Not All Tokens Are Equal: Per-Dimension Top-K Pooling for Adversarially Robust BERT Classification

Manoranjan Dash
School of Computing & Data Sciences
FLAME University
Pune, India
manoranjan.dash@flame.edu.in

Shivam Anand Aralikatti
School of Computing & Data Sciences
FLAME University
Pune, India
shivam.aralikatti@flame.edu.in

Shanay Sheth
School of Computing & Data Sciences
FLAME University
Pune, India
shanay.sheth@flame.edu.in

Pranav Shinde
School of Computing & Data Sciences
FLAME University
Pune, India
pranav.shinde@flame.edu.in

Abstract

Contextual text classification with BERT typically relies on the [CLS] token representation for downstream prediction. While effective under standard conditions, [CLS]-based pooling is brittle under adversarial perturbation, as its single-vector representation is indiscriminately influenced by injected adversarial tokens. We propose Per-Dimension Top-K Average Pooling, a pooling strategy that, for each hidden dimension, selectively aggregates only the top-K token activations rather than the full sequence — effectively controlling which tokens contribute to the final representation. This token-level selectivity acts as a natural filter against adversarial injection: tokens that do not rank among the top-K for a given dimension are suppressed from aggregation. We evaluate our approach against CLS, Global Average Pooling (GAP), Global Max Pooling (GMP), and Hybrid variants across three text classification domains: spam detection (Enron and LingSpam), automated essay scoring (ASAP), and hate speech classification. On the Enron spam dataset under adversarial attack, our best Hybrid (K=3) variant reduces the Attack Success Rate from 70.65% to 37.07% while maintaining clean accuracy above 99%, compared to CLS which degrades to 63.64% adversarial accuracy. Representation-level analyses further corroborate these findings: Top-K pooling variants exhibit substantially lower cosine similarity shift under attack, and adversarially injected tokens enter the top-K selection in far fewer dimensions compared to CLS. Adversarial robustness gains are most pronounced on the Enron spam dataset; improvements on LingSpam are more modest, and the hate speech and AES experiments do not include adversarial evaluation. These results suggest that per-dimension token selectivity offers a principled and lightweight mechanism for adversarial robustness in BERT-based spam classifiers without any modification to the underlying model architecture.

1 Introduction

Text classification is one of the most foundational tasks in natural language processing (NLP), underpinning a broad range of real-world applications including spam detection (Androutsopoulos et al., 2000), hate speech moderation (Zampieri et al., 2019), sentiment analysis, and automated essay scoring (AES) (Taghipour and Ng, 2016). Early approaches to text classification relied on handcrafted features and shallow machine learning models (Sahami et al., 1998). The emergence of deep neural networks, and more recently large pre-trained language models, has substantially advanced the state of the art across all these tasks.

Among pre-trained language models, BERT (Devlin et al., 2019) and its variants (Liu et al., 2019; Yang et al., 2019) have come to dominate the NLP landscape. Built on the Transformer architecture (Vaswani et al., 2017), BERT is pre-trained on massive corpora using masked language modelling and next-sentence prediction objectives, producing rich contextualised token representations. In the standard fine-tuning paradigm, the representation of the special `[CLS]` token — prepended to every input sequence and trained to aggregate sentence-level information — is passed to a downstream classification head. This approach is simple, parameter-efficient, and has become the de facto standard for text classification with BERT.

Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM), pages 285–295

Despite its widespread adoption, [CLS]-based pooling carries a fundamental vulnerability. The [CLS] token aggregates information from the entire token sequence via self-attention (Vaswani et al., 2017), meaning every token shapes the final representation without any selectivity criterion. Under adversarial conditions — where tokens are crafted or injected to shift model predictions (Goodfellow et al., 2015; Jia and Liang, 2017) — this indiscriminate aggregation is directly exploitable. Jin et al. (2020) showed that word substitutions (TextFooler) dramatically reduce BERT accuracy with only a handful of replacements; Ebrahimi et al. (2018) demonstrated gradient-guided character-level flips; Li et al. (2020b) introduced BERT-ATTACK using BERT to generate coherent adversarial substitutions; and Wallace et al. (2019) showed that universal adversarial triggers cause consistent misprediction at scale.

Alternative pooling strategies have been explored as replacements for [CLS] aggregation. Global Average Pooling (GAP) averages the hidden states of all tokens in the sequence, while Global Max Pooling (GMP) takes the element-wise maximum. Sentence-BERT (Reimers and Gurevych, 2019) demonstrated that mean-pooled BERT representations substantially outperform raw [CLS] representations on semantic similarity tasks, establishing that the choice of pooling strategy meaningfully affects representation quality. However, GAP considers all tokens equally, inheriting the same vulnerability to adversarial injection as [CLS]. GMP selects the single maximum-activating token per dimension, which introduces a degree of selectivity but is simultaneously susceptible to high-magnitude adversarial insertions, since a single injected token with an extreme activation can dominate the pooled representation. Neither strategy provides a principled mechanism for balancing broad sequence coverage against robustness to token-level perturbation.

The idea of selecting top-K activations for pooling has precedent in deep learning. Kalchbrenner et al. (2014) introduced k-max pooling for CNN-based sentence modelling, and similar principles appear in graph pooling methods such as gPool (Gao and Ji, 2019) and SAGPool (Lee et al., 2019). Kim (2014) employed max-over-time pooling — the k=1 special case — in text CNNs. However, the principle has not been systematically applied to Transformer-based models, and its adversarial robustness properties in BERT-based text classification have not, to our knowledge, been previously explored.

In this work, we propose **Per-Dimension Top-K Average Pooling**, a simple yet effective pooling strategy that, for each hidden dimension, aggregates only the top-K token activations rather than the full sequence. This design introduces explicit *token-level selectivity*: tokens that do not rank among the top-K for a given dimension are excluded from aggregation entirely. We argue that this selectivity acts as a natural, architecture-agnostic filter against adversarial token injection: adversarially constructed tokens are unlikely to consistently rank among the top-K across all dimensions simultaneously, and their influence on the final pooled representation is therefore structurally limited. Crucially, this mechanism requires no modification to the underlying BERT model, no adversarial training, and no task-specific augmentation — it operates purely at the level of the pooling function applied to frozen or fine-tuned BERT token representations.

We evaluate Per-Dimension Top-K Average Pooling and a set of Hybrid variants — which concatenate the [CLS] token representation with Top-K pooling (or with GMP/GAP) — across three text classification domains: spam detection on the Enron and LingSpam datasets, automated essay scoring on the ASAP benchmark, and hate speech multi-classification(ElSherief et al., 2021). Our primary focus is adversarial robustness in the spam detection setting, where the gains are most pronounced and where adversarial injection is a highly realistic threat model. We note that the adversarial evaluation is restricted to the magic-word injection paradigm on spam data; hate speech and AES experiments evaluate clean-data performance only. To understand *why* Top-K pooling confers robustness, we conduct two representation-level analyses: a cosine similarity shift analysis that measures how much the final representations change under attack, and a dimension-level activation analysis that examines how frequently adversarially injected tokens enter the top-K selection for a given dimension. Together, these analyses provide mechanistic evidence that the observed robustness is a direct and principled consequence of token-selective aggregation, not an artefact of any particular evaluation setting.

Contributions. Our main contributions are as follows:

- We propose **Per-Dimension Top-K Average Pooling**, a lightweight and architecture-agnostic pooling strategy for BERT-based text classifiers that requires no modification to the underlying model, no adversarial training, and no data augmentation.

- We demonstrate that Top-K pooling and its Hybrid variants substantially improve adversarial robustness on the Enron spam detection dataset, reducing the Attack Success Rate from 70.65% (CLS baseline) to 37.07% (Hybrid, $K{=}3$) while maintaining clean accuracy above 99%. Gains on LingSpam are present but more modest, and we analyse the dataset-level factors that explain this divergence (Section 5).

- We introduce two representation-level diagnostic analyses — cosine similarity shift and dimension-level activation analysis — that provide mechanistic insight into the source of adversarial robustness in Top-K pooling, showing that adversarially injected tokens enter the top-K selection in substantially fewer dimensions compared to CLS or GAP.

- We conduct a comprehensive evaluation across three diverse NLP tasks (spam detection, AES, and hate speech classification) and multiple values of K, systematically characterizing when and where Top-K pooling is most beneficial.

2 Related Work

2.1 Pooling Strategies for BERT Representations

The choice of pooling strategy for aggregating BERT token representations has received considerable attention. The original BERT paper (Devlin et al., 2019) proposed using the `[CLS]` token representation for classification, a convention widely adopted in subsequent work. Reimers and Gurevych (2019) demonstrated that mean pooling of all token representations substantially outperforms raw `[CLS]` representations for semantic similarity tasks, motivating the exploration of alternatives. Li et al. (2020a) further investigated pooling strategies for sentence embeddings, finding that the optimal choice is task-dependent. Beyond simple aggregation, attention-weighted pooling (Yang et al., 2016) has been proposed as a way to focus on task-relevant tokens, though this introduces additional learnable parameters and does not inherently confer adversarial robustness. Our work focuses explicitly on adversarial robustness properties of pooling strategies, which prior work has not examined.

2.2 Adversarial Attacks on Text Classification

Adversarial robustness in NLP has been studied extensively. Goodfellow et al. (2015) laid the theoretical groundwork for adversarial examples in continuous input spaces, and subsequent work adapted these ideas to discrete text inputs. Ebrahimi et al. (2018) introduced HotFlip, a white-box attack using gradient-guided character- and word-level substitutions. Jin et al. (2020) proposed TextFooler, which generates semantically similar adversarial examples by replacing words with BERT-predicted substitutes, and showed that even strong BERT-based classifiers are highly vulnerable. Li et al. (2020b) introduced BERT-ATTACK, which uses BERT itself to generate fluent and semantically coherent adversarial substitutions. Alzantot et al. (2018) employed population-based search to generate natural-language adversarial examples, and Wallace et al. (2019) showed that universal adversarial triggers — short token sequences appended to any input — can cause consistent misprediction across a model's entire input distribution. Our work does not propose a new attack method but instead evaluates how pooling strategy choice affects vulnerability to these established attack paradigms. Evaluating Top-K pooling against substitution-based attacks (e.g. TextFooler, BERT-ATTACK) and adaptive adversaries is an important direction for future work.

2.3 Application Domains and Defences

NLP defences against adversarial attacks fall into three categories: adversarial training (Madry et al., 2018), input transformation, and architecture modifications — all of which are computationally expensive or brittle to adaptive attacks. Our approach requires none of these, as robustness emerges directly from token-selective aggregation. The Enron and LingSpam datasets (Androutsopoulos et al., 2000) are standard spam detection benchmarks; AES has evolved from feature-engineered models (**?**) to BERT-based approaches evaluated on ASAP; and hate speech classification (Zampieri et al., 2019) presents additional challenges owing to class imbalance and linguistic subtlety.

3 Methodology

3.1 Preliminaries

Let $\mathbf{X} = [x_1, x_2, \ldots, x_n]$ denote an input token sequence of length n, with $x_1 =$ `[CLS]` as the prepended classification token. A pre-trained BERT encoder maps $\mathbf{X}$ to a matrix of contextualised token representations:

$$\mathbf{H} = \text{BERT}(\mathbf{X}) \in \mathbb{R}^{n \times d} \tag{1}$$

where d is the hidden dimension size (e.g., $d = 768$ for `bert-base`). Each row $\mathbf{h}_i \in \mathbb{R}^d$ is the contextualised representation of token x_i, and $\mathbf{H}^j \in \mathbb{R}^n$ denotes the j-th column of $\mathbf{H}$, i.e., the activations of all n tokens along hidden dimension j. A pooling function $\phi : \mathbb{R}^{n \times d} \to \mathbb{R}^d$ aggregates $\mathbf{H}$ into a fixed-size sentence representation $\mathbf{s} = \phi(\mathbf{H})$, which is subsequently passed to a classification or regression head.

3.2 Existing Pooling Strategies

CLS Pooling. The standard BERT approach uses only the representation of the `[CLS]` token as the sentence representation:

$$\mathbf{s}_{\text{CLS}} = \mathbf{h}_1 \tag{2}$$

While simple and effective on clean data, `[CLS]` is shaped by all tokens through the self-attention mechanism, making it susceptible to adversarial token injection.

Global Average Pooling (GAP). GAP averages the representations of all tokens:

$$\mathbf{s}_{\text{GAP}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{h}_i \tag{3}$$

Every token contributes equally, meaning adversarially injected tokens always influence the pooled representation.

Global Max Pooling (GMP). GMP takes the element-wise maximum across all token representations:

$$s^j_{\text{GMP}} = \max_{i \in [n]} h^j_i \quad \forall j \in [d] \tag{4}$$

While selective, GMP is dominated by the single maximum-activating token per dimension, which is precisely the token an adversary would target with a high-magnitude injection.

3.3 Per-Dimension Top-K Average Pooling

We propose **Per-Dimension Top-K Average Pooling**, which for each hidden dimension j selects the K highest-valued token activations and averages them:

$$s^j_{\text{TopK}} = \frac{1}{K} \sum_{i \in \mathcal{T}^j_K} h^j_i \quad \forall j \in [d] \tag{5}$$

where $\mathcal{T}^j_K \subseteq [n]$ is the index set of the K tokens with the largest activations in dimension j:

$$\mathcal{T}^j_K = \underset{\substack{S \subseteq [n] \\ |S| = K}}{\arg\max} \sum_{i \in S} h^j_i \tag{6}$$

The full pooled representation is:

$$\mathbf{s}_{\text{TopK}} = \left[s^1_{\text{TopK}}, s^2_{\text{TopK}}, \ldots, s^d_{\text{TopK}} \right] \in \mathbb{R}^d \tag{7}$$

Note that Top-K Average Pooling generalises both GAP (when $K = n$) and GMP (when $K = 1$), interpolating between full sequence coverage and single-token selection. The choice of K thus directly controls the trade-off between representational breadth and token selectivity. Critically, this pooling operation requires no additional parameters, no modification to the BERT encoder, and no task-specific augmentation.

3.4 Hybrid Variant: `[CLS]` + Top-K

To combine the global sentence-level signal encoded in the `[CLS]` token with the adversarially robust local aggregation of Top-K pooling, we introduce a **Hybrid** representation formed by concatenating the two:

$$\mathbf{s}_{\text{Hybrid}} = \left[\mathbf{s}_{\text{CLS}} \parallel \mathbf{s}_{\text{TopK}} \right] \in \mathbb{R}^{2d} \tag{8}$$

where $\parallel$ denotes vector concatenation. This doubles the input dimensionality to the classification head but preserves the full expressive power of the `[CLS]` representation while supplementing it with a token-selective aggregate. In practice, we also evaluate Hybrid variants where Top-K is replaced by GMP (Hybrid-GMP) or GAP (Hybrid-GAP) as ablation baselines.

3.5 Theoretical Analysis

We now formalise why Per-Dimension Top-K Pooling is structurally more resistant to adversarial token injection than CLS or GAP.

Theorem 1 (Bounded Adversarial Influence under Top-K Pooling)**.** *Let $\mathbf{H} \in \mathbb{R}^{n\times d}$ be the clean BERT token representation matrix, and let $\mathbf{a} \in \mathbb{R}^d$ be the representation of a single adversarially injected token appended to the sequence, yielding $\mathbf{H}' \in \mathbb{R}^{(n+1)\times d}$. Let $h^j_{(K)}$ denote the K-th largest clean activation in dimension j. Then:*

(i) ***GAP:*** *The adversarial perturbation to the pooled representation in dimension j is:*

$$|\Delta s^j_{GAP}| = \frac{|a^j - \bar{h}^j|}{n+1} > 0 \qquad (9)$$

i.e., every injected token always *perturbs the GAP representation.*

(ii) ***Top-K Pooling:*** *The adversarial perturbation in dimension j is:*

$$|\Delta s^j_{TopK}| = \begin{cases} \frac{1}{K}\left|a^j - h^j_{(K)}\right| & \text{if } a^j > h^j_{(K)} \\ 0 & \text{otherwise} \end{cases} \qquad (10)$$

i.e., an injected token perturbs the Top-K representation in dimension j only if *its activation exceeds the top-K threshold $h^j_{(K)}$.*

Proof. (i) Under GAP, the pooled value in dimension j over $n+1$ tokens is $\frac{1}{n+1}(\sum_{i=1}^n h^j_i + a^j)$, compared to $\frac{1}{n}\sum_{i=1}^n h^j_i = \bar{h}^j$ over n clean tokens. The difference is $\frac{a^j - \bar{h}^j}{n+1}$, which is nonzero whenever $a^j \neq \bar{h}^j$, i.e., almost surely for any nontrivial injection.

(ii) Under Top-K pooling, the injected token a^j enters the top-K selection for dimension j if and only if $a^j > h^j_{(K)}$. If it does not enter, the selected set $\mathcal{T}^j_K$ is unchanged and $|\Delta s^j_{\text{TopK}}| = 0$. If it does enter, it displaces $h^j_{(K)}$ from the selection, yielding a change of $\frac{1}{K}(a^j - h^j_{(K)})$. □

Corollary 1.1. *As K decreases, $h^j_{(K)}$ increases (since higher-ranked activations have larger values), raising the entry threshold for adversarial tokens. Consequently, smaller values of K provide stronger structural protection against adversarial injection, at the cost of reduced sequence coverage.*

Note that Theorem 1 formally bounds only the GAP comparison; a rigorous perturbation bound for `[CLS]` is non-trivial as its sequence integration occurs implicitly through BERT's self-attention mechanism. We leave this as an open problem, with Table 5 providing indirect empirical evidence that `[CLS]` undergoes substantially larger representational displacement under attack.

This directly predicts the empirical findings in Section 5: Hybrid (K=3) achieves the lowest ASR (37.07%) and smallest cosine shift (0.359), with only 9.1% of adversarially injected dimensions entering the top-K selection.

3.6 Classification and Regression Head

The pooled representation $\mathbf{s}$ (of dimension d for Top-K/GAP/GMP and $2d$ for Hybrid variants) is passed to a task-specific head. For classification tasks (spam detection and hate speech), we apply a linear layer followed by softmax:

$$\hat{y} = \text{softmax}(\mathbf{W}\mathbf{s} + \mathbf{b}) \qquad (11)$$

where $\mathbf{W} \in \mathbb{R}^{C\times d'}$ and C is the number of classes. For the regression task (AES), we apply a linear layer followed by a sigmoid activation scaled to the essay score range, optimised using mean squared error loss. All pooling strategies share identical head architectures to ensure that performance differences are attributable solely to the pooling mechanism.

4 Dataset

4.1 Data Sources

For AES, we use the ASAP dataset[1], containing 12,976 student essays across eight prompts scored by human raters on domain-specific rubrics.

For adversarial spam detection, we use two complementary datasets: the LingSpam corpus[2] and the Enron spam dataset.[3] LingSpam is a smaller, highly structured corpus of 2,832 messages from a linguistics mailing list, with a spam prevalence of roughly 16.9%. Its constrained vocabulary and formal register make it a useful probe for how pooling strategies behave under low-noise conditions. The Enron dataset comprises approximately 33,717 emails across six pre-labelled subsets reflecting real-world lexical diversity.

4.2 Preprocessing

All text is lowercased and tokenised using the BERT WordPiece tokeniser (Schuster and Nakamura, 2012) with a maximum sequence length of

[1]ASAP-AES: https://www.kaggle.com/c/asap-aes/

[2]LingSpam: https://www.kaggle.com/datasets/mandygu/lingspam-dataset

[3]Enron: https://www2.aueb.gr/users/ion/data/enron-spam/

512 tokens for ASAP, 512 tokens for Adversarial Spam Detection and 128 tokens for Hate Speech Classification. Emails are truncated by removing redundant headers prior to tokenisation. Spam mail datasets are lightly normalised: HTML artefacts and non-ASCII characters are removed, but punctuation and capitalisation are preserved as stylistic signals.

For adversarial evaluation, Magic Word injection is applied *only to test-set malicious samples* using a fixed vocabulary of 15 benign tokens selected for their strongly positive connotations. Each malicious sample is attacked at tail (tokens appended) yielding one adversarial variant per original sample.

4.3 Statistics

Domain	Dataset	#Exps	Avg. Len	%Positive
Spam	Enron	33,717	214.3	49.1%
	LingSpam	2,832	87.6	16.9%
AES	ASAP	12,976	326.1	N/A
Hate Speech	Implicit Hate Corpus	6,346	96.1	N/A

Table 1: Dataset statistics. Avg. Len is average email length. %Positive is the proportion of spam/hate samples; AES is a regression task.

5 Experimental Setup

5.1 Baselines

We compare `[CLS]` pooling (primary baseline), GMP (Top-K with K=1), GAP (Top-K with K=n), Top-K Average Pooling across $K \in \{3, 5, 10, 20, 50\}$, and Hybrid (`[CLS]` + Top-$K$) concatenation variants.

5.2 Implementation Details

All models are implemented in PyTorch using the HuggingFace Transformers library (Wolf and et al, 2020) with `bert-base-uncased` (110M parameters) as the encoder backbone. Experiments are conducted on a single NVIDIA A6000 (40 GB) GPU. All token pooling operations exclude the `[CLS]` token and ignore padding positions, operating only over the n genuine input tokens. Key hyperparameters are fixed across all configurations: learning rate 5×10^{-5} (AdamW), weight decay 0.01, batch size 64, maximum sequence length 512 tokens, and 2 training epochs. Pre-tokenisation of the full dataset is performed once per fold before training begins. Code and experimental scripts are available at the repository:https://github.com/ShivamAralikatti/Not-All-Tokens-Are-Equal.

5.3 Adversarial Attack: Magic Word Injection

Magic words are extracted per fold using a TF-IDF + LinearSVC pipeline. Feature weights identify tokens that push the classifier toward the ham class; these are appended exclusively to spam messages in the test set. This design prevents leakage. A budget of 15 tokens is used as the primary setting.

5.4 Evaluation Protocol for Spam Detection and Hate Speech

Experiment 1 to 3 are for Spam detection, experiment 4 is for hate speech detection. Experiment 5 is for AES which is in the next subsection.

Experiment 1 — Repeated Seeds (5×5-fold CV). We run the main K-sweep under repeated cross-validation using 5 independent random seeds × 5 stratified folds, yielding 25 training cycles per configuration. Results are aggregated to produce mean clean accuracy, adversarial accuracy, Macro-F1, and ASR. Statistical significance relative to `[CLS]` is assessed using McNemar's test on pooled out-of-fold predictions.

Experiment 2 — Cosine Similarity Shift. For each spam message that `[CLS]` fails under attack, we compute the cosine shift $\Delta_{\cos}(f, x) = 1 - \cos(f(x), f(\tilde{x}))$ between clean and adversarial representations, measuring geometric displacement in embedding space across all five folds. A smaller $\Delta_{\cos}$ indicates a more stable representation under injection. This analysis is complementary to ASR: while ASR measures the downstream classification effect, cosine shift directly measures representational stability in embedding space.

Experiment 3 — Dimensional Activation Analysis. For the top-12 diagnostically active spam dimensions, we compare which tokens achieve maximum activation before and after injection, measuring the fraction of dimensions where injected tokens enter the top-K selection and the margin by which they do so.

Experiment 4 — Hate Speech. Hate Speech classification experiment focused on fine-grained categorization of implicit hate speech into the 6-class taxonomy performing 5 fold CV of 3 mod-

els (Hybrid, Topk, CLS). Note that no adversarial evaluation is performed for this task; results reflect clean-data performance only. The purpose of this experiment is to assess whether Top-K pooling incurs a clean-accuracy cost relative to CLS in a multi-class setting.

5.5 Experiment 5 — AES: Top-K Pooling within the R2BERT Framework

The previous experiments establish Top-K Pooling as an adversarially robust replacement for the `[CLS]` bottleneck in spam and hate speech classification. To test whether the same pooling strategy generalises to a fundamentally different high-stakes task—one characterised by ordinal regression over long documents rather than binary classification under token injection—we integrate Top-K Pooling into the R2BERT framework (Yang et al., 2020) and evaluate on the ASAP automated essay scoring benchmark.

Implementation Details. All models use `bert-base-uncased` trained with AdamW, weight decay 0.01, batch size 32, mixed-precision (AMP), and 5-fold CV with 60/20/20 train/val/test splits. The best checkpoint per fold is selected by validation QWK; the reported metric is macro-average QWK across all eight ASAP prompts.

Baseline: R2BERT with `[CLS]` Pooling. We re-implemented R2BERT following the original architectural specification, using a dynamically weighted combination of MSE regression loss and ListNet ranking loss with a sigmoid-shaped weight schedule (see Appendix A for full details). Our re-implementation achieves an average QWK of 0.769 across all eight ASAP prompts under 5-fold cross-validation, compared to the paper-reported 0.792; we attribute this gap to undisclosed implementation details, as no official code was released.

Top-K Average Pooling. We replace the `[CLS]` representation with Per-Dimension Top-K Average Pooling (Equation 5), holding all other components of the R2BERT framework—the combined loss, dynamic weight schedule, bias initialisation, and training procedure—identical. The `[CLS]` token is excluded from the pooling operation; only the n content token positions contribute to the top-K selection. All inputs are truncated to 512 WordPiece tokens. This isolates the contribution of the pooling strategy within an otherwise unchanged training pipeline.

K Sweep. To identify the optimal pooling width within the R2BERT training regime, we sweep $K \in \{1, 3, 5, 10, 20, 50, 768\}$ using 60 epochs, lr $= 2 \times 10^{-5}$, and $\tau_1 = 10^{-4}$. Results are reported in Table 3.

6 Experimental Results

Tables 2–4 report results across all tasks and pooling configurations. We discuss the key findings below.

Model	Clean		Adv		ASR
	Acc	F1	Acc	F1	
Enron					
CLS (Baseline)	99.27	99.27	63.64	58.85	70.65
GAP	99.38	99.38	76.20	74.33	46.27
GMP	99.00	99.00	72.83	70.47	52.87
Hybrid (K=GMP)	99.42	99.42	74.70	71.36	49.04
Hybrid (K=3)	**99.51**	**99.51**	**80.86**	**80.00**	**37.07**
Hybrid (K=5)	99.51	99.51	71.67	69.17	54.97
Hybrid (K=10)	99.18	99.18	60.77	54.45	76.57
Hybrid (K=20)	99.16	99.16	64.47	58.72	69.06
Hybrid (K=50)	99.35	99.35	72.33	68.99	53.70
Hybrid (K=GAP)	99.31	99.31	74.88	72.12	48.32
Top-K (K=3)	98.57	98.57	74.60	72.24	49.09
Top-K (K=5)	99.44	99.44	72.79	69.33	52.79
Top-K (K=10)	99.23	99.23	75.06	72.43	48.08
Top-K (K=20)	99.33	99.33	68.96	65.25	60.44
Top-K (K=50)	98.81	98.81	71.35	66.71	55.93
LingSpam					
CLS (Baseline)	99.44	98.99	92.49	82.90	43.64
GAP	99.34	98.82	91.49	80.75	49.06
GMP	99.37	98.87	91.60	81.03	48.85
Hybrid (K=GMP)	99.34	98.82	90.84	78.78	53.43
Hybrid (K=3)	99.41	98.95	91.39	80.47	49.68
Hybrid (K=5)	99.45	99.00	91.15	79.59	51.98
Hybrid (K=10)	**99.52**	99.12	91.50	80.41	50.31
Hybrid (K=20)	99.38	98.88	91.36	80.17	50.31
Hybrid (K=50)	99.38	98.89	91.81	81.59	47.00
Hybrid (K=GAP)	99.24	98.62	91.77	81.45	48.03
Top-K (K=3)	99.07	98.32	91.74	81.50	47.20
Top-K (K=5)	99.52	**99.13**	**92.85**	**83.99**	**41.78**
Top-K (K=10)	99.41	98.94	92.19	82.68	45.32
Top-K (K=20)	99.38	98.89	91.63	81.00	48.23
Top-K (K=50)	98.96	98.11	90.74	78.70	53.22

Table 2: Spam detection results . Acc = accuracy (%), F1 = macro-F1 (%), and ASR = attack success rate (%); lower is better. Bold indicates the best result within each dataset.

Enron vs. LingSpam divergence. The contrast between Enron and LingSpam results We attribute this to LingSpam's shorter sequences (87.6 vs. 214.3 tokens), lower spam prevalence (16.9%), and the already-low CLS baseline ASR of 43.64%, all of which reduce the structural advantage of Top-K suppression. The mechanism is most effective

Model	QWK
R2BERT (CLS)	76.92
BERT (GMP)	76.46
BERT (GAP)	**77.82**
Hybrid (GMP)	77.31
Hybrid (K=3)	77.57
Hybrid (K=5)	77.43
Hybrid (K=10)	77.50
Hybrid (K=20)	77.18
Hybrid (K=50)	77.36
Hybrid (GAP)	77.36
Top-K (K=3)	76.95
Top-K (K=5)	77.13
Top-K (K=10)	77.07
Top-K (K=20)	**77.79***
Top-K (K=50)	77.56

Table 3: Performance on the ASAP-AES dataset measured using Quadratic Weighted Kappa (QWK). Higher is better. * denotes the best-performing Top-K variant.

Model	Accuracy	Macro F1
CLS	62.06	58.59
GMP	62.43	57.72
GAP	60.72	57.29
Hybrid (K=GMP)	62.37	**58.91**
Hybrid (K=3)	61.69	57.59
Hybrid (K=5)	61.66	57.47
Hybrid (K=10)	**62.53**	57.86
Hybrid (K=20)	61.93	57.98
Hybrid (K=50)	61.42	56.88
Hybrid (K=GAP)	61.20	56.85
Top-K (K=3)	61.94	58.53
Top-K (K=5)	60.21	55.75
Top-K (K=10)	61.88	58.14
Top-K (K=20)	62.15	58.06
Top-K (K=50)	61.99	58.24

Table 4: Hate-speech classification results (%). Best values in each metric are shown in bold.

when injected tokens form a small fraction of a longer, lexically diverse sequence — conditions that favour Enron over LingSpam.

In Table 5 we show the results for cosine similarity shift and in Table 6 we show the results for Dimension-level activation analysis for the spam detection over Enron data.

Model	Mean Shift	Mean Cosine Similarity
CLS	0.7803	0.2197
Hybrid	**0.3591**	**0.6409**
Top-K	0.6095	0.3905

Table 5: Cosine Similarity shift for Spam detection Enron data

Model	#Rows	Frac. Dims	Mean Margin
Hybrid(K=3)	33732	**0.0909**	**0.6942**
Top-K(K=10)	6912	0.3031	0.1405

Table 6: Top-K entry analysis for robust models under adversarial injection. *# Rows* denotes the number of samples where CLS failed but the model persisted. *Frac. Dims* denotes the fraction of dimensions in which injected benign tokens enter the selected Top-K set. *Mean Margin* is the mean difference between the K-th selected activation and the maximum injected-token activation in the adversarial example. Lower entry fraction and larger margin indicate greater robustness.

6.1 Guidance on Selecting K

Small K values favour adversarial robustness (Hybrid K=3 achieves ASR 37.07% vs. 76.57% for K=10), while **clean accuracy remains largely insensitive** to K (98.57–99.51% on Enron). **Moderate K** 10, 20 is **preferable for regression tasks** such as AES, where broader sequence coverage is beneficial. The Hybrid variant is a safe default in new domains, rarely degrading below the `[CLS]` baseline.

7 Conclusion

We proposed Per-Dimension Top-K Average Pooling, a lightweight and architecture-agnostic pooling strategy that introduces explicit token-level selectivity into BERT-based text classifiers. We formalised this through a bounded adversarial influence theorem, showing that injected tokens are structurally suppressed unless their activations exceed the top-K threshold — a property not shared by CLS, GAP, or GMP.

Empirically, our approach reduces the Attack Success Rate from 70.65% to 37.07% on Enron spam detection while maintaining clean accuracy above 99%. Cosine similarity shift and dimension-level activation analyses confirm these gains are mechanistically principled rather than incidental. On LingSpam, gains are present but modest; we attribute this to the dataset's shorter sequences, constrained vocabulary, and lower spam prevalence, which together reduce the structural advantage of top-K suppression. On ASAP-AES, Top-K (K=20) is competitive with the best pooling baseline, and hate speech results demonstrate broad task applicability on clean data.

Future work includes providing a formal perturbation bound for the `[CLS]` representation through BERT's self-attention mechanism, bound-

ing adversarial influence through BERT's self-attention mechanism, evaluating against broader attack paradigms (including substitution-based attacks such as TextFooler and BERT-ATTACK, and adaptive adversaries aware of the Top-K mechanism), and investigating learned K selection as an adaptive pooling strategy. Extending the evaluation to larger BERT variants (e.g. `bert-large`) and other Transformer architectures would further establish the generality of the method.

Limitations

The primary empirical findings of this work are demonstrated on the Enron and LingSpam spam detection datasets under a specific adversarial attack setting (magic-word injection with a budget of 15 tokens). While results across AES and hate speech classification corroborate the general utility of Top-K pooling on clean data, the adversarial robustness gains are most pronounced in the spam detection setting and may not generalise uniformly to other tasks, domains, or attack paradigms such as character-level perturbations or paraphrase-based attacks. In particular, the evaluation does not include substitution-based attacks (e.g. TextFooler (Jin et al., 2020), BERT-ATTACK (Li et al., 2020b)), universal adversarial triggers (Wallace et al., 2019), or adaptive attacks specifically designed to exploit the Top-K mechanism — for example, an adversary distributing perturbation energy across many tokens at sub-threshold activation levels. Such adaptive attacks may circumvent the structural suppression effect described in Theorem 1 and evaluating Top-K pooling in this setting is an important open question. The theoretical analysis in Theorem 1 is bounded to the pooling layer and does not formally account for adversarial influence propagated through BERT's self-attention mechanism. Specifically, a formal perturbation bound for `[CLS]` — the primary empirical comparison target — is absent from the theoretical analysis, as formalising the effect of token injection through multi-layer self-attention is non-trivial and left for future work. Additionally, all experiments use `bert-base-uncased`; behaviour may differ for larger models (e.g., `bert-large`) or other Transformer architectures. The hate speech experiment does not include adversarial evaluation. The hate speech results demonstrate only that Top-K pooling does not degrade clean-data performance relative to CLS. Finally, the optimal value of K is task- and condition-dependent and requires validation on a held-out set, which may be a practical limitation in low-resource settings.

Ethics Statement

This work proposes a pooling mechanism for improving adversarial robustness in BERT-based text classifiers, with primary evaluation on spam detection, automated essay scoring, and hate speech classification. We do not introduce new datasets, crowdsourced annotations, or human subject studies. The Enron, LingSpam, ASAP, and hate speech datasets used in this work are publicly available benchmarks widely used in the NLP community. Improved adversarial robustness in spam and hate speech classifiers has direct positive societal implications, as it reduces the ability of malicious actors to evade automated content moderation systems through token injection. We acknowledge, however, that more robust classifiers could also be misused in adversarial contexts where suppression of legitimate speech is a concern, and we encourage responsible deployment in conjunction with human oversight. No personally identifiable information was used or generated in this work.

A R2BERT Framework: Loss Function Details

The R2BERT baseline uses a dynamically weighted combination of a regression loss and a ranking loss:

$$\mathcal{L} = \tau(e)\mathcal{L}_m + (1 - \tau(e))\,\mathcal{L}_r \tag{12}$$

where $\mathcal{L}_m = \text{MSE}(\hat{s}, s^*)$ is the regression loss on normalised scores, $\mathcal{L}_r$ is the batchwise ListNet top-1 cross-entropy ranking loss, and $\tau(e)$ is a sigmoid-shaped dynamic weight schedule that transitions from ranking-dominant at early epochs to regression-dominant at later epochs:

$$\tau(e) = \frac{1}{1 + \exp\left(\gamma\left(\frac{E}{2} - e\right)\right)} \tag{13}$$

with γ solved such that $\tau(1) \approx 10^{-6}$. The linear head bias is initialised to $\text{logit}(\bar{s}_{\text{train}})$, the logit of the mean training score, following the paper's initialisation strategy. Scores are normalised per-prompt to $[0, 1]$ using min-max scaling and denormalised before computing Quadratic Weighted Kappa (QWK).

B Cosine Similarity Shift: Extended Analysis

For completeness, we restate the cosine shift metric used in Experiment 3. For each spam message x that `[CLS]` fails to classify correctly after adversarial injection, and its adversarially perturbed variant $\tilde{x}$, the cosine shift is:

$$\Delta_{\cos}(f, x) = 1 - \cos\left(f(x),\ f(\tilde{x})\right) \tag{14}$$

where $f(\cdot) \in \mathbb{R}^{768}$ is the pooled representation produced by pooling strategy f. A smaller $\Delta_{\cos}$ indicates a more stable representation under adversarial injection. Distributions are computed over all spam test samples across all five folds with a magic word budget of 15 tokens.

Table 7 reports the mean shift and mean cosine similarity for all three evaluated configurations.

Model	Mean Shift	Mean Cosine Sim.
CLS	0.7803	0.2197
Hybrid (K=3)	0.3591	0.6409
Top-K (K=10)	0.6095	0.3905

Table 7: Cosine similarity shift for spam detection on Enron. Lower mean shift indicates greater representational stability under adversarial injection.

C Dimension-Level Activation Analysis: Extended Details

Experiment 4 identifies the top-12 most diagnostically active dimensions for the spam class by computing the mean activation difference between spam and ham representations in the clean test set. For each diagnostic dimension d^*, we compare the token achieving the maximum activation in the clean versus adversarially perturbed input. The key metrics reported are: (i) the fraction of dimensions in which injected tokens enter the top-K selection (*Frac. Dims*), and (ii) the mean margin between the K-th selected activation and the maximum injected-token activation (*Mean Margin*). Lower entry fraction and larger margin indicate greater structural robustness.

Table 8 reports extended results for the two robust model configurations evaluated in this analysis.

D Hyperparameter Summary

Table 9 summarises all key hyperparameters used across experiments.

Model	#Rows	Frac. Dims	Mean Margin
Hybrid (K=3)	33,732	0.0909	0.6942
Top-K (K=10)	6,912	0.3031	0.1405

Table 8: Top-K entry analysis under adversarial injection on Enron. #Rows denotes the number of samples where `[CLS]` failed but the model persisted. Frac. Dims is the fraction of dimensions where injected tokens entered the top-K selection. Mean Margin is the mean gap between the K-th selected activation and the maximum injected-token activation.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of EMNLP*, pages 2890–2896. Association for Computational Linguistics.

Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, Georgios Paliouras, and Constantine D. Spyropoulos. 2000. An evaluation of naive bayesian anti-spam filtering. In *Proceedings of the Workshop on Machine Learning in the New Information Age, ECML*, pages 9–17.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-box adversarial examples for text classification. In *Proceedings of ACL*, pages 31–36. Association for Computational Linguistics.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hongyang Gao and Shuiwang Ji. 2019. Graph u-nets. In *Proceedings of ICML*, pages 2083–2092.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Di Jin, 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. In *Proceedings of AAAI*, pages 8018–8025.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*, pages 655–665. Association for Computational Linguistics.

Hyperparameter	Spam / Hate Speech	AES
Encoder backbone	`bert-base-uncased`	`bert-base-uncased`
Max sequence length	512 (spam), 128 (hate)	512
Optimiser	AdamW	AdamW
Learning rate	5×10^{-5}	2×10^{-5}
Weight decay	0.01	0.01
Batch size	64	32
Training epochs	2	60
Precision	FP32	Mixed (AMP)
CV protocol	5×5-fold	5-fold
GPU	NVIDIA A6000 (40 GB)	NVIDIA A6000 (40 GB)
Magic word budget	15 tokens	N/A

Table 9: Hyperparameter settings across all experiments.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, pages 1746–1751. Association for Computational Linguistics.

Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. In *Proceedings of ICML*, pages 3734–3743.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020a. On the sentence embeddings from pre-trained language models. In *Proceedings of EMNLP*, pages 9119–9130. Association for Computational Linguistics.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. BERT-ATTACK: Adversarial attack against BERT using BERT. In *Proceedings of EMNLP*, pages 6193–6202. Association for Computational Linguistics.

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

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. In *Proceedings of ICLR*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*, pages 3982–3992. Association for Computational Linguistics.

Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. 1998. A bayesian approach to filtering junk E-mail. In *Proceedings of the AAAI Workshop on Learning for Text Categorization*.

Mike Schuster and Kaisuke Nakamura. 2012. Japanese and korean voice search. In *Proceedings of ICASSP*, pages 5149–5152.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of EMNLP*, pages 1882–1891. Association for Computational Linguistics.

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

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal adversarial triggers for attacking and analyzing NLP. In *Proceedings of EMNLP-IJCNLP*, pages 2153–2162. Association for Computational Linguistics.

Thomas Wolf and et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of NAACL-HLT*, pages 1480–1489. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. In *Proceedings of NAACL-HLT*, pages 1415–1420. Association for Computational Linguistics.