsilon of 1×10^{-8}. Our model was trained on an NVIDIA RTX 3090 GPU, requiring approximately 17GB of VRAM with a batch size of 4, making it feasible to implement under most laboratory conditions.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contributions and performance of our OSTAR to utilizing statistical features to guide classifier-based detection methods and MFCL optimizing strategy, are reflected in the Introduction and Abstract sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, this paper discuss the limitations in Appendix B.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theoretical results, we validate our method by experimental results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We fully disclose all the information needed to reproduce the main experimetal results of the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will provide open access to the code once the paper is published.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all the training and test details in Section 4.1 and Appendix C,D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: [NA]

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide sufficient information on the computer resources in Appendix F.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics and our research was conducted with that in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The societal impacts of our work are discussed in introduction

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly credited it.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code and models will be well documented.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing experiments and research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [No]

Justification: The LLM is used only for writing

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.