Defensive Dual Masking for Robust Adversarial Defense

Wangli Yang[1], Jie Yang[1*], Yi Guo[2*], Johan Barthelemy[3]

[1]School of Computing and Information Technology, University of Wollongong
wangli@uow.edu.au, jiey@uow.edu.au
[2]School of Computer, Data and Mathematical Sciences, Western Sydney University
y.guo@westernsydney.edu.au
[3]NVIDIA
jbarthelemy@nvidia.com

*Adversarial defenses for textual data have gained considerable attention in recent years due to the increasing vulnerability of Natural Language Processing (NLP) models to adversarial attacks. These attacks exploit subtle perturbations in input text to deceive models, posing significant challenges to model robustness and reliability. This article introduces **D**efensive **D**ual **M**asking (DDM), a simple yet effective algorithm that uses two unique masking strategies to mitigate adversarial threats. Specifically, during training, `[MASK]` tokens are directly inserted into input samples to prepare the model for handling perturbed inputs. At inference time, suspicious tokens are identified and strategically replaced with `[MASK]` tokens, effectively neutralizing perturbations while preserving core semantics of the input text. The theoretical foundation of DDM demonstrates how the proposed masking strategies enhance the model capacity to mitigate adversarial attacks. Empirical evaluations based on four benchmark datasets and four adversarial attacks consistently demonstrate that DDM outperforms state-of-the-art defense techniques, achieving superior robustness and substantial improvements in model accuracy. Furthermore, DDM seamlessly integrates with Large Language Models, enhancing their resilience to adversarial attacks and providing a scalable defense solution for large-scale NLP applications.*

1. Introduction

Language Models (LMs) significantly advance the performance of many Natural Language Processing (NLP) tasks, spanning text/document classification, semantic analysis, and topic clustering (Li et al. 2024a; Czinczoll et al. 2024; Li et al. 2024b). However, extensive research reveals that LMs are susceptible to adversarial attacks, where even subtle perturbations to input texts adversely affect model performance. Specifically,

* Corresponding authors.

Action Editor: Xuanjing Huang. Submission received: 30 January 2025; revised version received: 20 June 2025; accepted for publication: 26 August 2025.

https://doi.org/10.1162/COLI.a.574

fine-tuned LMs demonstrate a significant decrease in performance, up to 85%, due to the presence of even a single-character misspelling within input texts (Gao et al. 2018; Li et al. 2019; Li et al. 2020; Jin et al. 2020), highlighting their limited robustness in generalization. Consequently, considerable focus is placed on developing adversarial detection and defense methods to ensure robust model performance on both original (clean) and polluted (adversarial) inputs.

Adversarial detection aims to differentiate clean from adversarial inputs by analyzing feature representations or model behavior (Carrara et al. 2018; Freitas et al. 2020; Li, Angelov, and Suri 2025). While detection provides a critical safeguard to models, it does not mitigate attack impact. On the other hand, adversarial defenses focus on proactively enhancing model robustness through various strategies, including data augmentation, model adaptation, and randomized smoothing. **Data augmentation** techniques, commonly referred to as *adversarial training*, introduce controlled perturbations to clean data, generating noisy variants that are used alongside the clean data for model fine-tuning (Yoo and Qi 2021; Li et al. 2021; Meng et al. 2022; Li, Song, and Qiu 2023; Rafiei Asl et al. 2024). Although effective, these methods often demand substantial computational resources due to the need for both generating and training on additional augmented samples. **Model adaptation** approaches focus on refining the vanilla model via either modifying the training loss function or adjusting the network architecture (Wang et al. 2021; Liu et al. 2022; Zhan et al. 2023; Moraffah et al. 2024). However, these modifications typically require extensive hyperparameter tuning and are prone to overfitting, which may undermine the model's generalization capabilities. Another line of work explores ensemble-based **randomized smoothing** techniques (Ye, Gong, and Liu 2020; Zeng et al. 2023; Zhang et al. 2024). These methods, however, incur overhead due to the nature of ensemble classification and tend to exhibit inconsistent performance against various attack types (Zhang et al. 2022; Xu et al. 2022). Detailed elaboration on existing adversarial defense methods is provided in Section 2. Thus, further investigations are necessary to improve the generalizability and robustness of models against adversarial attacks.

This article introduces a novel algorithm, termed **D**efensive **D**ual **M**asking (***DDM***). The core of *DDM* lies in strategically integrating `[MASK]` tokens during both training and inference. Specifically, during training, *DDM* directly inserts `[MASK]` tokens into input sequences rather than replacing existing tokens. This simple yet effective strategy generates masked variants of input data, which fine-tune the model without requiring training on the original unmodified data. During inference, *DDM* identifies suspicious tokens in unseen samples and selectively masks them. Importantly, *DDM* does not attempt to predict the content of masked tokens; instead, the masked samples are fed directly into the fine-tuned model for inference. The proposed *DDM* demonstrates simplicity and effectiveness compared to existing adversarial defense methods:

- Unlike traditional data augmentation techniques, *DDM* injects a few `[MASK]` tokens into the original inputs and fine-tunes using these masked variants only. This slightly increases input length but eliminates the computational overhead of generating and training on additional data.
- In contrast to model adaptation approaches, *DDM* preserves the standard model architecture and loss functions, ensuring consistency and compatibility. Its plug-and-play design facilitates seamless integration into other existing frameworks.

Table 1
Summary of existing masking strategies used in adversarial defense, where **RP**, **RO**, and **IM** refer to **R**eplace-then-**P**redict, **R**eplace **O**nly, and **I**sert **M**ask processes, respectively, and **M** denotes applying masking multiple times to a single input.

Method	Training			Inference		
	RP	**RO**	**IM**	**RP**	**RO**	**IM**
RMLM (Wang et al. 2023)	✓			✓		
Adv-Purification (Li, Song, and Qiu 2023)	✓					
MVP (Raman et al. 2023)	✓			✓		
GenerAT (Zhao and Mao 2023)	✓					
RanMASK (Zeng et al. 2023)		✓(M)			✓(M)	
RSMI (Moon et al. 2023)					✓(M)	
MI4D (Hu et al. 2023)			✓			✓
RobustSentEmbed (Rafiei Asl et al. 2024)	✓			✓		
DeCoGLM (Li and Wang 2024)	✓			✓		
MaskPure (Gietz and Kalita 2024)	✓			✓		
Proposed *DDM*			✓		✓	

- Compared with randomized smoothing methods relying on ensemble learning, *DDM* eliminates ensembling requirements to reduce implementation complexity.
- The proposed method also unifies adversarial detection and defense, offering a systematic approach to utilize detection outcomes for effectively mitigating the influence of adversarial tokens.

We further observe that several existing studies incorporate masking strategies for adversarial defense. However, our proposed method differs from these masking-based approaches in several aspects, as summarized in Table 1. First, most existing approaches (Wang et al. 2023; Li, Song, and Qiu 2023; Zhao and Mao 2023; Raman et al. 2023; Rafiei Asl et al. 2024; Li and Wang 2024; Gietz and Kalita 2024) utilize the `[Mask]` token to occlude portions of the input sequence and then predict the missing tokens, either during the training or inference stage. Their primary goal is to generate augmented training data or modify unseen testing data. In contrast, *DDM* directly uses masked variants for fine-tuning and inference, eliminating the need for prediction. This design reduces potential noise and computational cost. Second, some methods (Zeng et al. 2023; Moon et al. 2023) generate multiple masked variants of the same input during inference, leading to significant computational overhead. *DDM*, however, eliminates this requirement by adopting a single-pass strategy to identify and discard suspicious tokens. This design maintains computational efficiency without compromising robustness against adversarial attacks. The above distinctions demonstrate that *DDM* offers a streamlined and computationally efficient mechanism, setting it apart from prior masking-based approaches.

A preliminary version of this work appears in Hu et al. (2023), where `[MASK]` tokens are randomly inserted during both the training and inference stages. This article extends the previous approach by specifically masking suspicious tokens during the inference stage. Unlike prior work from Hu et al. (2023), which lacks targeted adversarial

handling, this extension is non-trivial as it strategically identifies and neutralizes suspicious tokens during inference. Importantly, we provide a comprehensive theoretical analysis to substantiate the robustness and effectiveness of the proposed approach. Specifically, this analysis includes a formal proof of the method's stability under mild conditions and uninformative distributions, ensuring consistent performance under varying conditions, as well as its defense guarantees. Furthermore, the proposed method bridges adversarial detection and defense as a unified framework, providing a systematic guideline for leveraging detection results to effectively mitigate the impact of adversarial tokens. We conduct extensive experiments and ablation studies across four widely adopted benchmarks and four adversarial attack methods to evaluate the effectiveness and robustness of our approach. These experiments include direct comparisons with the baseline results reported in Hu et al. (2023), as well as other recent state-of-the-art baselines, and additional evaluations in the context of Large Language Models (LLMs). Empirically, the proposed *DDM* method consistently outperforms existing methods, via achieving an absolute improvement of 1.4 to 12.1 points in accuracy on average.

The article is structured as follows: Section 2 surveys existing work on adversarial attack, detection, and defense methods. Section 3 introduces the proposed method and offers theoretical analysis into the effects of masked variations on model fine-tuning and inference. Section 4 evaluates the method across a combination of four highly competitive benchmarks and four attacking mechanisms, followed by a comprehensive ablation study and discussion in Section 5. Finally, Section 6 concludes and outlines future research directions.[1]

2. Related Work

As Transformer-based LMs gain widespread use in tasks such as text classification (Li et al. 2024a), topic clustering (Li et al. 2024b), and question answering (Yang et al. 2024), their susceptibility to adversarial attacks emerges as a critical research focus. This section reviews existing literature on textual adversarial attack, detection, and defense mechanisms, including techniques for generating adversarial examples at the character and word levels, strategies for detecting these attacks, and methods for improving model robustness.

2.1 Adversarial Attack

Textual adversarial attacks subtly alter input sequences to mislead vanilla models into incorrect predictions while preserving coherence, semantic meaning, and natural grammatical structure, ensuring alignment with human interpretation (Jin et al. 2020). Adversarial attacks in the text domain are primarily categorized based on perturbation granularity into two types: *character*-level and *word*-level perturbations.

Character-level Attacks. These attacks primarily manipulate individual characters within words from the original sample. Yet, human prediction could still remain relatively unaffected to a certain extent due to visual similarity. HotFlip (Ebrahimi et al. 2018) is a character-based attack method that leverages gradients from one-hot input representations to identify changes maximizing the model training loss. It uses beam

1 The source code is publicly available on GitHub at `https://github.com/wlyang538/DDM`.

search to discover character manipulations that effectively confuse the model. DeepWordBug (Gao et al. 2018) utilizes four scoring functions to identify crucial words. Subsequently, different token transformers are utilized to alter these significant words, which involves swapping two adjacent letters, substituting a letter with a random one, deleting a random letter, and inserting a random letter. Similarly, Textbugger (Li et al. 2019) initially identifies important words by either computing the Jacobian matrix of the model output or comparing the model changes before and after word deletion. For identified words, Textbugger extends character-level attacks beyond inserting, deleting, and swapping letters by suggesting the replacement of characters with visually similar or adjacent ones from the keyboard.

Word-level Attacks. These attacks deceive models through subtle word manipulations, such as synonym substitution, while maintaining grammatical correctness and semantic similarity. The Probability Weighted Word Saliency approach (PWWS) uses probability-weighted word saliency to evaluate the sensitivity of the victim model to each input word. Subsequently, candidate words are replaced by their synonyms (from WordNet), taking into account the magnitude of change in the model's output probability. Wang et al. (2020) propose the Fast Gradient Projection Method (FGPM) to construct a synonym set for each input word using its nearest neighbors in the GloVe vector space. Target words are then selected by evaluating the projected distance and gradient change between the original word and its synonym candidates in the gradient direction. Finally, this method achieves textual attack through word replacement using its synonym set. In TextFooler (Jin et al. 2020), target words are identified by comparing changes in prediction results before and after a word deletion. Subsequently, TextFooler replaces these important words with synonyms that are both semantically similar (minimizing their cosine distance) and grammatically correct (verified through part-of-speech checking). To improve, Morris et al. (2020a) further introduce the TFAdjusted attack, which strengthens the original TextFooler by applying more semantic and syntactic constraints during the word replacement. BERT-Attack (Li et al. 2020) adopts the Masked Language Modeling (MLM) approach by applying the `[MASK]` token to replace existing words in the input sentence. Subsequently, the change in output from the victim model serves as an importance score to select target words, which are then replaced by filling the corresponding `[MASK]` token(s) as part of the MLM process. Recent adversarial attack methods aim not only to deceive the target models but also to retain the semantic meaning and linguistic fluency of the original inputs. For instance, the Semantic Spaces Attack (SemAttack) (Wang et al. 2022) generates adversarial samples by optimizing perturbations within multiple semantic spaces, including typo space, lexical-knowledge space, and contextualized embedding space. Additionally, Liu et al. (2023) propose the Simple and Sweet Paradigm Textual Adversarial Attack (SSPAttack), which initializes adversarial examples through synonym substitution. It then refines these examples by reverting unnecessary changes and adjusting substitutions to be semantically closer to the original texts, thereby enhancing fluency and preserving meaning without requiring an exhaustive global search.

2.2 Adversarial Detection

Adversarial detection aims to differentiate between clean inputs and adversarially crafted ones. Specifically, the Frequency-Guided Word Substitution (FGWS) method (Mozes et al. 2021) exploits word frequency patterns as the detection results, under the hypothesis that adversarial attacks tend to substitute high-frequency words with

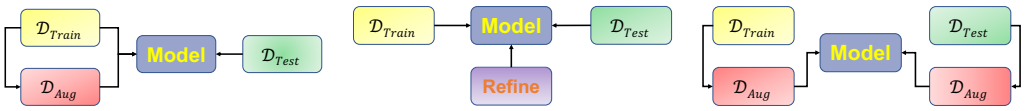
low-frequency counterparts. Extending this idea, Mosca et al. (2022) introduce the Word-level Differential Reaction (WDR) algorithm, which selectively deletes words from input samples, measuring prediction difference to identify adversarial tokens. Recent advances include the Local Outlier Factor (LOF) algorithm (Omar and Sukthankar 2023), which detects adversarial examples by identifying inputs that occupy low-density regions within the data distribution. Similarly, the Class-Aware Score Network (CASN) approach (Bao et al. 2023) uses log-density gradient estimation combined with Langevin dynamics to analyze data distributional discrepancies. The Masked Language Model Detection (MLMD) (Zhang et al. 2023) method leverages representation manifold deviations to identify adversarial attacks. The Universal Adversarial Perturbations for Adversarial Detection (UAPAD) algorithm (Gao et al. 2023) first obtains the universal adversarial perturbations (UAPs). Then, by applying UAPs to testing samples, UAPAD identifies adversarial examples based on prediction differences with and without UAPs. Similarly, the Prediction & Attribution Sensitivity Analysis method (PASA) (Bhusal et al. 2024) introduces controllable perturbations to inputs. PASA then detects adversarial manipulations via examining the sensitivity of model outputs.

Nevertheless, detection methods, while effective, are inherently reactive and do not mitigate adversarial effects or adjust model outputs, usually leaving models vulnerable during attacks. As such, researchers also explore proactive defense algorithms.

2.3 Adversarial Defense

Text adversarial defenses, in contrast to attacks, aim to form a resilient model maintaining high accuracy on both clean (original) and polluted (adversarial) samples. To mitigate the adverse impact of adversarial attacks, defense methods are typically categorized into three strategies (Hu et al. 2023): data augmentation, model adaptation, and randomized smoothing, as shown in Figure 1.

Data Augmentation. This approach involves strategically augmenting original samples to generate several noisy variants, which are simultaneously utilized to fine-tune the victim model. Importantly, the noise introduced during augmentation typically differs from that used in attacks (in a black-box manner). Specifically, Attacking to Training (A2T) (Yoo and Qi 2021) generates noisy variants by utilizing a gradient-based method to identify crucial words, iteratively substituting them with synonyms. The Free Large Batch method (FreeLB) (Zhu et al. 2020), along with its variant FreeLB++ (Li et al. 2021), imposes norm-bounded noise on input embeddings to generate diverse representations. The Anomaly Detection with Frequency Aware Randomization method (Bao, Wang, and Zhao 2021) uses frequency-aware randomization on original and adversarial examples (generated by other attack methods) to create a randomized adversarial set, which



(a) Data augmentation (b) Model adaptation (c) Randomized smoothing

Figure 1
Summary of existing adversarial defense methods categorized by data augmentation, model adaptation, and randomized smoothing.

is then combined with the original samples for model training. Wang et al. (2023) introduce Randomization Masked Language Modeling (RMLM), a synonym-based transformation that randomly corrupts input samples (which could be adversarial) before using an MLM-based defender to reconstruct denoised inputs. A similar approach, termed Text Adversarial Purification (Adv-Purification), is presented in (Li, Song, and Qiu 2023). This method iteratively introduces noise by masking input texts and reconstructing them as part of a multi-run purification process. Additionally, Generative Adversarial Training (GenerAT) (Zhao and Mao 2023) integrates a generative adversarial attack with adversarial training, where the generative model uses classifier gradients to generate perturbed tokens. Model-tuning Via Prompts (MVP) (Raman et al. 2023) uses a prompt template with `[MASK]` tokens to perform classification by filling these tokens. More recently, the Robust Sentence Embeddings method (RobustSentEmbed) (Rafiei Asl et al. 2024) leverages an adversarial perturbation generator to produce high-risk token- and sentence-level perturbations. DeCoGLM (Li and Wang 2024), built on the General Language Model, uses a fault-tolerant detection template for error identification and autoregressive mask infilling for localized correction. Fast Adversarial Training (FAT) (Yang, Liu, and He 2024) addresses synonym-unaware scenarios by using single-step gradient ascent and historical perturbations to generate augmented samples. These augmented samples are typically combined with the original training data for fine-tuning, enhancing the model robustness by exposing it to potential perturbations in advance.

Model Adaptation. This strategy, without generating noisy variants, enhances the victim model architecture and/or training losses. For example, InfoBERT (Wang et al. 2021) refines the model by introducing an Information Bottleneck regularizer to suppress noisy information between inputs and latent representations. Similarly, the Information Bottleneck algorithm (IB) (Zhang et al. 2022) inserts an additional Information Bottleneck layer between the output layer and the encoder to robustify the extracted representation. Le, Park, and Lee (2022) propose the Stochastic Multi-Expert Neural Patcher framework (SHIELD) to modify the last layer of the victim model, formulating it as an ensemble of multi-expert predictors. Flooding-x (Liu et al. 2022) adopts the gradient consistency criterion as a threshold to monitor the training loss and introduces an early-stop technique to prevent overfitting. Zheng et al. (2022) propose the Robust Tickets method (RobustT), which identifies smaller matching subnetworks (robust tickets) within the victim model using binary masks and L_0 regularization. An adversarial loss is also used to ensure that these tickets perform well in terms of both accuracy and robustness. Adversarial Text Interceptor and Rewriter (ATINTER) (Gupta et al. 2023) integrates an additional encoder-decoder module to rewrite adversarial inputs, eliminating adversarial perturbations before model inference. The Similarizing the Influence of Words with Contrastive Learning method (SIWCon) (Zhan et al. 2023) introduces a contrastive learning-based loss to ensure less important input words/tokens have comparable influence on model performance as their more important counterparts. The label smoothing technique (LSDefense) is systematically investigated in Yang et al. (2023) to enhance the model robustness by modifying the loss function through altered target labels. Specifically, the standard Label Smoothing method is utilized to soften the target distribution by assigning a small uniform probability to non-target classes, while Adversarial Label Smoothing allocates the entire smoothing weight to the class with the lowest predicted probability, thereby introducing targeted uncertainty and promoting robustness. More recently, Moraffah et al. (2024) propose the LLM-guided Purification Method (LLMPM), which uses LLMs as purifiers to remove potentially adversarial

perturbations from input texts. The RObust text Inference and Classification Diffusion Model (ROIC-DM) is proposed in Yuan, Yuan, and He (2024), applying diffusion and reverse processes to generate noise for label recovery. Additionally, ROIC-DM integrates pre-trained language models as advisors to guide the denoising process. Another diffusion-based approach, DiffuseDef (Li, Rei, and Specia 2024), uses a diffusion layer to predict randomly sampled noise at a given time step, functioning iteratively as a denoiser to remove adversarial noise from hidden states.

Randomized Smoothing. This approach typically utilizes ensemble-based methods to enhance model robustness against adversarial attacks. A structure-free approach for certified robustness, SAFER, is proposed in Ye, Gong, and Liu (2020) to construct stochastic input ensembles and leverage their statistical properties for classification. In RanMASK (Zeng et al. 2023), input tokens are randomly substituted with `[MASK]` during fine-tuning, while testing samples are also masked to generate multiple masked versions. The final prediction is determined by a majority vote from the ensemble of these masked versions. Randomized Smoothing with Masked Inference (RSMI) (Moon et al. 2023) is a two-stage framework to train a smooth classifier. Tokens with significant loss gradients are selected for masking, followed by multiple Monte-Carlo sampling to craft multiple masked samples. RSMI then produces the outcome by averaging predictions from all these masked samples. The `[MASK]`-Insertion for Defense method (MI4D) (Hu et al. 2023) randomly inserts `[MASK]` tokens into input sequences during training and inference, maximizing the intersection between the new and original convex hulls. More recently, Gietz and Kalita (2024) introduce MaskPure, a method that enhances robustness by randomly masking and refilling portions of the input text prior to classification, offering provable certified robustness guarantees. **Text-RS** (Zhang et al. 2024) treats word substitutions as continuous perturbations on word embeddings and integrates a random smoothing-based certified defense, achieving smooth text representations for improved model robustness.

2.4 Section Summary

Despite demonstrating promising performance, existing methods often face challenges balancing robustness and efficiency. Accordingly, we propose a novel approach leveraging a token-masking mechanism. Our method differs significantly from existing defense approaches by avoiding reliance on complex augmentation strategies for generating additional data (e.g., data augmentation techniques), modifications to the training loss function or model architecture (e.g., model adaptation approaches), and ensemble-based training procedures (e.g., randomized smoothing). Furthermore, it distinguishes itself from existing masking-based techniques by eliminating the need for a separate prediction step for masked tokens, significantly improving computational efficiency and being less susceptible to noise introduced by iterative predictions. Notably, our method seamlessly integrates reactive detection with proactive defense mechanisms, providing a unified framework to identify adversarial threats and mitigate their impact effectively.

3. Proposed Method

This section introduces a simple yet effective algorithm designed to enhance the model resilience against adversarial attacks, termed **D**efensive **D**ual **M**asking (*DDM*). The proposed method is characterized by strategically injecting `[MASK]` tokens into input

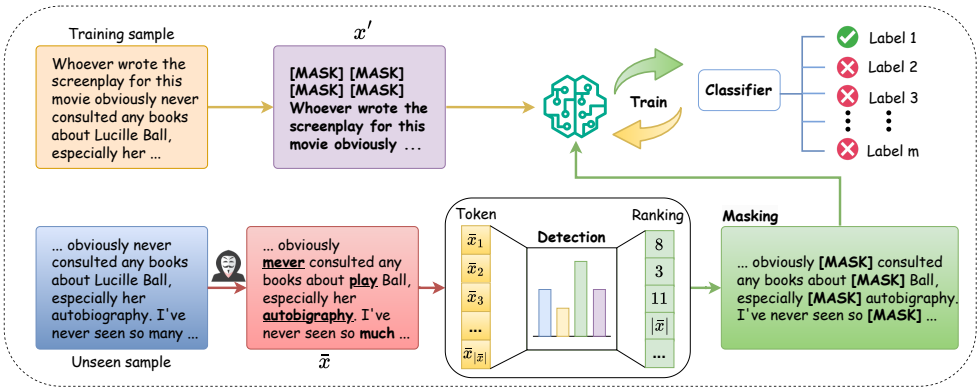


Figure 2
The workflow of our proposed *DDM*, which preserves the model architecture and loss function as the vanilla model. Its distinctiveness lies in integrating `[MASK]` tokens into input sequences during both training and inference stages.

sequences during both training and inference stages. The workflow of the proposed *DDM* is shown in Figure 2.

3.1 Defensive Dual Masking

The proposed method involves two primary stages. In the *training* stage, *DDM* follows the standard fine-tuning process, utilizing the identical network architecture and training loss as the vanilla model. However, it introduces a unique step of directly inserting `[MASK]` tokens into input sequences. In the *inference* stage, our method substitutes suspicious tokens with `[MASK]` before forwarding the sequence for prediction.

Specifically, consider the tokenized (clean) input sequence $\boldsymbol{x}$ (i.e., $\boldsymbol{x} =$ `[CLS]` $\boldsymbol{x}_1 \cdots \boldsymbol{x}_{|\boldsymbol{x}|}$ `[SEP]`), where $\boldsymbol{x}_i$ represents the i-th token from $\boldsymbol{x}$. In the context of text classification, the goal is to optimize an encoder model $\text{Enc}(\cdot)$ and a Multilayer Perceptron layer $\mathcal{F}(\cdot)$ to map $\boldsymbol{x}$ to a desired label y, i.e., $\mathcal{F}(\text{Enc}(\boldsymbol{x})) = y$. Note that, in this article, the `[CLS]` token serves as the aggregate representation of the input sequence and is utilized as the final input representation for downstream tasks.

Accordingly, during *training*, *DDM* injects M consecutive masks after `[CLS]` within $\boldsymbol{x}$ to create a masked sequence, denoted as

$$\boldsymbol{x}' = \texttt{[CLS]}\ \texttt{[MASK]}_1 \cdots \texttt{[MASK]}_M\ \boldsymbol{x}_1 \cdots \boldsymbol{x}_{|\boldsymbol{x}|}\ \texttt{[SEP]}$$

where M is determined as $\lceil b_M \times |\boldsymbol{x}| \rceil$, b_M is the predefined masking budget (or the fraction of masked tokens) and $|\boldsymbol{x}|$ is the cardinality of $\boldsymbol{x}$, i.e., the number of tokens in x. Subsequently, only $\boldsymbol{x}'$ (instead of $\boldsymbol{x}$) is utilized for training, and a standard (Cross-Entropy) loss function, denoted as $\mathcal{L}(\mathcal{F}(\text{Enc}(\boldsymbol{x}')), y)$, is implemented.

During *inference*, when presented with an unseen sequence $\bar{\boldsymbol{x}}$, our method initially computes a suspicious score for each input token. Subsequently, M tokens with the highest suspicious scores from $\bar{\boldsymbol{x}}$ are successively replaced by `[MASK]`, resulting in a

modified sequence $\bar{\boldsymbol{x}}'$. For example, with $M = 2$, let $\bar{\boldsymbol{x}}_i$ and $\bar{\boldsymbol{x}}_j$ be the two tokens with the highest suspicious scores. The modified sequence $\bar{\boldsymbol{x}}'$ then can be expressed as:

$$\bar{\boldsymbol{x}}' = \texttt{[CLS]}\ \bar{\boldsymbol{x}}_1\ \cdots\ \bar{\boldsymbol{x}}_{i-1}\ \texttt{[MASK]}\ \bar{\boldsymbol{x}}_{i+1}\ \cdots\ \bar{\boldsymbol{x}}_{j-1}\ \texttt{[MASK]}\ \bar{\boldsymbol{x}}_{j+1}\ \cdots\ \bar{\boldsymbol{x}}_{|\bar{\boldsymbol{x}}|}\ \texttt{[SEP]}$$

The label of $\bar{\boldsymbol{x}}$ accordingly is determined by $\mathcal{F}(\text{Enc}(\bar{\boldsymbol{x}}'))$. Notably, when either inserting or substituting [MASK] tokens in *DDM*, we set the position embeddings of [MASK] as zero to minimize the positional impact, while preserving their relevant token and token type embeddings.

3.2 Analysis on *DDM*

Our approach leverages [MASK] tokens during both the training and inference phases, each with distinct objectives. During training, [MASK] tokens are inserted at the beginning of samples to introduce perturbations that deviate from the original dataset, effectively acting as a form of noise. This use of [MASK] as a *placeholder* helps the model learn to generalize by exposing it to incomplete or partially obscured data, thus preparing against unseen or adversarial inputs. In the inference phase, [MASK] tokens are strategically utilized to replace suspicious tokens, allowing the model to mitigate the influence of adversarial perturbations while preserving the semantic integrity of the underlying context.

This method builds upon the approach introduced in Hu et al. (2023) by addressing a critical limitation in the previous method that lacks targeted mechanisms to mitigate adversarial attacks. That is, unlike the indiscriminate insertion of [MASK] tokens and the retention of all tokens as proposed in Hu et al. (2023), our method adaptively neutralizes adversarial inputs by identifying suspicious tokens and selectively replacing them with [MASK]. This adaptive masking strategy, while conceptually straightforward, represents a significant advancement in adversarial defense, enabling the retention of essential contextual information while effectively mitigating the impact of perturbations. The proposed detect-then-mask strategy also effectively bridges the gap between identifying adversarial threats and defending against their impact. A comprehensive theoretical analysis below further demonstrates these advantages.

Let $\mathbf{a}$, $\mathbf{r}$, $\mathbf{S}$, and $\mathbf{m}$ represent the victim token (being attacked), the replaced token (after the attack), the remaining unchanged tokens, and the [MASK] token, respectively,[2] and the hidden dimension is d. Unchanged tokens, i.e., tokens that are not subjected to adversarial attacks, can be "folded" into a single contracted point. The rationale behind is rooted in the attention mechanism of a Transformer model. Consider three tokens, $\mathbf{x}$, $\mathbf{s}$, and $\mathbf{p}$, with their corresponding projections resulting from the linear transformations applied within the attention mechanism:

$$\mathbf{x}_i = \mathbf{x}\mathbf{W}_i, \quad i = 1, 2, 3$$

2 Notation: lowercase letters denote single values; bold lowercase/uppercase letters, e.g., $\mathbf{a}/\mathbf{A}$, represent vectors/matrices, respectively.

Similarly for the other two tokens, i.e., $\mathbf{s}_i = \mathbf{s}\mathbf{W}_i$ and $\mathbf{p}_i = \mathbf{p}\mathbf{W}_i$. The reconstructed token $\tilde{\mathbf{x}}$ after the attention mechanism then is expressed as:

$$\tilde{\mathbf{x}} = \frac{\exp(\mathbf{x}_1\mathbf{x}_2^\top)\mathbf{x}_3 + \exp(\mathbf{x}_1\mathbf{s}_2^\top)\mathbf{s}_3 + \exp(\mathbf{x}_1\mathbf{p}_2^\top)\mathbf{p}_3}{\exp(\mathbf{x}_1\mathbf{x}_2^\top) + \exp(\mathbf{x}_1\mathbf{s}_2^\top) + \exp(\mathbf{x}_1\mathbf{p}_2^\top)} = \frac{\mathbf{x}_3^{\mathbf{s}} + w'\mathbf{p}_3}{1 + w'}$$

where

$$\mathbf{x}_3^{\mathbf{s}} = \frac{\exp(\mathbf{x}_1\mathbf{x}_2^\top)\mathbf{x}_3 + \exp(\mathbf{x}_1\mathbf{s}_2^\top)\mathbf{s}_3}{\exp(\mathbf{x}_1\mathbf{x}_2^\top) + \exp(\mathbf{x}_1\mathbf{s}_2^\top)}, \quad w' = \frac{\exp(\mathbf{x}_1\mathbf{p}_2^\top)}{\exp(\mathbf{x}_1\mathbf{x}_2^\top) + \exp(\mathbf{x}_1\mathbf{s}_2^\top)}$$

That is, $\mathbf{x}_3^{\mathbf{s}}$ is independent of $\mathbf{p}$, and can be regarded as a contraction of $\mathbf{x}$ and $\mathbf{s}$. Due to the flexibility of $\mathbf{W}_2$, we have $w' \in \mathbb{R}^+$, i.e., w' can take any non-negative real value. Applying these observations to the tokens in $\mathbf{S}$ leads to a contracted single point $\mathbf{s}$, whose position is determined by the model parameters $\mathbf{W}_i$.

We continue by analyzing the reconstruction of the `[CLS]` token as the final input representation. The original reconstructed `[CLS]` token, denoted as $\tilde{\mathbf{x}}_a$, can be computed using the vectors $\mathbf{s}$ and $\mathbf{a}$. However, $\tilde{\mathbf{x}}_a$ is perturbed and replaced by $\tilde{\mathbf{x}}_r$, when $\mathbf{a}$ is substituted with $\mathbf{r}$. In the proposed *DDM*, we further introduce the `[MASK]` token, represented as $\mathbf{m}$. Ideally, the reconstructed `[CLS]` token should now be derived solely from $\mathbf{s}$ and $\mathbf{m}$, denoted as $\tilde{\mathbf{x}}_m$. Nevertheless, if the perturbation token $\mathbf{r}$ is retained, the resulting reconstructed `[CLS]` lies within the convex hull of $\mathbf{s}$, $\mathbf{m}$, and $\mathbf{r}$, which follows the MI4D process (Hu et al. 2023). The geometric relationships between these tokens are illustrated in Figure 3. Here, $\mathbf{s}$ represents the compressed token from the intact token set $\mathbf{S}$ (the remaining unchanged tokens). All tokens are assumed to lie on a manifold embedded in $\mathbb{R}^d$, represented as a smooth surface. Due to the attention mechanism, the reconstructed `[CLS]` token must reside within the convex hull formed by the relevant tokens. For example, $\tilde{\mathbf{x}}_a$ lies within $\text{conv}\{\mathbf{s}, \mathbf{a}\}$, where $\text{conv}\{\mathbf{s}, \mathbf{a}\}$ denotes the convex hull between $\mathbf{s}$ and $\mathbf{a}$, geometrically a straight line connecting $\mathbf{s}$ and $\mathbf{a}$. Similarly, this applies to other reconstructions, such as $\tilde{\mathbf{x}}_m$ and $\tilde{\mathbf{x}}_r$. Additionally, we have $\tilde{\mathbf{x}}_{mr} \in \text{conv}\{\mathbf{s}, \mathbf{m}, \mathbf{r}\}$, where $\text{conv}\{\mathbf{s}, \mathbf{m}, \mathbf{r}\}$ forms a triangle enclosed by $\mathbf{s}$, $\mathbf{m}$, and $\mathbf{r}$. Given the model's high complexity, precisely locating the reconstructed `[CLS]` token is challenging. Therefore,

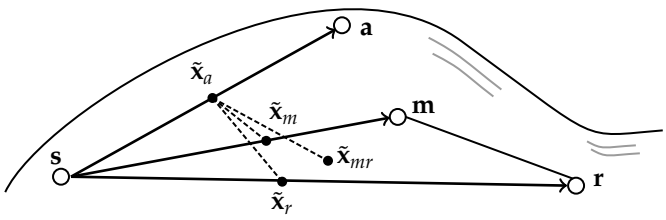


Figure 3
The token geometry where $\mathbf{a}$, $\mathbf{r}$, $\mathbf{s}$, and $\mathbf{m}$ represent the victim token, replaced token, compressed unchanged tokens, and `[MASK]` token, respectively. $\tilde{\mathbf{x}}_a$, $\tilde{\mathbf{x}}_m$, $\tilde{\mathbf{x}}_r$, and $\tilde{\mathbf{x}}_{mr}$ denote the reconstructed `[CLS]` token using combinations of $\mathbf{a}$, $\mathbf{m}$, $\mathbf{r}$, and $\mathbf{s}$.

we assume uniformity in the distribution of the reconstructed token within the convex hull of the relevant tokens, which we formalize in the following assumption:

Assumption 1
The reconstructed `[CLS]` token is uniformly distributed within the convex hull formed by the relevant tokens.

For example, $\tilde{\mathbf{x}}_m$ is then uniformly distributed within $\text{conv}\{\mathbf{s}, \mathbf{m}\}$. This represents an uninformative distribution assumption with no preference, i.e., all modes of the random variable are treated equally within its space. Now we can establish the expected distance between a pair of reconstructed `[CLS]` by the following Lemmas.

Lemma 1
Let two vectors $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^d$. $\tilde{\mathbf{a}}$ uniformly distributed between the origin $\mathbf{o}$ and $\mathbf{a}$, and similarly $\tilde{\mathbf{b}}$ uniformly distributed between $\mathbf{o}$ and $\mathbf{b}$ independent of $\tilde{\mathbf{a}}$. Then

$$\mathbb{E}(\|\tilde{\mathbf{a}} - \tilde{\mathbf{b}}\|^2) = \frac{1}{3}\|\mathbf{a}\|^2 + \frac{1}{3}\|\mathbf{b}\|^2 - \frac{1}{2}\mathbf{a}^\top\mathbf{b}$$

where $\|\mathbf{x}\|$ is the ℓ_2 norm of vector $\mathbf{x}$ and $\mathbb{E}$ is the expectation.

Proof. We rewrite the expectation as

$$\mathbb{E}(\|\tilde{\mathbf{a}} - \tilde{\mathbf{b}}\|^2) = \mathbb{E}(\|\alpha\mathbf{a} - \beta\mathbf{b}\|^2)$$

where α and β are two independent random variables uniformly distributed in $[0, 1]$, i.e., $\alpha, \beta \sim \mathcal{U}(0, 1)$. Then we have

$$\begin{aligned}
\mathbb{E}(\|\alpha\mathbf{a} - \beta\mathbf{b}\|^2) &= \mathbb{E}(\alpha^2\|\mathbf{a}\|^2 + \beta^2\|\mathbf{b}\|^2 - 2\alpha\beta\mathbf{a}^\top\mathbf{b}) \\
&= \mathbb{E}(\alpha^2)\|\mathbf{a}\|^2 + \mathbb{E}(\beta^2)\|\mathbf{b}\|^2 - 2\mathbb{E}(\alpha\beta)\mathbf{a}^\top\mathbf{b} \\
&= \frac{1}{3}\|\mathbf{a}\|^2 + \frac{1}{3}\|\mathbf{b}\|^2 - \frac{1}{2}\mathbf{a}^\top\mathbf{b}
\end{aligned}$$

In the last equality, we used the second momentum of uniform distribution, i.e., $\mathbb{E}(\beta^2) = \frac{1}{3}$, and expectation of a product of two independent random variables, i.e., $\mathbb{E}(\alpha\beta) = \mathbb{E}(\alpha)\mathbb{E}(\beta)$. □

We further extend the above to the case with three vectors.

Lemma 2
Let three vectors $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c} \in \mathbb{R}^d$. $\tilde{\mathbf{a}}$ uniformly distributed between the origin $\mathbf{o}$ and $\mathbf{a}$, and $\tilde{\mathbf{m}}$ uniformly distributed within $\text{conv}\{\mathbf{o}, \mathbf{b}, \mathbf{c}\}$ independent of $\tilde{\mathbf{a}}$. Then

$$\mathbb{E}(\|\tilde{\mathbf{a}} - \tilde{\mathbf{m}}\|^2) = \frac{1}{3}\|\mathbf{a}\|^2 + \frac{1}{9}\|\mathbf{b}\|^2 + \frac{1}{9}\|\mathbf{c}\|^2 - \frac{1}{4}\mathbf{a}^\top\mathbf{b} - \frac{1}{4}\mathbf{a}^\top\mathbf{c} + \frac{1}{9}\mathbf{b}^\top\mathbf{c}$$

Proof. Let $x, y, z \sim \mathcal{U}(0, 1)$ independently. We first construct $\tilde{\mathbf{m}}$ from $\mathbf{b}$ and $\mathbf{c}$ as

$$\tilde{\mathbf{m}} = z(y\mathbf{b} + (1 - y)\mathbf{c})$$

It is easy to see that $\tilde{\mathbf{m}} \subseteq \text{conv}\{\mathbf{o}, \mathbf{b}, \mathbf{c}\}$ as the coefficient for $\mathbf{o}$ is $1 - z(y + (1 - y)) = 1 - z \in [0, 1]$. By following a similar approach as outlined in the proof of Lemma 1, we can rewrite the expectation as follows:

$$\mathbb{E}(\|\tilde{\mathbf{a}} - \tilde{\mathbf{m}}\|^2) = \mathbb{E}(\|x\mathbf{a} - yz\mathbf{b} - z(1-y)\mathbf{c}\|^2)$$

and we further derive

$$\begin{aligned}\mathbb{E}(\|x\mathbf{a} - yz\mathbf{b} - z(1-y)\mathbf{c}\|^2) &= \mathbb{E}(x^2\|\mathbf{a}\|^2 + y^2z^2\|\mathbf{b}\|^2 + (1-y)^2z^2\|\mathbf{c}\|^2 \\ &\quad - 2xyz\mathbf{a}^\top\mathbf{b} - 2xz(1-y)\mathbf{a}^\top\mathbf{c} + 2y(1-y)z^2\mathbf{b}^\top\mathbf{c}) \\ &= \mathbb{E}(x^2)\|\mathbf{a}\|^2 + \mathbb{E}(y^2z^2)\|\mathbf{b}\|^2 + \mathbb{E}((1-y)^2z^2)\|\mathbf{c}\|^2 \\ &\quad - 2\mathbb{E}(xyz)\mathbf{a}^\top\mathbf{b} - 2\mathbb{E}(xz(1-y))\mathbf{a}^\top\mathbf{c} + 2\mathbb{E}(y(1-y)z^2)\mathbf{b}^\top\mathbf{c}\end{aligned}$$

Note that $(1 - y) \sim \mathcal{U}(0, 1)$ and $\mathbb{E}(y(1 - y)) = \mathbb{E}(y - y^2) = \frac{1}{6}$. By using independence of x, y, and z, we have $\mathbb{E}(xyz) = \frac{1}{8}$, $\mathbb{E}(z^2y^2) = \mathbb{E}(z^2)\mathbb{E}(y^2) = \frac{1}{9}$, and $\mathbb{E}(y(1 - y)z^2) = \frac{1}{18}$. Similar results for switching variables. Substituting these values into the above gives the claimed result. □

Theorem 1 (Success condition for *DDM*)
Let $\mathbf{v}_a$, $\mathbf{v}_m$, and $\mathbf{v}_r$ be the vectors of $\mathbf{a}$, $\mathbf{m}$, and $\mathbf{r}$ rooted at $\mathbf{s}$, respectively, l_{am}, l_{ar}, and l_{mr} be cosine similarities between $\mathbf{v}_a$ and $\mathbf{v}_m$, $\mathbf{v}_a$ and $\mathbf{v}_r$, and $\mathbf{v}_m$ and $\mathbf{v}_r$, respectively, for example, $l_{am} = \frac{\mathbf{v}_a^\top \mathbf{v}_m}{\|\mathbf{v}_a\|\|\mathbf{v}_m\|}$. Assuming $\tilde{\mathbf{x}}_a$, $\tilde{\mathbf{x}}_m$, $\tilde{\mathbf{x}}_r$, and $\tilde{\mathbf{x}}_{mr}$ are uniformly distributed in their corresponding convex hulls (Assumption 1) and $l_{mr} > 0$, then we have

$$\mathbb{E}(\|\tilde{\mathbf{x}}_a - \tilde{\mathbf{x}}_m\|^2) \leq \mathbb{E}(\|\tilde{\mathbf{x}}_a - \tilde{\mathbf{x}}_{mr}\|^2) \leq \mathbb{E}(\|\tilde{\mathbf{x}}_a - \tilde{\mathbf{x}}_r\|^2) \tag{3.1}$$

when the following condition is satisfied

$$\|\mathbf{v}_r\| \geq \max\left\{\mathbf{1}(\Delta_1 \geq 0)\frac{9}{8}(\|\mathbf{v}_a(l_{ar} + 4\sqrt{\Delta_1}), \mathbf{1}(\Delta_2 \geq 0)\frac{3}{4}(\|\mathbf{v}_a(l_{ar} + 2\sqrt{\Delta_2}), \frac{\|\mathbf{v}_m\|}{l_{mr}}\right\} \tag{3.2}$$

where

$$\begin{cases}\Delta_1 = (\frac{2}{9}\|\mathbf{v}_m\| - \frac{\|\mathbf{v}_a\|}{4}l_{am})^2 - \frac{\|\mathbf{v}_a\|^2}{16}(l_{am}^2 - l_{ar}^2) & \text{(3.3a)} \\ \Delta_2 = (\frac{2}{3}\|\mathbf{v}_m\| - \frac{\|\mathbf{v}_a\|}{2}l_{am})^2 - \frac{\|\mathbf{v}_a\|^2}{4}(l_{am}^2 - l_{ar}^2) & \text{(3.3b)}\end{cases}$$

and $\mathbf{1}(e)$ is an indicator function returning 1 when e is true and 0 otherwise.

Proof. By setting $\mathbf{s}$ as the origin, and defining $\mathbf{a} = \mathbf{v}_a$, $\mathbf{b} = \mathbf{v}_m$, and $\mathbf{c} = \mathbf{v}_r$, we can apply Lemma 1 and Lemma 2 to derive all the expectations in Equation (3.1). Consequently, we obtain the following result

$$\left\{\begin{array}{ll} \frac{1}{3}\|\mathbf{b}\|^2 - \frac{1}{2}\mathbf{a}^\top\mathbf{b} \leq \frac{1}{3}\|\mathbf{c}\|^2 - \frac{1}{2}\mathbf{a}^\top\mathbf{c} & \text{(3.4a)} \\ \frac{1}{3}\|\mathbf{b}\|^2 - \frac{1}{2}\mathbf{a}^\top\mathbf{b} \leq \frac{1}{9}\|\mathbf{b}\|^2 + \frac{1}{9}\|\mathbf{c}\|^2 - \frac{1}{4}\mathbf{a}^\top\mathbf{b} - \frac{1}{4}\mathbf{a}^\top\mathbf{c} + \frac{1}{9}\mathbf{b}^\top\mathbf{c} & \text{(3.4b)} \end{array}\right\}$$

if Equation (3.1) holds. Note that the common terms related to $\|\mathbf{a}\|$ in Equation (3.4) are omitted, as they do not affect the inequalities.

We first prove that if inequalities in Equation (3.4) hold, then Equation (3.1) holds naturally, i.e.,

$$\frac{1}{3}\|\mathbf{c}\|^2 + \frac{1}{3}\|\mathbf{b}\|^2 - \frac{1}{2}\mathbf{a}^\top\mathbf{c} \geq \frac{1}{9}\|\mathbf{b}\|^2 + \frac{1}{9}\|\mathbf{c}\|^2 - \frac{1}{4}\mathbf{a}^\top\mathbf{b} - \frac{1}{4}\mathbf{a}^\top\mathbf{c} + \frac{1}{9}\mathbf{b}^\top\mathbf{c} \tag{3.5}$$

Assuming the above Equation (3.5) is false, then we have

$$\left\{\begin{array}{ll} \frac{2}{9}\|\mathbf{c}\|^2 - \frac{1}{4}\mathbf{a}^\top\mathbf{c} \leq \frac{1}{9}\|\mathbf{b}\|^2 - \frac{1}{4}\mathbf{a}^\top\mathbf{b} + \frac{1}{9}\mathbf{b}^\top\mathbf{c} & \text{(3.6a)} \\ \frac{2}{9}\|\mathbf{b}\|^2 - \frac{1}{4}\mathbf{a}^\top\mathbf{b} \leq \frac{1}{9}\|\mathbf{c}\|^2 - \frac{1}{4}\mathbf{a}^\top\mathbf{c} + \frac{1}{9}\mathbf{b}^\top\mathbf{c} & \text{(3.6b)} \end{array}\right\}$$

where Equation (3.6a) is from the assumed contradiction, and Equation (3.6b) is from the second inequality of Equation (3.4). Adding both sides gives $\frac{1}{9}\|\mathbf{c}\|^2 + \frac{1}{9}\|\mathbf{b}\|^2 < \frac{2}{9}\mathbf{b}^\top\mathbf{c}$ leading to $\|\mathbf{c} - \mathbf{b}\|^2 < 0$, which is impossible. Therefore Equation (3.5) must be true and hence we prove the equivalency between Equation (3.1) and Equation (3.4).

We now focus on Equation (3.4). Consider a general function in the following form

$$f_e(x \mid a, b, l) = ax^2 - b(l - e)x, \ (\forall x, a, b > 0, \ e \in \mathbb{R})$$

where a and b are fixed constants. Hereafter we omit $|a, b, l$ for simplicity, and some examples of function $f_0(x)$ and $f_e(x)$ are shown in Figure 4. We examine the conditions

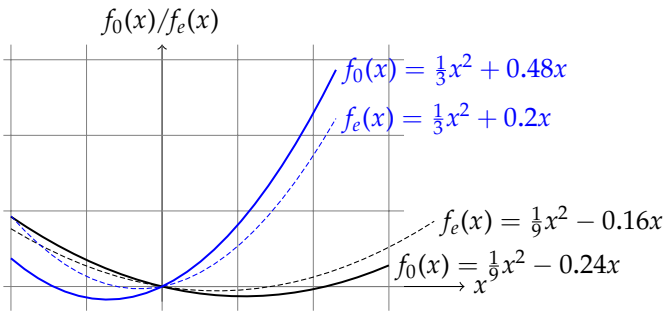


Figure 4
$f_0(x)$ and $f_e(x)$ function values.

for $f_e(y) \geq f_0(x)$ for given x and e but varying y. Since $a > 0$, $f_e(y) \geq f_0(x)$ is always possible and

$$f_e(y) \geq f_0(x) \Leftrightarrow y \in \begin{cases} \mathbb{R}^+ - \frac{1}{2a}\mathcal{B}(b(l-e)), \sqrt{\Delta}), \ \Delta \geq 0 \\ \mathbb{R}^+, \ \Delta < 0 \end{cases} \tag{3.7}$$

where $\Delta = 4af_0(x) + b^2(l-e)^2$ and $\mathcal{B}(x,r) = (x-r, x+r)$ is the disc at x with radius r without the boundary. The above is derived by solving $f_e(y) = f_0(x)$ for a fixed x. We identify Equation (3.4) to $f_e(x \mid \frac{1}{3}, \frac{1}{2}\|\mathbf{a}\|, l_{am})$ and $f_e(x \mid \frac{1}{9}, \frac{1}{4}\|\mathbf{a}\|, l_{am})$, for example, Equation (3.4a) can be rewritten as

$$f_0(\|\mathbf{b}\| \mid \frac{1}{3}, \frac{\|\mathbf{a}\|}{2}, l_{am}) \leq f_e(\|\mathbf{c}\| \mid \frac{1}{3}, \frac{\|\mathbf{a}\|}{2}, l_{am})$$

where e is to account for the difference between l_{am} and l_{ar} so that $l_{ar} = l_{am} - e$. By using Equation (3.7), we observe that

$$\text{(3.4a)} \Leftrightarrow y \in \mathbb{R}^+ - \frac{9}{8}\mathcal{B}(\|\mathbf{v}_a\| l_{ar}, 4\sqrt{\Delta_1}) \tag{3.8}$$

Similarly we have

$$\frac{1}{9}\|\mathbf{b}\|^2 - \frac{1}{4}\mathbf{a}^\top\mathbf{b} \leq \frac{1}{9}\|\mathbf{c}\|^2 - \frac{1}{4}\mathbf{a}^\top\mathbf{c} \Leftrightarrow y \in \mathbb{R}^+ - \frac{3}{4}\mathcal{B}(\|\mathbf{v}_a\| l_{ar}, 2\sqrt{\Delta_2}) \tag{3.9}$$

It is important to note that in both Equation (3.8) and Equation (3.9), we omit the case when Δ_i, for $i = 1, 2$, where $y \in \mathbb{R}^+$. It is now evident that the desired condition defines a subset of the intersection of the sets specified in Equation (3.8) and Equation (3.9) (appearing on the right-hand sides of the inequalities). By leveraging the fact that $x^2 \leq xyl_{mr}$ when $y \geq \frac{x}{l_{mr}}$ for $l_{mr} > 0$, we deduce that $\|\mathbf{b}\|^2 \leq \mathbf{b}^\top\mathbf{c}$ holds when $\|\mathbf{v_r}\| \geq \frac{\|\mathbf{v_m}\|}{l_{mr}}$. This completes our proof. □

Remark 1

Theorem 1 establishes that, when the condition in Equation (3.2) is satisfied, the reconstructed `[CLS]` token, obtained by leveraging only the `[MASK]` token and residual tokens, is closer to the original reconstructed `[CLS]` token (before the attack) than any other reconstructed version. This result is formalized in the inequality presented in Equation (3.1). Consequently, this validates the proposed *DDM* in identifying the suspicious token(s) for masking. In cases where multiple tokens are attacked, the analysis remains applicable due to the contraction in convex reconstruction. Specifically, the vectors **a** and **r** can be understood as contracted versions of the vector forms of the attacked/victim tokens and the replacement tokens, respectively. Notably, the condition in Equation (3.2) is relatively mild, primarily stating that the distributions of **a**, **m**, and **r** are centered around **s**. Although the underlying manifold may influence the token locations, as it directly impacts their spatial configuration, the analysis itself does not depend on any specific manifold geometry.

Remark 2

It is worth noting that replacing a normal token, incorrectly identified as adversarial, with `[MASK]` could introduce semantic information loss. As demonstrated later in the ablation study on the masking budget, when too many normal tokens are masked, the input suffers from semantic distortion, leading to degraded model performance. A potential solution could involve enhancing the detection of suspicious tokens or, alternatively, making a random injection of `[MASK]` tokens rather than substituting them. Detailed results are discussed in Section 5.3.

4. Experiments

4.1 Set-up

Datasets. To evaluate the effectiveness of *DDM*, we use four highly competitive text classification datasets for benchmarking adversarial defense methods: SST2 (Socher et al. 2013), AGNEWS (Zhang, Zhao, and LeCun 2015), IMDB (Maas et al. 2011), and MR (Pang and Lee 2005). The SST2 dataset comprises binary sentiment classification tasks derived from movie reviews, requiring the prediction of positive or negative sentiment. AGNEWS contains news articles categorized into four classes: World, Sports, Business, and Science/Technology. IMDB provides binary sentiment classification with significantly longer reviews compared to SST2. The MR dataset involves a four-class sentiment polarity classification task for reviews. These datasets span diverse domains, binary and multi-class classification settings, and varying text complexities. This diversity ensures a comprehensive evaluation of adversarial defense methods. Table 2 summarizes the dataset statistics.

Attacking Algorithms. We utilize four different attack strategies, implemented using the TextAttack framework (Morris et al. 2020b), including:

1. TextFooler (Jin et al. 2020) introduces word-level perturbations by replacing original words with their synonyms.
2. BERT-Attack (Li et al. 2020) applies word-level perturbations by leveraging a pre-trained masked language model to substitute candidate words.

Table 2
Statistics on the utilized text classification benchmarks, where #. Train, #. Test, #. Length, and #. Class represent, respectively, the number instances in the training and test sets, the average length of input text, and the number of class labels.

Dataset	#. Train	#. Test	#. Length	#. Class
SST2	67,791	1,821	19	2
AGNEWS	120,000	7,600	43	4
IMDB	25,000	25,000	215	2
MR	8,530	1,070	20	4

3. DeepWordBug (Gao et al. 2018) focuses on character-level perturbations, including letter substitutions, deletions, insertions, and swaps within words.
4. TextBugger (Li et al. 2019) combines both symbol- and word-level perturbations, utilizing techniques such as inserting spaces, replacing words, deleting characters, and swapping adjacent letters to craft adversarial examples.

Defense Algorithms. The proposed *DDM* is evaluated against state-of-the-art methods across three categories, as outlined below:

1. Data augmentation: FreeLB++ (Li et al. 2021), RMLM (Wang et al. 2023), Adv-Purification (Li, Song, and Qiu 2023), MVP (Raman et al. 2023), and FAT (Yang, Liu, and He 2024).
2. Model adaptation: InfoBERT (Wang et al. 2021), Flooding-x (Liu et al. 2022), RobustT (Zheng et al. 2022), ATINTER (Gupta et al. 2023), SIWCon (Zhan et al. 2023), LLMPM (Moraffah et al. 2024), and ROIC-DM (Yuan, Yuan, and He 2024).
3. Randomized smoothing: RanMASK (Zeng et al. 2023), RSMI (Moon et al. 2023), MI4D (Hu et al. 2023), and Text-RS (Zhang et al. 2024).

Among these methods, RMLM, Adv-Purification, MVP, RanMASK, RSMI, and MI4D also use masking strategies. A detailed review of these contender methods can be found in Section 2.3. The majority of baseline results are directly sourced from their respective original papers to ensure consistency, except for MI4D. Due to the unavailability of results, we reimplemented MI4D using its default configuration, including a 30% masking rate, a learning rate of $2e^{-5}$, and the minimum learning rate of $1e^{-6}$.

Implementation. For our main experiments (Section 4.2), we use the BERT-base model (Devlin et al. 2019) as the encoder. Training is conducted using a batch size of 32 sequences, each with a maximum length of 128 tokens. We reserve 10% of the training set for validation, and early stopping is applied if the validation accuracy does not improve with 10 epochs. A dropout rate of 0.1 is applied across all layers. We utilize the Adam optimizer with a learning rate that warms up to $5e^{-5}$ over the first 10,000 steps and then decays linearly to $1e^{-6}$ following a cosine annealing schedule. Gradient clipping is enforced within the range $(-1, 1)$. During inference, *DDM* masks suspicious tokens by using the FGWS strategy (Mozes et al. 2021) to estimate the suspicious score for each candidate token, where we calculate their occurrence frequency within the training dataset. For inputs with very short lengths, at least one `[MASK]` token is applied; otherwise, the masking budget (b_M) during both training and testing is set to 20% of the input length. All experiments are repeated five times using different random seeds, with the results averaged to ensure reliability. Finally, all computations are performed on an NVIDIA A100 GPU server.

Evaluation Metrics. Following the experimental setup outlined in previous work (Zhang et al. 2022; Hu et al. 2023; Yuan, Yuan, and He 2024), we select 1,000 samples that were successfully attacked, originally drawn at random from the testing set, to

evaluate the model robustness. The following metrics are employed: (1) CLA%, which represents the classification accuracy of the model on the original (clean) data; (2) CAA%, denoting the classification accuracy under specific adversarial attacks. A higher CAA% reflects better defense performance; and (3) SUCC%, which measures the success rate of adversarial attacks, defined as the proportion of examples successfully misclassified out of the total attack attempts. A lower SUCC% indicates greater robustness of the model.

4.2 Main Results

In addition to *existing defense methods*, the Baseline, implemented using the vanilla BERT-base model, is also employed as a reference. Table 3 summarizes the average results over five trials for adversarial defense performance, while detailed statistical results and the relevant computational complexity analysis of our method are provided in Appendix A. Key observations are as follows: (1) TextFooler is considered the most challenging adversarial attack due to its ability to generate semantically coherent perturbations. In contrast, character-based attacks exhibit lower effectiveness compared to word-based attacks. Additionally, datasets with lengthy inputs and larger training sizes, such as AGNEWS and IMDB, demonstrate greater resilience against adversarial attacks compared to shorter and smaller datasets like SST2 and MR. This difference is attributed to the richer contextual information and broader coverage of linguistic variations present in larger datasets, which enhance model robustness against adversarial perturbations. (2) Classification accuracy (CLA%): the proposed *DDM* maintains classification accuracy on clean test data, achieving performance comparable to the standard Baseline. For the SST2, IMDB, and MR datasets, *DDM* even achieves a slightly higher CLA% (91.9, 91.9, and 86.0) compared to the Baseline (91.6, 91.4, and 85.7). (3) Defense performance (CAA% and SUCC%): our proposed method consistently achieves superior or competitive results compared to state-of-the-art adversarial defense techniques across all evaluated datasets. Specifically, against the TextFooler attack, *DDM* achieves best CAA% values on three out of four datasets. Notably, it secures a CAA% of 76.3% on the IMDB dataset, ranking second only to Adv-Purification's 81.5%. For BERT-Attack, *DDM* demonstrates robust performance, achieving a CAA% of 70.3% on the SST2 dataset, slightly below the top-performing MVP's score of 75.9% and the second-best InfoBERT. However, *DDM* significantly outperforms other methods under different attack strategies, such as DeepWordBug, where it reports the highest CAA% values of 71.3%, 85.8%, and 68.7% on the SST2, AGNEWS, and MR datasets, respectively. This indicates the stability and effectiveness of *DDM* across diverse adversarial scenarios. When evaluated against TextBugger, *DDM* secures the leading position, achieving top CAA% scores of 71.5%, 83.3%, 81.8%, and 63.3%, surpassing all competing methods. On the other hand, the SUCC% scores, which quantify the success rate of adversarial attacks, are typically inversely proportional to the CAA% results. Consequently, our proposed method consistently demonstrates similar performance in terms of SUCC% scores. Clearly, while our method may not achieve the best defense performance against individual attack methods in certain cases (such as SST2 with BERT-Attack and IMDB with TextFooler), its average defense effectiveness across the four evaluated attack methods is the highest among all compared techniques.

We then focus on evaluating various masking strategies, denoted with $*$ in Table 3. Typically, existing methods that incorporate masking into data augmentation primarily adopt the *replace-then-predict* strategy (such as RMLM, Adv-Pur, and MVP), i.e., replacing specific tokens and predicting their context. However, this approach could

Table 3
Defense performance comparison between *DDM* and existing methods. Superscripts (1), (2), and (3) indicate the use of *data augmentation*, *model adaptation*, and *randomized smoothing* techniques, respectively. Additionally, methods using the masking strategy are further labelled with *. Avg. is the average result of CAA%. The best performance is highlighted in **bold**, while the second-best is underlined. Statistically significant results at $p < 10^{-3}$ are marked with †, while "–" indicates that no results are available in the original paper.

Datasets	Methods	CLA%	TextFooler		BERT-Attack		DeepWordBug		TextBugger		Avg.
			CAA%	SUCC%	CAA%	SUCC%	CAA%	SUCC%	CAA%	SUCC%	
SST2	Baseline	91.6	4.6	94.1	5.1	94.3	15.3	81.1	27.1	69.6	13.0
	FreeLB++(1)	92.3	42.2	52.6	72.0	22.1	51.1	44.5	63.0	32.8	57.1
	RMLM(1)*	87.9	52.6	39.5	18.5	78.7	–	–	–	–	46.1
	Adv-Pur(1)*	92.7	62.9	27.5	47.7	48.1	–	–	–	–	55.3
	MVP(1)*	91.7	44.6	49.1	**75.9**	**18.4**	–	–	65.1	28.5	61.9
	ROIC-DM(2)	**95.1**	55.4	39.5	46.4	49.7	–	–	–	–	50.9
	InfoBERT(2)	91.8	43.1	49.3	72.8	22.4	59.7	35.4	64.6	31.8	60.1
	ATINTER(2)	91.4	24.0	67.5	17.1	81.3	22.6	75.3	40.5	56.3	26.1
	Flooding-X(2)	91.9	34.9	62.4	27.7	70.7	34.5	62.4	51.7	45.3	37.2
	SIWCon(2)	91.2	19.3	74.4	11.7	87.2	32.0	65.1	30.9	66.1	23.5
	RobustT(2)	90.9	26.7	70.6	17.9	80.3	–	–	42.1	53.7	28.9
	RanMASK(3)*	90.0	12.9	86.1	11.4	87.7	27.5	70.3	39.9	57.0	22.9
	RSMI(3)*	91.7	53.5	41.7	45.6	50.3	59.3	35.3	70.1	23.6	57.1
	MI4D(3)*	92.0	44.1	53.2	42.7	54.0	54.7	42.2	67.0	29.8	52.1
	DDM	91.9†	**63.3**†	**26.5**†	70.3†	25.2†	**71.3**†	**21.8**†	**71.5**†	**22.8**†	**69.1**†
AGNEWS	Baseline	92.8	19.1	79.6	27.2	71.0	16.6	82.4	23.5	75.0	21.6
	FreeLB++(1)	93.3	51.5	46.0	41.8	56.2	55.1	42.1	55.9	41.4	51.1
	RMLM(1)*	94.0	81.0	13.7	48.1	48.7	–	–	–	–	64.6
	Adv-Pur(1)*	92.0	61.5	27.9	49.7	46.5	–	–	–	–	55.6
	MVP(1)*	93.7	46.3	49.3	82.1	11.4	–	–	66.0	27.4	64.8
	FAT(1)	**95.1**	62.3	33.2	48.0	47.4	–	–	63.6	32.4	58.0
	ROIC-DM(2)	94.1	78.7	18.4	49.0	47.0	–	–	–	–	63.9
	InfoBERT(2)	93.2	44.0	52.2	80.7	15.4	53.9	42.4	64.1	32.9	60.7
	ATINTER(2)	92.6	73.0	21.1	22.9	75.3	21.7	76.6	63.9	32.5	45.4
	Flooding-X(2)	92.7	42.4	54.9	27.4	71.0	54.1	42.8	62.2	34.0	46.5
	SIWCon(2)	92.6	19.7	78.7	23.1	75.1	20.6	77.8	52.9	42.9	29.1
	RobustT(2)	94.9	28.5	70.0	12.1	87.2	–	–	53.4	43.7	31.3
	LLMPM(2)	**95.1**	81.3	13.4	–	–	–	–	–	–	81.3
	RanMASK(3)*	92.6	37.9	58.7	49.5	46.1	38.4	54.6	45.0	50.9	42.7
	RSMI(3)*	92.9	71.7	22.8	67.5	27.3	72.8	21.6	69.7	25.0	70.4
	MI4D(3)*	93.0	67.3	29.1	70.1	25.6	66.7	31.0	75.1	20.3	69.8
	DDM	92.8†	**82.3**†	**10.6**†	**83.6**†	**9.8**†	**85.8**†	**8.7**†	**83.3**†	**10.1**†	**83.8**†
IMDB	Baseline	91.4	10.3	88.8	5.8	93.7	12.3	83.4	5.3	94.3	8.4
	FreeLB++(1)	92.3	45.3	51.0	39.9	56.9	78.3	16.1	42.9	53.6	51.6
	RMLM(1)*	92.3	54.7	39.4	32.5	64.0	–	–	–	–	43.6
	Adv-Pur(1)*	94.1	**81.5**	**11.8**	76.7	16.3	–	–	–	–	79.1
	FAT(1)	**95.0**	70.8	21.3	55.1	34.7	–	–	75.0	18.3	70.0
	InfoBERT(2)	91.8	16.9	81.6	15.8	82.8	62.3	32.1	37.6	59.0	33.2
	ATINTER(2)	91.2	13.9	84.8	23.6	74.1	52.5	42.4	25.2	72.4	28.8
	Flooding-X(2)	91.6	40.5	58.5	32.3	65.8	77.9	18.5	62.3	35.8	53.3
	SIWCon(2)	91.7	10.3	85.9	20.1	78.1	22.3	74.1	16.8	81.7	17.4
	RobustT(2)	93.8	55.6	40.7	55.2	41.2	–	–	57.6	38.6	56.1
	LLMPM(2)	94.5	73.5	19.8	–	–	–	–	–	–	73.5
	RanMASK(3)*	91.1	22.0	74.6	36.0	58.4	66.0	28.7	18.0	79.2	35.5
	Text-RS(3)	91.2	–	–	38.3	55.7	–	–	–	–	38.3
	RSMI(3)*	91.3	54.8	40.0	60.1	34.2	56.4	38.2	51.5	43.6	55.7
	MI4D(3)*	91.8	56.6	40.3	63.5	33.1	**89.9**	**5.8**	74.3	21.0	71.1
	DDM	91.9†	76.3†	17.4†	**79.6**†	**13.4**†	84.2†	8.5†	**81.8**†	**11.3**†	**80.5**†
MR	Baseline	85.7	6.5	92.4	8.7	89.8	19.4	77.4	26.3	69.3	15.2
	FreeLB++(1)	**86.4**	10.8	87.5	11.2	87.0	22.3	74.2	31.8	63.2	19.0
	ATINTER(2)	85.6	21.1	74.6	19.3	77.5	30.6	64.3	45.7	47.8	29.2
	SIWCon(2)	85.4	30.7	64.2	17.6	79.4	60.3	36.1	36.8	56.9	36.4
	RanMASK(3)*	85.1	12.9	84.8	16.8	80.3	19.6	77.0	33.1	61.1	20.6
	RSMI(3)*	86.1	47.6	44.7	39.5	54.1	58.1	32.5	56.4	34.5	50.4
	MI4D(3)*	85.8	45.7	47.1	50.8	44.4	59.5	35.3	54.2	40.3	52.6
	DDM	86.0†	**55.2**†	**37.9**†	**62.6**†	**29.9**†	**68.7**†	**19.6**†	**63.3**†	**26.5**†	**62.5**†

introduce substantial noise due to incorrect predictions. Similarly, when integrating masking with randomized smoothing, existing techniques also rely on random masking to produce multiple variants for ensemble prediction. In contrast, *DDM* uses distinct masking strategies tailored for adversarial defense. During training, `[MASK]` tokens are inserted at the beginning of clean sequences as a placeholder to prepare the model for perturbations. Compared with the replace-then-predict strategy, this approach reduces noise caused by prediction errors. During inference, *DDM* substitutes suspicious tokens with `[MASK]` to mitigate adversarial influence, eliminating the need for random-based ensemble mechanisms. Overall, our approach consistently achieves the best performance across diverse attack types on average, demonstrating absolute improvements in CAA% by 7.2, 2.5, 1.4, and 12.1 points over state-of-the-art methods.

4.3 Ablation Study

To provide a comprehensive analysis and deepen the understanding of the proposed *DDM*, we conduct ablation studies to address the following research questions:

- **Effectiveness across encoders**: How does *DDM* perform when integrated with different encoder architectures?
- **Adversarial detection capability**: How accurately can the detection mechanism identify adversarially perturbed tokens? How does its performance compare when combined with other detection methods?
- **Masking strategy evaluation**: How effective is the proposed masking strategy? How does it compare to alternative masking strategies?
- **Masking budget**: How does the number of applied `[MASK]` affect model performance? What is the optimal number of masks required?
- **Performance of different maskers**: What are the impacts of employing different types of maskers on adversarial defense?

As in the main experiment, all reported results are averaged over five independent experimental runs to ensure consistency.

Effectiveness across Encoders. To validate the generalizability of *DDM* across different encoders, we use a RoBERTa-based model (Liu et al. 2019) as the encoder while maintaining consistency with the BERT-base configuration in all other settings, such as batch size and training epochs. For comparative analysis, we select top performing defense methods from Table 3, including MVP and Adv-Purification for data augmentation, Flooding-X for model adaptation, and RSMI and MI4D for randomized smoothing. The vanilla RoBERTa model is also used for reference. To ensure consistency and fairness in evaluation, baseline results are sourced from their respective original publications if available.

Results in Table 4 highlight the consistent efficacy of *DDM* across diverse encoders, datasets, and attack scenarios. For instance, on the SST2 dataset, *DDM* achieves the highest CAA% in three out of four attack scenarios. Similarly, on AGNEWS, IMDB, and MR, *DDM* achieves the best average CAA% (83.0, 82.2, and 63.4, respectively), demonstrating superior robustness across varying textual domains. In contrast, alternative methods such as Adv-Purification show significant performance fluctuations when

Table 4
Performance comparison of *DDM* with a RoBERTa-based encoder. The best results are marked in **bold**, and the second-best are underlined. Baseline results are directly cited from the respective original papers, and a "−" indicates unavailable results.

Datasets	Methods	CLA%	TextFooler		BERT-Attack		DeepWordBug		TextBugger		Avg.
			CAA%	SUCC%	CAA%	SUCC%	CAA%	SUCC%	CAA%	SUCC%	
SST2	RoBERTa	94.1	5.4	94.3	6.2	93.4	17.0	81.9	29.7	68.4	14.6
	MVP	93.9	46.9	48.5	**78.1**	**18.4**	–	–	69.8	25.5	64.9
	Flooding-X	94.2	32.2	65.8	35.4	62.4	38.2	59.4	49.9	47.0	38.9
	RSMI	94.2	52.1	44.7	33.5	64.4	61.9	34.3	60.6	35.7	52.0
	MI4D	**94.3**	36.4	61.4	34.5	63.4	45.6	51.6	58.3	38.2	43.7
	DDM	94.2	**68.5**	**27.8**	71.6	25.8	**70.8**	**24.8**	**73.4**	**23.2**	**71.1**
AGNEWS	Roberta	94.2	15.8	83.2	26.7	71.7	33.0	65.0	49.2	47.8	31.2
	Adv-Purification	**96.1**	34.2	63.3	19.5	77.1	–	–	–	–	26.9
	MVP	94.5	51.5	46.3	**85.3**	**11.5**	–	–	68.7	27.1	68.5
	Flooding-X	94.4	68.9	27.0	56.4	40.3	65.3	30.8	70.3	25.5	65.2
	RSMI	94.6	74.1	21.5	66.3	29.9	68.8	27.3	72.6	23.3	70.5
	MI4D	94.2	66.7	29.2	69.7	26.0	62.4	33.8	73.9	21.5	68.2
	DDM	95.1	**78.8**	**16.3**	84.9	12.5	**86.0**	**8.7**	**82.4**	**12.5**	**83.0**
IMDB	Roberta	91.5	0.5	99.4	0.6	99.3	48.5	47.0	11.9	87.0	15.4
	Adv-Purification	**96.1**	54.3	43.1	52.2	43.7	–	–	–	–	53.3
	Flooding-X	94.7	48.5	48.8	33.4	64.7	83.1	12.2	62.3	34.2	56.8
	RSMI	91.2	73.4	21.1	60.5	33.7	86.9	4.7	75.3	17.4	74.0
	MI4D	94.5	56.2	40.5	54.2	42.6	**93.6**	**1.0**	69.8	26.1	68.5
	DDM	94.9	**80.3**	**12.2**	**79.4**	**16.2**	84.3	7.9	**84.9**	**7.2**	**82.2**
MR	Roberta	85.9	6.0	93.0	6.6	92.3	15.9	81.5	29.8	65.3	14.6
	RSMI	86.8	**58.1**	**33.1**	45.3	47.8	62.6	27.9	64.3	25.9	57.6
	MI4D	87.6	51.9	40.8	44.7	49.0	61.9	29.3	58.6	33.1	54.3
	DDM	**87.9**	56.1	35.0	**66.0**	**29.4**	**66.8**	**22.2**	**64.7**	**24.7**	**63.4**

switching between encoders. For example, Adv-Purification with BERT attains average CAA% scores of 55.6 (AGNEWS) and 79.1 (IMDB), compared with 26.9 and 53.3 with the RoBERTa encoder. These fluctuations show the sensitivity to encoder configurations, whereas *DDM* maintains strong stability and consistent efficacy regardless of the underlying encoder. To ensure alignment with the primary results, subsequent experiments adopt the BERT-base encoder for consistency.

Adversarial Detection Capability. To mitigate adversarial attacks during inference, the proposed *DDM* identifies and masks suspicious tokens within each input. This section presents a comprehensive analysis of how token-level detection accuracy influences overall defense performance. Specifically, three detection methods are considered: WDR (which identifies adversarial tokens by selectively removing words and measuring prediction shifts [Mosca et al. 2022]), MLMD (which leverages Masked Language Models to detect tokens that cause substantial changes in the representation manifold [Zhang et al. 2023]), and FGWS (the frequency-based heuristic proposed in Mozes et al. [2021]). For each method, the top 20% of suspicious tokens per input are masked. Evaluations are conducted across four benchmark datasets (i.e., SST2, AGNEWS, IMDB, and MR) under four adversarial attacks (i.e., TextFooler, BERT-Attack, DeepWordBug, and TextBugger).

To quantify detection effectiveness, token-level performance is measured by computing True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN), where adversarial tokens are treated as the positive class and normal tokens as the negative class. This setup reflects the objective of accurately identifying

adversarial perturbations while preserving clean input tokens. Accordingly, the following evaluation metrics are reported:

- **Recall** (= TP/(TP + FN)): the proportion of adversarial tokens correctly identified;
- **False Positive Rate** (**FPR**= FP/(FP + TN)): the proportion of normal tokens incorrectly identified as adversarial;
- **False Negative Rate** (**FNR**= FN/(FN + TP)): the proportion of adversarial tokens that remain undetected (i.e., left unmasked);
- **Weighted F1** (**WF1**): an aggregated F1 score that accounts for class imbalance as adversarial tokens are typically sparse compared to normal ones.

In addition, the corresponding CAA (i.e., the final classification accuracy after applying token-level masking) is also reported to reflect the extent to which the detection influences model robustness under adversarial attacks.

The averaged results from five runs are summarized in Table 5, revealing several notable trends.

First, we observe that detection methods with higher Recall scores (i.e., lower FNR, since Recall + FNR = 1) consistently achieve superior CAA performance across various datasets and attack types. For example, on the AGNEWS dataset under the TEXTFOOLER attack, MLMD achieves a Recall of 98.5%, an FNR of 1.5%, and a peak CAA of 88.7%. In contrast, FGWS yields a lower Recall of 91.8%, a higher FNR of 8.2%, and a reduced CAA of 82.3%. Notably, higher Recall and lower FNR values are attributed to an increased number of TP and a reduced number of FN, highlighting the importance of accurately identifying and removing adversarial tokens. This observation is also consistent with the analysis in Equation (3.1), which examines the perturbation effects associated with both TP and FN cases. As a result, the model is better able to maintain stable predictions and achieve improved overall robustness.

Second, detection methods with lower FNR scores are often associated with lower FPR scores, and, consequently, tend to exhibit improved defense performance. There are two cases, however, that reveal the impact of FNR and FPR. Specifically, on the IMDB dataset under the TEXTFOOLER attack, both WDR and MLMD report the same FPR of 14.4%. Nevertheless, MLMD, which achieves a lower FNR, also yields higher CAA scores. One intuitive explanation is that highly important tokens are often the primary targets of adversarial attacks and are thus more likely to be identified as adversarial. Consequently, mistakenly masking a small number of less-important normal tokens (FP) tends to have a smaller impact on the overall semantics than leaving adversarial tokens unmasked (FN). That said, this does not imply that FP can be overlooked. As demonstrated later in the ablation study on the *masking budget*, excessively masking normal tokens (i.e., high masking rates) introduces semantic distortion, ultimately degrading model performance. In other words, while minimizing FN is crucial for adversarial robustness, limiting FP is also essential to preserve semantic meaning and maintain model robustness.

Finally, under the same dataset and attack settings, we also observe a strong correlation between WF1 and CAA. For instance, on the AGNEWS dataset under the TEXTFOOLER attack, MLMD achieves a WF1 of 90.1% and a CAA of 88.7%, whereas FGWS, with a lower WF1 of 82.1%, yields a lower CAA (82.3%) result.

Table 5
Performance comparison of various detection methods under four adversarial attacks, evaluated by detection effectiveness (Recall, Error Rates, Weighted F1) and classification accuracy (CAA).

Attack	Dataset	Method	TP	TN	FP	FN	Recall% (↑)	FPR% (↓)	FNR% (↓)	WF1% (↑)	CAA% (↑)
TextFooler	SST2	WDR	2524.1	14212.7	1276.4	986.8	71.9	8.2	28.1	88.3	66.7
		MLMD	2566.4	14254.9	1234.4	944.2	73.1	8.0	26.9	88.7	67.0
		FGWS	2425.0	14115.0	1375.0	1085.0	69.1	8.9	30.9	87.2	63.3
	AGNEWS	WDR	7407.3	33976.2	1402.4	221.9	97.1	4.0	2.9	96.3	87.5
		MLMD	7203.8	34375.5	1309.3	108.2	98.5	3.6	1.5	96.8	88.7
		FGWS	7144.0	33574.0	1654.0	636.0	91.8	4.7	8.2	94.8	82.3
	IMDB	WDR	14209.2	171152.3	28790.8	840.8	94.4	14.4	5.6	89.0	84.5
		MLMD	14252.4	171205.2	28747.7	797.7	94.7	14.4	5.3	89.0	86.5
		FGWS	13365.0	170314.0	29634.0	1684.0	88.8	14.8	11.2	88.4	76.3
	MR	WDR	2356.2	14959.2	1643.8	1043.8	69.3	9.9	30.7	87.0	62.5
		MLMD	2587.4	15183.4	1412.6	812.6	76.1	8.5	23.9	89.2	65.3
		FGWS	2145.0	14748.0	1854.0	1254.0	63.1	11.2	36.9	84.9	55.2
BERT-Attack	SST2	WDR	1626.3	14716.1	2173.7	482.7	77.1	12.9	22.9	87.7	71.5
		MLMD	1683.6	14776.2	2116.4	425.4	79.8	12.5	20.2	88.2	73.1
		FGWS	1618.0	14710.0	2181.0	490.0	76.7	12.9	23.2	87.6	70.3
	AGNEWS	WDR	6077.9	33855.6	2522.1	544.1	91.8	6.9	8.2	93.2	83.3
		MLMD	6197.4	33974.2	2402.6	424.6	93.6	6.6	6.4	93.8	84.2
		FGWS	6117.0	33898.0	2482.0	504.0	92.4	6.8	7.6	93.4	83.6
	IMDB	WDR	10078.5	171325.5	32921.5	671.5	93.8	16.1	6.2	88.4	85.7
		MLMD	10402.6	171651.2	32598.1	348.7	96.8	15.9	3.2	88.6	88.5
		FGWS	9449.0	170704.0	33550.0	1300.0	87.9	16.4	12.1	88.0	79.6
	MR	WDR	1852.0	15412.9	2147.9	587.8	75.9	12.2	24.1	87.7	67.3
		MLMD	1970.1	15531.3	2029.9	469.9	80.7	11.6	19.3	88.7	68.9
		FGWS	1745.0	15306.0	2254.0	694.0	71.5	12.8	28.5	86.7	62.6
DeepWordBug	SST2	WDR	2568.4	14177.4	1231.6	1022.6	71.5	8.0	28.5	88.3	65.5
		MLMD	2847.7	14456.7	952.3	743.3	79.3	6.2	20.7	91.2	73.5
		FGWS	2795.0	14404.0	1005.0	796.0	77.8	6.5	22.2	90.6	71.3
	AGNEWS	WDR	3299.4	34044.2	5301.1	355.7	90.3	13.5	9.7	88.9	82.8
		MLMD	3435.1	34181.4	5164.2	219.1	94.0	13.1	6.0	89.5	84.4
		FGWS	3483.0	34229.0	5116.0	171.0	95.3	13.0	4.7	89.7	85.8
	IMDB	WDR	10659.6	170834.6	32340.4	1165.4	90.2	15.9	9.9	88.2	82.7
		MLMD	11065.8	171240.8	31934.2	759.2	93.6	15.7	6.4	88.5	85.4
		FGWS	10538.0	170715.0	32461.0	1286.0	89.1	16.0	10.9	88.1	84.2
	MR	WDR	1784.1	15583.5	2215.8	415.9	81.1	12.5	18.9	88.5	70.3
		MLMD	1812.2	15611.9	2187.8	387.1	82.4	12.3	17.6	88.5	70.7
		FGWS	1727.0	15528.0	2273.0	473.0	78.5	12.8	21.5	87.8	68.7
TextBugger	SST2	WDR	2205.9	14555.9	1594.1	644.1	77.4	9.9	22.6	88.9	70.7
		MLMD	2131.8	14481.8	1668.2	718.2	74.8	10.3	25.2	88.1	68.5
		FGWS	2226.0	14576.0	1574.0	624.0	78.1	9.7	21.9	89.0	71.5
	AGNEWS	WDR	3922.8	34122.8	4677.2	277.2	93.4	12.1	6.6	90.1	86.1
		MLMD	3990.1	34189.5	4610.3	210.1	95.0	11.9	5.0	90.4	86.9
		FGWS	3834.0	34036.0	4765.0	365.0	91.3	12.3	8.7	89.8	83.3
	IMDB	WDR	13958.2	171123.3	29041.8	876.7	94.1	14.5	5.9	88.9	86.1
		MLMD	14175.3	171340.3	28824.7	659.7	95.6	14.4	4.4	89.1	87.3
		FGWS	13602.0	170767.0	29399.0	1234.0	91.7	14.7	8.3	88.7	81.8
	MR	WDR	1667.6	15467.6	2332.4	532.4	75.8	13.1	24.2	87.4	66.1
		MLMD	1793.6	15591.8	2207.3	407.3	81.5	12.4	18.5	88.5	69.4
		FGWS	1632.0	15427.0	2373.0	570.0	74.1	13.3	25.9	87.0	63.3

In summary, for token-level detection methods, those that achieve high Recall, low FNR/FPR, and high WF1 performance tend to provide more effective adversarial defense, indicating a strong correlation between accurate adversarial detection and the preservation of both semantic integrity and model robustness.

Masking Strategy Evaluation. *DDM* utilizes two distinct masking strategies during the training and inference stages. To evaluate their effectiveness, we enumerate alternative masking approaches, including:

- **RO-S**: Strategically replace tokens with `[Mask]` only, without predicting the masked content.
- **RO-R**: Randomly replace tokens with `[Mask]` only, without predicting the masked content.
- **RP-S**: Strategically replace tokens and further substitute them with predictions for the masked content.
- **RP-R**: Randomly replace tokens and further substitute them with predictions for the masked content.
- **IM**: Insert `[Mask]` at the beginning of the input sequence.

For the variants RP-S and RO-S, the FGWS strategy is adopted. That is, tokens are ranked by their frequency of occurrence in the training dataset, and least-frequent tokens are to be replaced. In the RO-S method, selected tokens are directly replaced with the `[Mask]` token. On the other hand, the RP-S method takes an additional step: After replacing selected tokens with `[Mask]`, it utilizes a pre-trained MLM to predict and replace those masked tokens. Similarly, the RO-R and RP-R methods randomly select tokens for replacement, instead of relying on FGWS. The replacement process mirrors that of RO-S and RP-S, but RO-R directly replaces the tokens with `[Mask]`, while RP-R uses an MLM to predict and substitute `[Mask]` with predicted tokens. Additionally, the IM method inserts a `[Mask]` token at the beginning of the sequence. In all masking based scenarios, we set the masking budget $b_M = 20\%$. At last, for comparison purpose, we also adopt the vanilla model (i.e., NO) where no masking is used.

We systematically evaluate five masking strategies during both the training and inference phases and present the corresponding defense performance (CAA%) in Figure 5. Not surprisingly, the worst performance occurs when no masking mechanism is utilized during training or inference (i.e., the NO model), leaving the model highly susceptible to adversarial manipulations.

Then, analyzing the training strategies, we observe that models fine-tuned with RO-S and RP-S (i.e., the 2^{nd} and 4^{th} rows) exhibit superior robustness compared to their counterparts, RO-R and RP-R (i.e., the 3^{rd} and 5^{th} rows). This improvement is attributed to the strategic token selection in RO-S and RP-S, which prioritizes masking or replacing the least important tokens. As a result, this approach effectively preserves critical contextual information in the original (clean) training samples. In contrast, random token selection (i.e., RO-R and RP-R) risks masking or altering semantically important tokens, leading to degraded training performance. Additionally, the IM model, which directly inserts `[Mask]` tokens, achieves the best average training performance. This outcome is due to its ability to maintain the original semantic structure while minimizing the introduction of noise.

On the other hand, a similar pattern is observed during the inference phase. Specifically, strategic masking or replacement (i.e., the 2^{nd} and 4^{th} columns) consistently outperforms their random-selection-based counterparts (i.e., the 3^{rd} and 5^{th} columns). While IM performs better than RO-R and RP-R, it still retains adversarial tokens, potentially introducing noise into the subsequent classification process. By contrast, RO-S

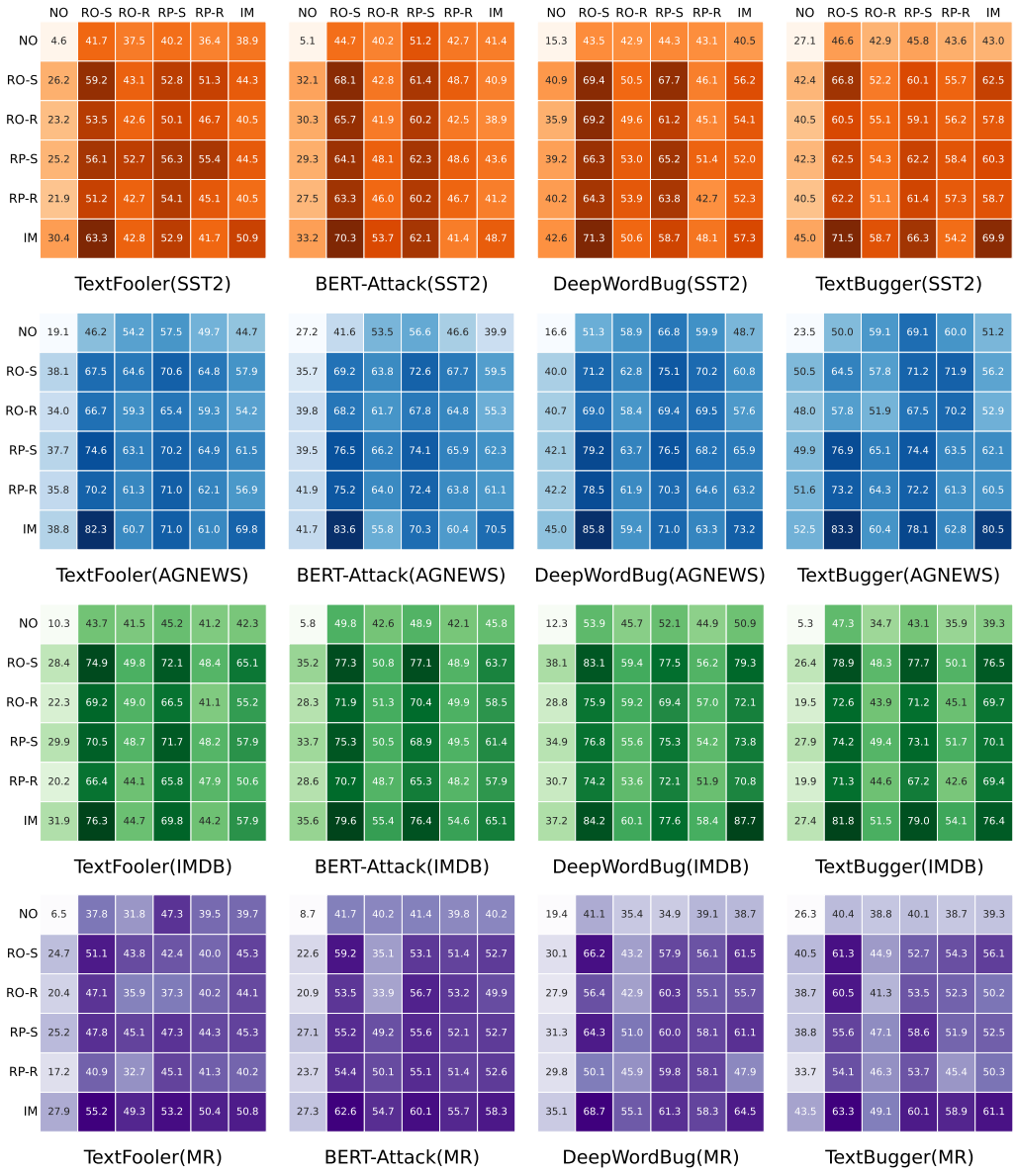


Figure 5
Defense performance (CAA%) comparison for different masking strategies. Darker colors indicate better performance, while lighter colors correspond to worse outcomes. The horizontal axis represents masking strategies used during training, and the vertical axis denotes those applied during inference.

explicitly removes suspicious tokens, preventing them from influencing downstream predictions. Meanwhile, RP-S also eliminates suspicious tokens but remains sensitive to the quality of token predictions; inaccurate predictions may mislead the classifier, resulting in suboptimal performance. Therefore, RO-S demonstrates the most reliable performance, as it not only eliminates adversarial perturbations but also bypasses the token predictions.

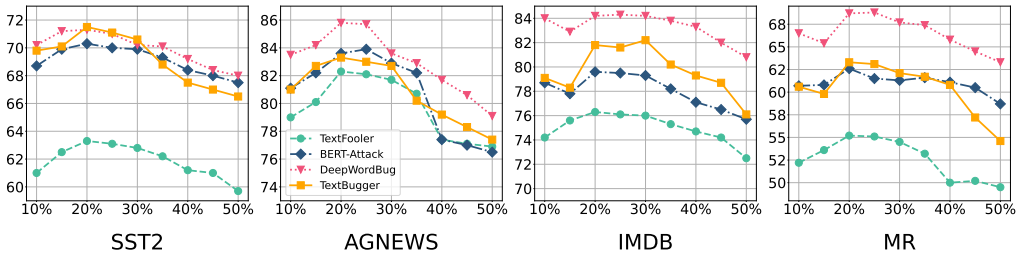


Figure 6
Model performance (CAA%) as a function of the masking budget (b_M) against four adversarial attack methods across four benchmarks.

The above analysis justifies our choice of the proposed *DDM*, which integrates a hybrid masking strategy tailored for different stages. During training, the IM mechanism inserts `[MASK]` tokens into clean samples, maintaining the original semantic structure while minimizing the introduction of noise. During inference, the RO-S mechanism dynamically identifies and removes adversarial tokens, preserving the semantic integrity of the input. By leveraging this dual-phase architecture, *DDM* effectively enhances the model robustness across diverse adversarial scenarios.

Masking Budget. The following section analyzes the impact of the masking ratio (b_M) on the performance of the proposed method. Increasing b_M introduces more `[MASK]` tokens in the input sequence, and vice versa. To evaluate this effect, we systematically vary b_M from 10% to 50% of the total number of tokens in an input, in increments of 5%.

The results presented in Figure 6 demonstrate that the proposed method maintains strong performance across different masking budgets, in particular when $15\% \leq b_M \leq 30\%$. Specifically, at a 10% masking rate, the model's performance is moderate, likely due to insufficient coverage of adversarial tokens. As the masking rate increases, performance progressively improves, generally peaking around 20% in most cases. This improvement can be attributed to a balanced trade-off between the removal of adversarial tokens and the preservation of sufficient contextual information from the original input. In certain scenarios, for example in AGNEWS and IMDB, optimal performance is observed at higher masking rates (i.e., 25% or 30%), suggesting that some datasets might require more `[MASK]` to effectively neutralize adversarial effects. However, with $b_M > 30\%$, further increases lead to a decline in performance. This decline is likely caused by excessive masking disrupting the original input semantics, thereby reducing the overall contextual coherence necessary for accurate predictions. Therefore, it is recommended to set $b_M = 20\%$, as this setting consistently achieves optimal defense performance across diverse scenarios.

While this configuration proves effective in our experiments, the optimal masking budget b_M is inherently dataset-dependent, influenced by factors such as average sequence length, linguistic complexity, and task-specific tolerance to token-level perturbations. For instance, shorter inputs tend to benefit from lower masking ratios to preserve core semantics, whereas longer documents may favor larger values of b_M to effectively mitigate adversarial attacks. To eliminate the manual tuning of the masking budget, one could use the *adaptive masking* scheme that determines b_M on-the-fly using the following complementary strategies: (1) *curriculum schedule*: A global masking ratio can be set to decay linearly over training epochs (Ankner et al. 2024), ensuring the model is first exposed to heavily masked instances and gradually transitions to lightly masked

ones. (2) *detection confidence or token-level suspiciousness scores*: Rather than relying on a fixed masking budget, masking decisions can also be guided by the relative ranking of token-level detection scores (e.g., thresholding accumulative probability derived from the scores) instead of absolute value cut-off. This approach adapts to the suspiciousness-score distribution of each token, allowing for context-sensitive masking without the need for manual hyperparameter tuning.

These heuristics offer promising directions for eliminating the need for a fixed masking budget b_M by dynamically aligning masking behavior with input characteristics. Exploring and refining such strategies remains an important direction for future work to enhance adversarial robustness across diverse datasets.

Performance of Different Maskers. *DDM* uses the `[MASK]` for token replacement during adversarial defense. To examine the impact of alternative token choices, we evaluate the effectiveness of using `[PAD]` (used for input length uniformity) and `[UNK]` (used for handling out-of-vocabulary tokens) as replacements for `[MASK]`. This analysis aims to explore the influence of token selection on adversarial defense performance. All other configurations remain constant (such as the encoder and training settings), with only the replacement token (`[PAD]` or `[UNK]`) differing in place of `[MASK]`.

The comparison results are shown in Figure 7. The variant using different replacement tokens achieves comparable performance. For instance, on the IMDB dataset under the BERT-Attack, the `[MASK]`, `[PAD]`, and `[UNK]` tokens achieve 79.6%, 79.3%, and 77.6% in terms of CAA, respectively. This suggests that the choice of token has limited impact under certain adversarial conditions. We further notice that, in specific cases, the `[PAD]` token even demonstrates superior defensive performance than that of `[MASK]`, as evidenced by SST2 under DeepWordBug attack and AGNEWS under TextFooler/TextBugger attack. The comparable performance of the `[PAD]`, `[MASK]`, and `[UNK]` tokens arises from their role as neutral placeholders within the proposed *DDM*. During inference, replacing adversarially perturbed tokens with these placeholders migrate the impact of perturbations, without significantly disrupting the model's comprehension of the remaining context. Consequently, the choice of token minimally affects performance under adversarial conditions.

5. Discussion

5.1 Robustness Against Semantically Preserving Attacks

This section evaluates the adversarial resilience of the proposed method against *semantically preserving* perturbations (Morris et al. 2020a; Wang et al. 2022; Liu et al. 2023). Specifically, the SemAttack framework (Wang et al. 2022) is used, in which inputs are perturbed in the embedding space while sentence-level semantic similarity is maintained. Following its configuration, at most 5 tokens are allowed to be perturbed per example. Substitutions are restricted to the Top-10 nearest neighbors in the latent space, provided that their cosine similarity with the original token exceeds 0.7. An attack is considered successful when it both alters the model prediction and preserves a sentence-level semantic similarity of at least 0.9 between the original and perturbed examples. Subsequently, experiments are conducted on all four benchmarks, i.e., SST-2, IMDB, AG NEWS, and MR, to ensure a comprehensive evaluation under semantically consistent adversarial scenarios.

The proposed method is compared with two strong defense baselines: LSDefense (Yang et al. 2023), in which label smoothing is applied to reduce reliance on discrete

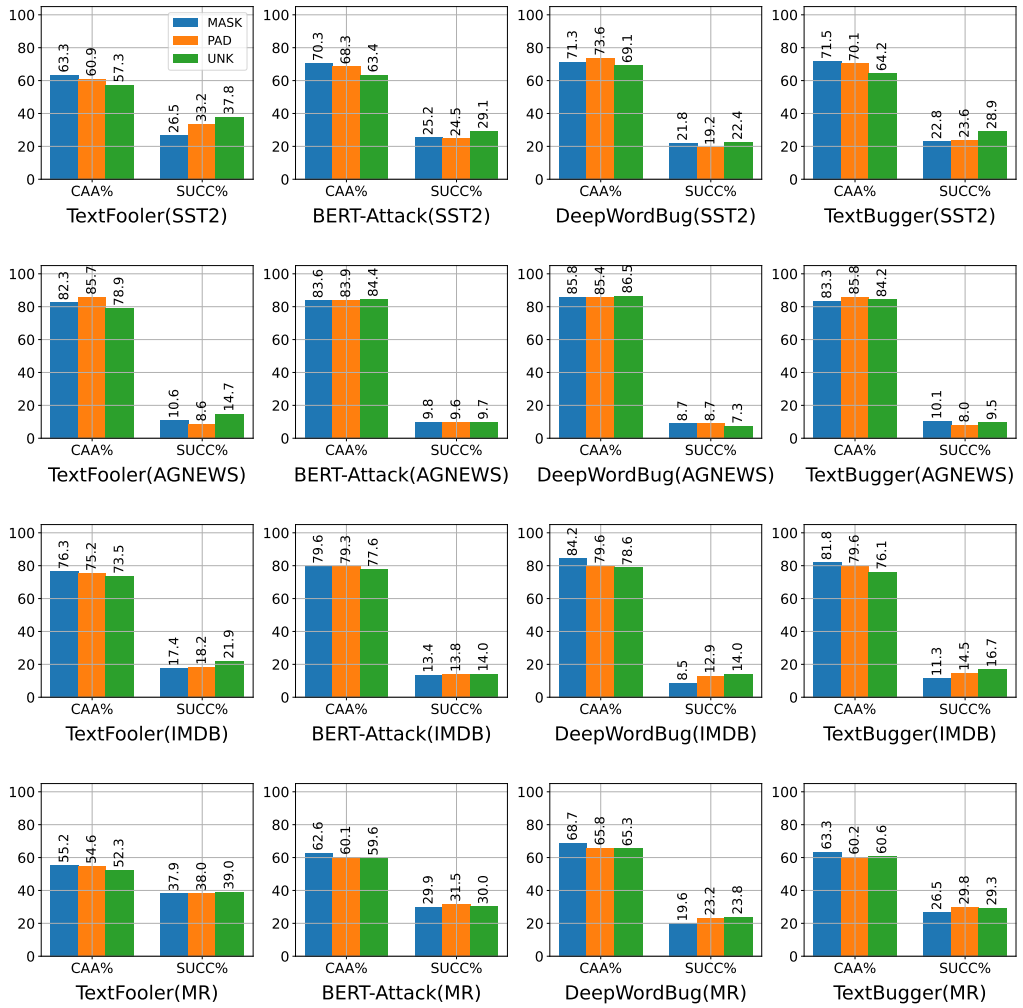


Figure 7
Impact analysis of the marker. Blue, orange, and green bar represents *DDM* using the `[MASK]`, `[PAD]`, and `[UNK]` token for masking.

tokens and enhance semantic robustness; and MI4D (Hu et al. 2023), a top-performing method evaluated under the standard adversarial setting (as shown in Table 3). For fairness, LSDefense results are reported directly from the original paper, whereas MI4D is re-implemented under the previous experimental setup. For *DDM*, all hyperparameters are kept consistent with those in Section 4.2, including the use of BERT-BASE as the backbone, frequency-based token detection, and a masking budget of $b_M = 20\%$.

As observed from Figure 8, *DDM* consistently outperforms the baselines across all four datasets, achieving the highest average CAA score of 61.7%, compared with 54.2% of *LSDefense* and 54.8% of *MI4D*. This gain is likely attributed to two key factors: (1) the introduced masking enables selective suppression of high-risk tokens while preserving the remaining informative content; and (2) suspicious tokens are masked rather than substituted, which avoids introducing semantically incompatible replacements. Such a strategy preserves sentence fluency and mitigates unintended semantic drift introduced

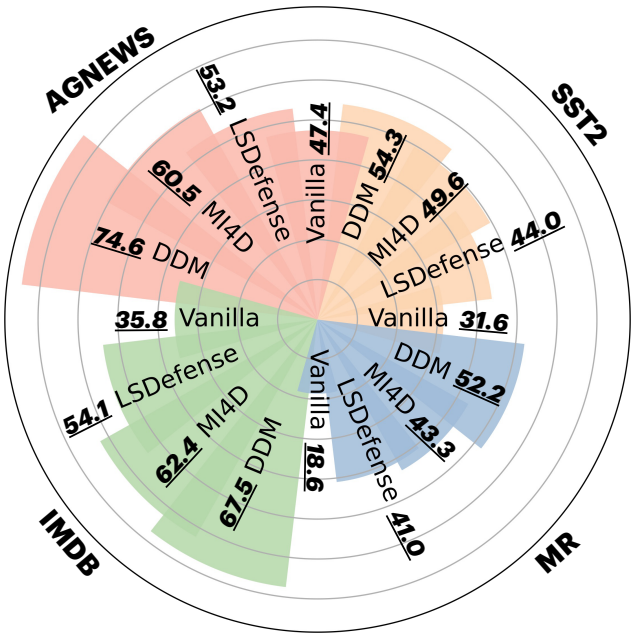


Figure 8
CAA% under SemAttack on four classification benchmarks.

by latent-space perturbations, as exploited by SemAttack. Consequently, these comparison results highlight that the proposed *DDM* remains robust and competitive even under semantically preserving adversarial attacks, particularly SemAttack.

5.2 Generalization to Paired-Input Benchmarks

The primary datasets used in the previous study are single-input classification tasks, such as SST-2, AG NEWS, and so forth. In contrast, paired-input classification tasks require reasoning about the relationship between *two sentences*. For instance, the MNLI dataset (Williams, Nangia, and Bowman 2018), a representative benchmark for Natural Language Inference (NLI), involves predicting the logical relationship, i.e., entailment, contradiction, or neutrality, between two sentences (e.g., a premise and a hypothesis) in a paired-input setting. This added complexity introduces greater challenges for defense methods, as subtle perturbations to either sentence can significantly affect the inferred relationship.

To evaluate the proposed method on paired-input tasks, the same adversarial attack strategies, i.e., TextFooler, BERT-Attack, DeepWordBug, and TextBugger, are applied to the MNLI dataset, and the BERT-BASE encoder is adopted as the Vanilla baseline. Unlike single-input tasks, each input in MNLI consists of a premise–hypothesis pair. Accordingly, all four attacks (with the same configuration described in Section 4.1) are applied to both sentences. In addition, the following defense methods, including InfoBERT, RSMI, and MI4D, are included for comparison. Their results are either re-implemented under the same attack configuration or directly reported from their original papers if available. Experimental settings for *DDM*, including model fine-tuning hyperparameters, are kept consistent with those used in the main experiments. In particular, the frequency-based detection is employed with a masking budget of 20%.

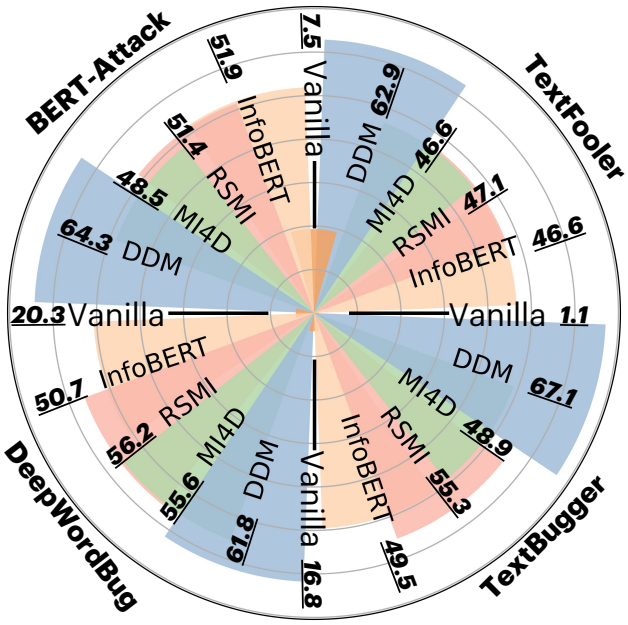


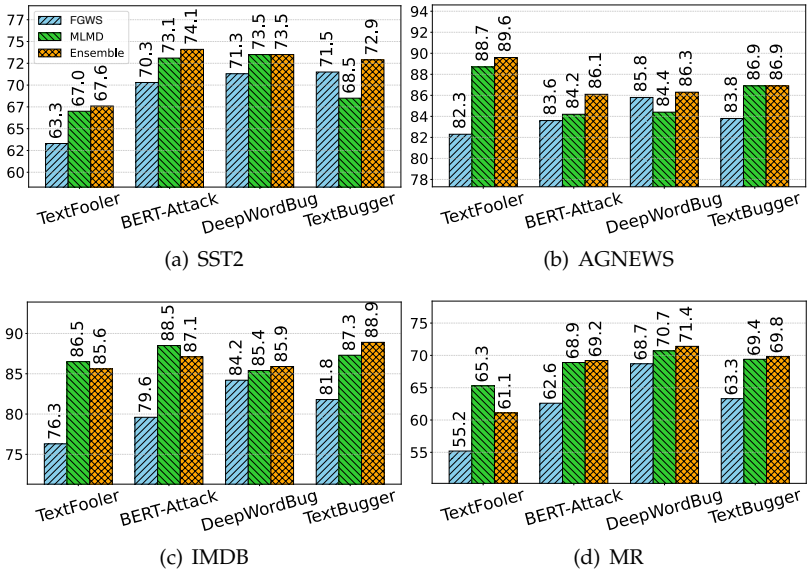
Figure 9
CAA% on MNLI under four different attacks.

The experimental results are presented in Figure 9, where *DDM* demonstrates competitive defense performance against all evaluated attack methods on the MNLI dataset. Specifically, the highest CAA is achieved across all four attacks, yielding the best overall average CAA of 64.0%. These results indicate that *DDM* provides robust adversarial resilience in the paired-input setting, where adversarially suspicious tokens are effectively masked while semantically informative ones are preserved, thereby preserving the logical consistency essential for sentence-pair inference.

5.3 Dependency on Detection Accuracy

Our method utilizes a two-step inference process: first, identifying anomalous tokens; second, substituting them with `[MASK]`. Detection accuracy, accordingly, plays a critical role in defense robustness, as shown in our ablation study (Table 5). A direct correlation exists between higher detection accuracy and improved adversarial resistance, and vice versa. To further strengthen adversarial robustness and reduce the dependency on detection performance, we introduce two improvements. First, one could use the ensemble detection to aggregate predictions from multiple detection models, leveraging diversity to enhance detection accuracy. Second, one could utilize the probabilistic token replacement to dynamically adjust the masking strategy based on a probabilistic replacement. These enhancements aim to reduce the reliance on single-model detection and establish a more resilient mechanism for countering adversarial attacks.

To begin with, we explore the effectiveness of the ensemble detection method by combining WDR, MLMD, and FGWS for suspicious token identification. The ensemble adopts a soft voting strategy based on normalized score averaging to determine which tokens to mask. Specifically, each base detector produces a token-level suspiciousness



(a) SST2 (b) AGNEWS

(c) IMDB (d) MR

Figure 10
A comparison of *DDM* implemented using single (blue using FGWS, green using MLMD) and ensemble (red) detection.

score, which is first normalized and then averaged across detectors. To maintain consistency with prior experiments, the masking ratio is fixed at $b_M = 20\%$.

Figure 10 shows that the ensemble approach consistently outperforms FGWS-based methods across all cases, surpasses MLMD-based detection in 11 out of 16 cases, and achieves comparable CAA scores in two additional cases. This improvement can be attributed to the ensemble ability to mitigate the biases of individual detectors, thereby enhancing overall adversarial token detection across diverse perturbations.

In addition, Table 6 presents a detailed evaluation of adversarial-token detection effectiveness achieved by the ensemble-based approach. The results reveal consistent trends aligned with previous findings: Detection methods with high Recall, low FNR/FPR, and high WF1 often lead to robust classification performance. Specifically, the ensemble method significantly improves Recall and reduces FNR compared to

Table 6
Comparison of adversarial-token detection performance using the single and ensemble-based methods.

Datasets	Methods	TextFooler			BERT-Attack			DeepWordBug			TextBugger		
		Recall%(↑)	FNR%(↓)	WF1%(↑)	Recall%(↑)	FNR%(↓)	WF1%(↑)	Recall%(↑)	FNR%(↓)	WF1%(↑)	Recall%(↑)	FNR%(↓)	WF1%(↑)
SST2	FGWS	69.1	30.9	87.2	76.7	23.2	87.6	77.8	22.2	90.6	78.1	21.9	89.0
	MLMD	73.1	26.9	88.7	79.8	20.2	88.2	79.3	20.7	91.2	74.8	25.2	88.1
	Ensemble	74.0	26.0	88.9	80.1	19.9	88.2	79.3	20.7	91.2	78.9	21.1	89.5
AGNEWS	FGWS	91.8	8.2	94.8	92.4	7.6	93.4	95.3	4.7	89.7	91.3	8.7	89.8
	MLMD	98.5	1.5	96.8	93.6	6.4	93.8	94.0	6.0	89.5	95.0	5.0	90.4
	Ensemble	98.6	1.4	97.9	94.5	5.5	94.0	95.5	4.5	90.9	95.0	5.0	90.4
IMDB	FGWS	88.8	11.2	88.4	87.9	12.1	88.0	89.1	10.9	88.1	91.7	8.3	88.7
	MLMD	94.7	5.3	89.0	96.8	3.2	88.6	93.6	6.4	88.5	95.6	4.4	89.1
	Ensemble	93.9	6.1	88.9	95.8	4.2	88.4	93.8	6.2	88.6	95.8	4.2	89.6
MR	FGWS	63.1	36.9	84.9	71.5	28.5	86.7	78.5	21.5	87.8	74.1	25.9	87.0
	MLMD	76.1	23.9	89.2	80.7	19.3	88.7	82.4	17.6	88.5	81.5	18.5	88.5
	Ensemble	70.2	29.8	87.7	81.0	18.9	88.9	83.1	17.0	89.6	81.8	18.2	88.6

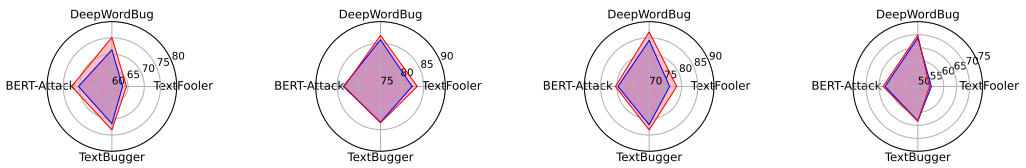
FGWS, while achieving performance comparable to MLMD. This is supported by an average Recall of 87.0% for the ensemble method, versus 82.3% for FGWS and 86.8% for MLMD across all attacks. These gains likely stem from the ensemble's ability to integrate the complementary strengths of individual detectors, thereby enhancing robustness against diverse adversarial perturbations.

Second, we introduce the probabilistic replacement based on a randomized strategy. In pre-training language models such as BERT (Devlin et al. 2019), the MLM is commonly adopted. Specifically, MLM randomly replaces 80% of selected tokens with `[MASK]`, 10% with random words, and leaves 10% unchanged. Inspired by this randomized masking strategy from MLM, we introduce additional rules for token replacement rather than the only process of masking identified tokens in our method. Specifically, for the identified token during inference, our approach adopts the following rules:

- 60% chance: replacing with a `[MASK]` token.
- 30% chance: inserting a `[MASK]` token before.
- 10% chance: leaving unchanged.

That is, the token replacement introduces greater randomness compared with the previous directly-masking strategy, leaving up to 40% of identified tokens unchanged. Among these, 75% include the insertion of an additional `[MASK]` token before the identified tokens, following a similar approach to the insertion strategy in MI4D (Hu et al. 2023). Notably, this approach is not applied during model training but only for inference, as the training input consists of clean data.

Figure 11 illustrates the performance with the probabilistic token replacement approach. Compared with the previous directly-masking strategy, the probabilistic based approach achieves better defense accuracy (CAA, %) in almost all scenarios. As exemplified by the SST2 and IMDB datasets, the previous direct-masking strategy achieves average CAA% scores of 69.1 and 80.5, respectively, while the probabilistic replacement approach demonstrates superior performance with scores of 71.4 and 82.3. The observed improvement can be attributed to an effective balance between preserving contextual integrity and mitigating the influence of adversarial tokens. Under this new probabilistic replacement, replacing 30% of the tokens with `[MASK]` while keeping 10% unchanged helps in retention of the original context, minimizing the risk of information loss. In other words, this strategy reduces the impact of detection errors by maintaining valuable context through minimally altered or unmodified tokens. Moreover, it is also noteworthy that `[MASK]` insertion simplifies *DDM* to MI4D, reflecting its flexibility.



(a) SST2 (b) AGNEWS (c) IMDB (d) MR

Figure 11
A comparison of the defense accuracy (CAA, %) with the probabilistic token replacement approach (red), compared with the previous directly masking strategy (blue).

5.4 Integration with LLMs

The following section examines the adversarial robustness of LLMs enhanced by the proposed *DDM*. The aim is to determine whether the proposed method effectively improves LLM performance when exposed to adversarial inputs. To integrate *DDM* into LLMs, the following adaptation steps are used: (1) few-shot prompt learning is utilized to guide the LLM. Specifically, 1,000 training samples are randomly selected as exemplars, with b_M% `[MASK]` tokens inserted at the beginning of these clean samples; (2) 100 testing samples are randomly drawn from the testing dataset and subjected to four types of adversarial attacks; (3) the LLM is prompted to identify adversarially modified tokens within the perturbed inputs and replaces the most suspicious ones with `[MASK]`, using a fixed masking budget b_M (see the example prompt from "Prompt for detection"); (4) the LLM predicts the labels of the masked samples using in-context learning, leveraging the exemplars generated in the first step (see the example prompt from "Prompt for classification").

Prompt for detection

You are an expert in adversarial detection. The text below may have been perturbed by attack methods. Identify the top 20% of tokens most likely to be disturbed and replace them with `[MASK]`. An example output is provided below, where `[MASK]` replaces suspicious tokens.
Input: Ky. Company Wins Grant to Study Peptides (ABP) AP - A company founded by a chemistry researcher at the University of Louisville won a grant to develop a method of producing better peptides, which are short chains of amino acids, the building blocks of proteins.
Output: Ky. Company Wins Grant to `[MASK]` Peptides (`[MASK]`) AP - A company `[MASK]` by a chemistry `[MASK]` at the `[MASK]` of `[MASK]` won a grant to develop a method of `[MASK]` better peptides, which are short chains of amino acids, the `[MASK]` blocks of `[MASK]`.
Based on the example above, detect the following input.
Input: · · ·

Prompt for classification

You are an expert in text classification. Below is the input context and its related label.
Input: `[MASK]` `[MASK]` `[MASK]` `[MASK]` `[MASK]` Fears for T N pension after talks. Unions representing workers at Turner Newall say they are 'disappointed' after talks with stricken parent firm Federal Mogul. **Label**: *business*.
· · ·
Based on the examples above, classify the following input test and output its label within the list of '*World*, *Sports*, *business*, *Science/Technology*':
Input: Ky. Company Wins Grant to `[MASK]` Peptides (`[MASK]`) AP - A company `[MASK]` by a chemistry `[MASK]` at the `[MASK]` of `[MASK]` won a grant to develop a method of `[MASK]` better peptides, which are short chains of amino acids, the `[MASK]` blocks of `[MASK]`.

Table 7
Performance evaluation of *DDM* integrated with LLM for adversarial defense (in terms of CAA %), where the best results are marked in **bold**.

Datasets	Methods	TextFooler	BERT-Attack	DeepWordbug	TextBugger
SST2	Llama3-8B	75.1	80.9	86.1	91.7
	+DDM	**82.3**	**87.5**	**90.1**	**93.2**
	GPT3.5	78.4	82.1	89.1	94.2
	+DDM	**84.4**	**88.1**	**91.2**	**94.7**
AGNEWS	Llama3-8B	69.4	68.9	76.1	71.5
	+DDM	**86.7**	**85.8**	**88.2**	**89.9**
	GPT3.5	70.9	70.3	74.7	71.2
	+DDM	**89.9**	**87.8**	**90.3**	**91.5**
IMDB	Llama3-8B	85.5	84.3	88.7	89.3
	+DDM	**89.7**	**90.1**	**93.4**	**95.3**
	GPT3.5	91.4	90.5	94.4	95.7
	+DDM	**92.9**	**94.3**	**96.1**	**96.9**
MR	Llama3-8B	71.7	83.9	79.9	84.2
	+DDM	**81.1**	**90.2**	**87.4**	**92.7**
	GPT3.5	73.8	83.7	81.4	86.3
	+DDM	**82.5**	**89.4**	**88.9**	**93.2**

Table 7 presents the performance gains achieved by integrating *DDM* with LLMs, specifically, Llama3-8B and GPT-3.5. Several key observations can be made: (1) The vanilla LLMs already outperform smaller-scale models (see Table 3). Highlighting the inherent advantages of LLMs in understanding the input context. (2) *DDM* enhances GPT-3.5 across four adversarial attack methods, yielding an average improvement of 6.0 on SST2, 6.0 on AGNEWS, 2.1 on IMDB, and 0.5 on MR. A similar improvement is observed when integrating *DDM* with Llama3-8B. (3) The combination of *DDM* with GPT-3.5 achieves a higher absolute performance than its integration with Llama3-8B. However, the relative improvement rate seems higher for Llama3-8B (with the SST2, IMDB, and MR benchmarks), suggesting that Llama3-8B benefits more from *DDM* due to its relatively weaker robustness in the vanilla setup, leaving more room for improvement. Overall, *DDM* demonstrates strong effectiveness in enhancing LLM robustness by filtering adversarial inputs and preserving critical information. This capability highlights its utility in improving LLM resilience under adversarial conditions.

6. Conclusion

In this article, we propose a novel adversarial defense method leveraging the strategic insertion and replacement of `[MASK]` tokens during both training and inference stages. Specifically, during training, `[MASK]` tokens are injected into samples as placeholders to prepare the model for unseen inputs. During inference, suspicious tokens are replaced by `[MASK]` tokens for enhanced classification robustness.

The proposed method unifies adversarial detection and defense, offering a systematic approach to utilize detection outcomes for effectively mitigating the influence of adversarial attack. More importantly, a comprehensive theoretical analysis is provided to validate the proposed method's effectiveness. We need to point out that in our analysis, we assumed that the reconstructed `[CLS]` token is uniformly distributed in the convex hulls. For the scenarios that this assumption may not hold, especially for complex adversarial attacks, more detailed analysis is desirable, for example, using

multi-modal distributions for the latent feature space and a full-scale geometric analysis on transformer block behavior to adapt to various scenarios. We leave it to our future work. Experimental results across diverse datasets and attack models demonstrate that the method consistently outperforms existing defense approaches, significantly enhancing the model robustness. Furthermore, applying our method to LLMs also shows substantial improvements in robustness, demonstrating its adaptability to various language models.

These findings highlight the potential of our approach to advance NLP security. Moreover, the proposed method is agnostic to downstream tasks, enabling its seamless integration into diverse applications.

Appendix A. Statistical and Computational Evaluation

Statistical Analysis. To assess the statistical significance of our experimental results, we utilize the one-sample *t*-test, which evaluates whether the observed performance improvement of our method over the strongest baseline is statistically significant. As a result, statistically significant improvements at $p < 10^{-3}$ are marked with † in Table 3. In addition, Table A.1 further reports the standard deviations from classification accuracy (CAA%) and success rate (SUCC%) across four attack methods for each dataset.

Table A.1
Performance of *DDM* (in terms of average CAA%, SUCC%, and relevant standard deviations) against four adversarial attacks across each dataset.

Datasets	TextFooler		BERT-Attack		DeepWordBug		TextBugger	
	CAA%	SUCC%	CAA%	SUCC%	CAA%	SUCC%	CAA%	SUCC%
SST2	63.3 ± 0.9	26.5 ± 1.3	70.3 ± 1.2	25.2 ± 1.5	71.3 ± 1.3	21.8 ± 1.0	71.5 ± 0.9	22.8 ± 1.4
AGNEWS	82.3 ± 1.7	10.6 ± 0.7	83.6 ± 1.0	9.8 ± 0.7	85.8 ± 1.9	8.7 ± 1.2	83.3 ± 1.6	10.1 ± 1.1
IMDB	76.3 ± 1.3	17.4 ± 1.0	79.6 ± 1.7	13.4 ± 0.8	84.2 ± 1.2	8.5 ± 0.5	81.8 ± 1.3	11.3 ± 0.9
MR	55.2 ± 0.9	37.9 ± 1.8	62.6 ± 1.0	29.9 ± 0.9	68.7 ± 2.4	19.6 ± 1.2	63.3 ± 1.3	26.5 ± 1.6

Overall, the consistently-low standard deviations across all datasets and attack types indicate that *DDM* maintains stable performance under diverse perturbations and random seeds. These results highlight the robustness and reliability of the proposed method, demonstrating its effectiveness across a broad range of adversarial scenarios.

Computational Evaluation. To provide a comprehensive view of the computational cost introduced by our method, we report the training time per epoch, average inference latency across all test samples, and FLOating-Point operations (FLOPs). FLOPs are computed based on standard multiply-accumulate operation counts for both the forward and backward passes. Specifically, the Vanilla model with a BERT-base encoder is employed as the baseline. The proposed *DDM* is implemented with a masking budget of $b_M = 20\%$. That is, during training, we inject `[MASK]` tokens equivalent to 20% of the input sequence length to each clean sample for model fine-tuning. At inference time, the top 20% of tokens with the highest suspicious scores are identified and replaced with `[MASK]` tokens.

As shown in Table A.2, the proposed masking-based training introduces moderate computational cost. On AGNEWS, for instance, *DDM* takes 612.57 seconds and 5.61 GFLOPs per epoch, while the Vanilla model requires 552.39 seconds and 4.75 GFLOPs, resulting in a relative increase of approximately 10.9% in training time and 18.1% in

Table A.2
Comparison of computational efficiency between the Vanilla model and the proposed *DDM* with the BERT-base encoder, measured in terms of training time, inference time, and FLOPs. In the *DDM* (Infer) column, the number in parentheses represents the detection time, and the number outside denotes the classification time.

Dataset	Vanilla (Train)		*DDM* (Train)		Vanilla (Infer)		*DDM* (Infer)	
	Time (s)	FLOPs (G)	Time (s)	FLOPs (G)	Time (s)	FLOPs (G)	Time (s)	FLOPs (G)
SST2	164.58	1.11	181.99	1.30	0.89	3.43	0.87 (+0.83)	2.57
AGNEWS	552.39	4.75	612.57	5.61	2.82	7.01	2.76 (+1.12)	4.21
IMDB	560.57	24.61	641.71	29.03	5.81	38.65	5.86 (+4.95)	32.27
MR	26.01	2.31	29.55	2.75	1.11	1.62	1.06 (+0.56)	1.28

FLOPs. Similar trends are observed across other datasets, primarily due to the increased input length after inserting additional `[MASK]` tokens during training.

Despite this overhead, *DDM* introduces minimal additional cost during inference. Specifically, the classification time (excluding detection) remains almost identical to that of the vanilla model, averaging 2.63s compared to 2.65s across all datasets. The detection stage itself is lightweight: for example, on AGNEWS, the FGWS takes only 1.12s, and on SST2, just 0.83s. These results confirm that the proposed method can efficiently identify suspicious tokens without significantly slowing down prediction.

Moreover, *DDM* consistently reduces FLOPs during classification, as evidenced by the drop from 3.43G to 2.57G on SST2 and from 7.01G to 4.21G on AGNEWS. This reduction is attributed to the masking of low-frequency or suspicious tokens during inference, which bypasses their contextual encoding and decreases self-attention computations in the BERT layers.

Acknowledgments

The authors would like to thank anonymous reviewers for their valuable suggestions to improve the quality of the article. This work is partially supported by the Australian Research Council Discovery Project (DP210101426), AEGiS Advance grant (888/008/268, University of Wollongong), and Telstra-UOW Hub for AIOT Solutions Seed Funding (2024, 2025).

References

Ankner, Zachary, Naomi Saphra, Davis Blalock, Jonathan Frankle, and Matthew Leavitt. 2024. Dynamic masking rate schedules for MLM pretraining. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 477–487. https://doi.org/10.18653/v1/2024.eacl-short.42

Bao, Rongzhou, Jiayi Wang, and Hai Zhao. 2021. Defending pre-trained language models from adversarial word substitution without performance sacrifice. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3248–3258. https://doi.org/10.18653/v1/2021.findings-acl.287

Bao, Rong, Rui Zheng, Liang Ding, Qi Zhang, and Dacheng Tao. 2023. CASN: Class-aware score network for textual adversarial detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 671–687. https://doi.org/10.18653/v1/2023.acl-long.40

Bhusal, Dipkamal, Md Tanvirul Alam, Monish K. Veerabhadran, Michael Clifford, Sara Rampazzi, and Nidhi Rastogi. 2024. PASA: Attack agnostic unsupervised adversarial detection using prediction & attribution sensitivity analysis. In *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)*, pages 21–40. https://doi.org/10.1109/EuroSP60621.2024.00010

Carrara, Fabio, Rudy Becarelli, Roberto Caldelli, Fabrizio Falchi, and Giuseppe Amato. 2018. Adversarial examples detection in features distance spaces. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.

https://doi.org/10.1007/978-3-030-11012-3_26

Czinczoll, Tamara, Christoph Hönes, Maximilian Schall, and Gerard De Melo. 2024. NextLevelBERT: Masked language modeling with higher-level representations for long documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4656–4666. https://doi.org/10.18653/v1/2024.acl-long.256

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186. https://doi.org/10.18653/v1/N19-1423

Ebrahimi, Javid, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36. https://doi.org/10.18653/v1/P18-2006

Freitas, Scott, Shang-Tse Chen, Zijie J Wang, and Duen Horng Chau. 2020. UnMask: Adversarial detection and defense through robust feature alignment. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1081–1088. https://doi.org/10.1109/BigData50022.2020.9378303

Gao, Ji, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. https://doi.org/10.1109/SPW.2018.00016

Gao, SongYang, Shihan Dou, Qi Zhang, Xuanjing Huang, Jin Ma, and Ying Shan. 2023. On the universal adversarial perturbations for efficient data-free adversarial detection. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13573–13581. https://doi.org/10.18653/v1/2023.findings-acl.857

Gietz, Harrison and Jugal Kalita. 2024. MaskPure: Improving defense against text adversaries with stochastic purification. In *Natural Language Processing and Information Systems: 29th International Conference on Applications of Natural Language to Information Systems, NLDB 2024, Proceedings, Part I*, pages 379–393. https://doi.org/10.1007/978-3-031-70239-6_26

Gupta, Ashim, Carter Blum, Temma Choji, Yingjie Fei, Shalin Shah, Alakananda Vempala, and Vivek Srikumar. 2023. Don't retrain, just rewrite: Countering adversarial perturbations by rewriting text. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13981–13998. https://doi.org/10.18653/v1/2023.acl-long.781

Hu, Xinrong, Ce Xu, Junlong Ma, Zijian Huang, Jie Yang, Yi Guo, and Johan Barthelemy. 2023. [MASK] insertion: A robust method for anti-adversarial attacks. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1058–1070. https://doi.org/10.18653/v1/2023.findings-eacl.78

Jin, Di, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(5):8018–8025. https://doi.org/10.1609/aaai.v34i05.6311

Le, Thai, Noseong Park, and Dongwon Lee. 2022. SHIELD: Defending textual neural networks against multiple black-box adversarial attacks with stochastic multi-expert patcher. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6661–6674. https://doi.org/10.18653/v1/2022.acl-long.459

Li, Jinfeng, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TextBugger: Generating adversarial text against real-world applications. In *Network and Distributed Systems Security (NDSS) Symposium*. https://doi.org/10.14722/ndss.2019.23138

Li, Linyang, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202. https://doi.org/10.18653/v1/2020.emnlp-main.500

Li, Linyang, Demin Song, and Xipeng Qiu. 2023. Text adversarial purification as defense against adversarial attacks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 338–350.

https://doi.org/10.18653/v1/2023.acl-long.20

Li, Qiwei, Zuchao Li, Ping Wang, Haojun Ai, and Hai Zhao. 2024a. Hypergraph based understanding for document semantic entity recognition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2950–2960. https://doi.org/10.18653/v1/2024.acl-long.162

Li, Wei and Houfeng Wang. 2024. Detection-correction structure via general language model for grammatical error correction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1748–1763. https://doi.org/10.18653/v1/2024.acl-long.96

Li, Yi, Plamen Angelov, and Neeraj Suri. 2025. Self-supervised representation learning for adversarial attack detection. In *European Conference on Computer Vision*, pages 236–252. https://doi.org/10.1007/978-3-031-73027-6_14

Li, Zhenhao, Marek Rei, and Lucia Specia. 2024. DiffuseDef: Improved robustness to adversarial attacks. *arXiv preprint arXiv:2407.00248*. https://doi.org/10.18653/v1/2025.acl-long.454

Li, Zetong, Qinliang Su, Shijing Si, and Jianxing Yu. 2024b. Leveraging BERT and TFIDF features for short text clustering via alignment-promoting co-training. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14897–14913. https://doi.org/10.18653/v1/2024.emnlp-main.828

Li, Zongyi, Jianhan Xu, Jiehang Zeng, Linyang Li, Xiaoqing Zheng, Qi Zhang, Kai-Wei Chang, and Cho-Jui Hsieh. 2021. Searching for an effective defender: Benchmarking defense against adversarial word substitution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3137–3147. https://doi.org/10.18653/v1/2021.emnlp-main.251

Liu, Han, Zhi Xu, Xiaotong Zhang, Xiaoming Xu, Feng Zhang, Fenglong Ma, Hongyang Chen, Hong Yu, and Xianchao Zhang. 2023. SSPAttack: A simple and sweet paradigm for black-box hard-label textual adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13228–13235. https://doi.org/10.1609/aaai.v37i11.26553

Liu, Qin, Rui Zheng, Bao Rong, Jingyi Liu, ZhiHua Liu, Zhanzhan Cheng, Liang Qiao, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022. Flooding-X: Improving BERT's resistance to adversarial attacks via loss-restricted fine-tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5634–5644. https://doi.org/10.18653/v1/2022.acl-long.386

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

Maas, Andrew L., Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150.

Meng, Zhao, Yihan Dong, Mrinmaya Sachan, and Roger Wattenhofer. 2022. Self-supervised contrastive learning with adversarial perturbations for defending word substitution-based attacks. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 87–101. https://doi.org/10.18653/v1/2022.findings-naacl.8

Moon, Han Cheol, Shafiq Joty, Ruochen Zhao, Megh Thakkar, and Chi Xu. 2023. Randomized smoothing with masked inference for adversarially robust text classifications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5145–5165. https://doi.org/10.18653/v1/2023.acl-long.282

Moraffah, Raha, Shubh Khandelwal, Amrita Bhattacharjee, and Huan Liu. 2024. Adversarial text purification: A large language model approach for defense. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 65–77. https://doi.org/10.1007/978-981-97-2262-4_6

Morris, John, Eli Lifland, Jack Lanchantin, Yangfeng Ji, and Yanjun Qi. 2020a. Reevaluating adversarial examples in natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3829–3839. https://doi.org/10.18653/v1/2020.findings-emnlp.341

Morris, John, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020b.

TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126. https://doi.org/10.18653/v1/2020.emnlp-demos.16

Mosca, Edoardo, Shreyash Agarwal, Javier Rando Ramírez, and Georg Groh. 2022. "That is a suspicious reaction!": Interpreting logits variation to detect NLP adversarial attacks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7806–7816. https://doi.org/10.18653/v1/2022.acl-long.538

Mozes, Maximilian, Pontus Stenetorp, Bennett Kleinberg, and Lewis Griffin. 2021. Frequency-guided word substitutions for detecting textual adversarial examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 171–186. https://doi.org/10.18653/v1/2021.eacl-main.13

Omar, Marwan and Gita Sukthankar. 2023. Text-Defend: Detecting adversarial examples using Local Outlier Factor. In *2023 IEEE 17th International Conference on Semantic Computing (ICSC)*, pages 118–122. https://doi.org/10.1109/ICSC56153.2023.00026

Pang, Bo and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124. https://doi.org/10.3115/1219840.1219855

Rafiei Asl, Javad, Prajwal Panzade, Eduardo Blanco, Daniel Takabi, and Zhipeng Cai. 2024. RobustSentEmbed: Robust sentence embeddings using adversarial self-supervised contrastive learning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3795–3809. https://doi.org/10.18653/v1/2024.findings-naacl.241

Raman, Mrigank, Pratyush Maini, J. Kolter, Zachary Lipton, and Danish Pruthi. 2023. Model-tuning via prompts makes NLP models adversarially robust. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9266–9286. https://doi.org/10.18653/v1/2023.emnlp-main.576

Socher, Richard, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642. https://doi.org/10.18653/v1/D13-1170

Wang, Boxin, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021. InfoBERT: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*.

Wang, Boxin, Chejian Xu, Xiangyu Liu, Yu Cheng, and Bo Li. 2022. SemAttack: Natural textual attacks via different semantic spaces. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 176–205. https://doi.org/10.18653/v1/2022.findings-naacl.14

Wang, Xiaosen, Yichen Yang, Yihe Deng, and Kun He. 2020. Adversarial training with fast gradient projection method against synonym substitution based text attacks. *arXiv preprint arXiv:2008.03709*. https://doi.org/10.1609/aaai.v35i16.17648

Wang, Zhaoyang, Zhiyue Liu, Xiaopeng Zheng, Qinliang Su, and Jiahai Wang. 2023. RMLM: A flexible defense framework for proactively mitigating word-level adversarial attacks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2757–2774. https://doi.org/10.18653/v1/2023.acl-long.155

Williams, Adina, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. https://doi.org/10.18653/v1/N18-1101

Xu, Jianhan, Cenyuan Zhang, Xiaoqing Zheng, Linyang Li, Cho-Jui Hsieh, Kai-Wei Chang, and Xuanjing Huang. 2022. Towards adversarially robust text classifiers by learning to reweight clean examples. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1694–1707. https://doi.org/10.18653/v1/2022.findings-acl.134

Yang, Wangli, Jie Yang, Wanqing Li, and Yi Guo. 2024. ConClue: Conditional clue extraction formultiple choice question answering. In *Document Analysis and Recognition - ICDAR 2024*, pages 183–198. https://doi.org/10.1007/978-3-031-70552-6_11

Yang, Yahan, Soham Dan, Dan Roth, and Insup Lee. 2023. In and out-of-domain text adversarial robustness via label smoothing. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 657–669. https://doi.org/10.18653/v1/2023.acl-short.58

Yang, Yichen, Xin Liu, and Kun He. 2024. Fast adversarial training against textual adversarial attacks. *arXiv preprint arXiv:2401.12461*. https://doi.org/10.18653/v1/2025.findings-naacl.43

Ye, Mao, Chengyue Gong, and Qiang Liu. 2020. SAFER: A structure-free approach for certified robustness to adversarial word substitutions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3465–3475. https://doi.org/10.18653/v1/2020.acl-main.317

Yoo, Jin Yong and Yanjun Qi. 2021. Towards improving adversarial training of NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 945–956. https://doi.org/10.18653/v1/2021.findings-emnlp.81

Yuan, Shilong, Wei Yuan, and Tieke He. 2024. ROIC-DM: Robust text inference and classification via diffusion model. *arXiv preprint arXiv:2401.03514*.

Zeng, Jiehang, Jianhan Xu, Xiaoqing Zheng, and Xuanjing Huang. 2023. Certified robustness to text adversarial attacks by randomized [MASK]. *Computational Linguistics*, 49(2):395–427. https://doi.org/10.1162/coli_a_00476

Zhan, Pengwei, Jing Yang, He Wang, Chao Zheng, Xiao Huang, and Liming Wang. 2023. Similarizing the influence of words with contrastive learning to defend word-level adversarial text attack. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7891–7906. https://doi.org/10.18653/v1/2023.findings-acl.500

Zhang, Cenyuan, Xiang Zhou, Yixin Wan, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. 2022. Improving the adversarial robustness of NLP models by information bottleneck. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3588–3598. https://doi.org/10.18653/v1/2022.findings-acl.284

Zhang, Xiang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 649–657.

Zhang, Xiaomei, Zhaoxi Zhang, Qi Zhong, Xufei Zheng, Yanjun Zhang, Shengshan Hu, and Leo Yu Zhang. 2023. Masked language model based textual adversarial example detection. In *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, pages 925–937. https://doi.org/10.1145/3579856.3590339

Zhang, Zeliang, Wei Yao, Susan Liang, and Chenliang Xu. 2024. Random smooth-based certified defense against text adversarial attack. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1251–1265.

Zhao, Jiahao and Wenji Mao. 2023. Generative adversarial training with perturbed token detection for model robustness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13012–13025. https://doi.org/10.18653/v1/2023.emnlp-main.804

Zheng, Rui, Bao Rong, Yuhao Zhou, Di Liang, Sirui Wang, Wei Wu, Tao Gui, Qi Zhang, and Xuanjing Huang. 2022. Robust lottery tickets for pre-trained language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2211–2224. https://doi.org/10.18653/v1/2022.acl-long.157

Zhu, Chen, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. FreeLB: Enhanced adversarial training for Natural Language Understanding. In *International Conference on Learning Representations*, pages 26–30.