Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 7386888, 19 pages
https://doi.org/10.1155/2023/7386888

Research Article

Augmenting Feature Representation with Gradient Penalty for Robust Text Categorization

Depei Wang,[1] **Lianglun Cheng**,[2] **and Zhuowei Wang**[2]

[1]*School of Automation, Guangdong University of Technology, Guangzhou 510006, China*
[2]*School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China*

Correspondence should be addressed to Zhuowei Wang; wangzhuowei0710@163.com

Received 23 September 2022; Revised 8 November 2022; Accepted 20 December 2022; Published 20 February 2023

Academic Editor: Jin Li

The capabilities of deep models are constantly mined for extraction and representation of features among text classification tasks. However, these models are sensitive to changes in input data, resulting in poor robustness. Meanwhile, the model lacks information interaction and weak representation ability. In this work, for feature extraction, a joint model that consists of a convolutional neural network, a bidirectional gated recurrent unit, and an attention mechanism is proposed. This new model can improve versatility and fully discover category information in text. For feature representation, a projector under the supervised contrastive learning method is introduced. The method can improve the representation of an encoder and realize aggregation of the same category. Considering the robustness of the PCRA, the gradient penalty is added to a contrastive loss function. Experiments are performed on four datasets to assess the proposed model (PCRA and PCRA-GP) using an accuracy metric. The experimental results show that our model is suitable for variable-length and bilingual texts. Compared with the baseline model, it remains competitive, and it reaches SOTA on the 20 Newsgroups dataset. Moreover, the performance of the model is evaluated under different hyperparameters to clarify its working mechanism.

1. Introduction

Deep learning is constantly expanding its application areas, such as convolutional and recurrent neural networks and their increasing use in image processing [1], natural language processing (NLP) [2], and many other scenarios [3], accompanied by the steady increase in computing power and data volume which are able to be processed. In text category tasks, deep learning models can automatically obtain the features of input samples and perform self-learning feature representation. However, the extracted features from training data are solidified, causing the fragility of representation. Due to the weak generalization ability of representation, models are vulnerable to the attack of the input side, resulting in wrong decisions. Therefore, there are great needs for studying model robustness in text classification tasks at present.

Initially, adversarial examples appeared in image-processing tasks. Szegedy et al. [4] added small disturbances to samples to propose the concept of adversarial samples for the first time. Inspired by this, many data augmentation methods are used to improve the performance of deep learning models, and common methods entail elastic distortion, scaling, translation, and rotation in image classification. In the NLP domain, which is analogous to pixel-level disturbances in images, perturbation is added to the input by manipulating words in the sentence, such as adding or subtracting characters and swapping positions. Specifically for classification tasks, it not only requires enhanced features but also high-quality representation capabilities of the input.

Representation learning (RL) encodes a high-dimensional input into a much lower dimensional space, which captures high-level and useful concepts, and the mapping process involves the conversion of raw input data into a feature vector or tensor. Fortunately, contrastive learning (CL) is an easy way to meet these constraints [5]. It uses a self-supervised paradigm to compare and learn the distribution of positive samples and negative samples so that similar species are closer and heterogeneous species are further apart in the mapping space. Besides, the selection of the projection head can be linear or nonlinear after encoding. Inspired by triplet, max-margin, and N-pairs loss, supervised contrastive learning (SCL) introduces CL to the fully supervised method. Researchers have never stopped in their quest to model expressiveness. Therefore, the feature representation ability is still worthy of attention.

In this paper, a multichannel joint architecture, PCRA-GP, is established based on SCL with the gradient penalty. It aims to address the model's robustness through adversarial training of feature representation. First, the multichannel convolutional neural network (CNN, C) that enhances the adaptability to sentence length, the bidirectional gated recurrent unit (BiGRU, R) that extracts contextual information, and the attention mechanism (A) that calculates the contribution of different features are integrated. Second, the projector (P) of contrastive learning is pretrained to improve the representation ability of features with a contrastive loss function. Finally, the gradient penalty (GP) is used to generate perturbation at the feature level, and the robust representation of text features produced by using the PCRA encoder is realized. The main contributions of our work are summarized as follows:

(i) To realize feature extraction, a joint deep learning model that consists of the CNN, BiGRU and attention mechanism is established, which are used to adapt to sentence length, capture contextual information, and filter features. The joint model can efficiently improve versatility and fully detect category information in text.

(ii) For feature representation, contrastive learning and a projector are introduced under the supervised method. The method can improve the representation of an encoder and realize aggregation of data that belong to the same category.

(iii) For model robustness, the gradient penalty is added to the contrastive loss function, which achieves robust training.

The remainder of this paper is organized as follows: Section 2 reviews contrastive learning, generation of adversarial samples, and deep learning models for text classification. Section 3 describes the proposed PCRA-GP architecture. Section 4 introduces the experimental setup in detail, conducts comparison experiments, and analyzes the experimental results. Section 5 summarizes the research.

2. Related Work

2.1. Contrastive Learning. Recently, CL has been widely applied to self-supervised RL for CV [6, 7], NLP audio [8], graph [9], multimodal methods, and other domains. CL is a self-supervised learning method inspired by noise contrastive estimation or N-pair losses. It learns representations through comparison between different input samples, closing similar inputs and repelling dissimilar inputs. Le-Kha et al. [5] declared that CL methods provide a simple yet powerful approach to learning representations in a discriminative manner in both supervised and self-supervised setups. PIRL [7] verified this conclusion by learning invariant representations based on pretext task solving jigsaw puzzles and found that the resulting invariant representations perform better than their covariant counterparts across a range of vision tasks in self-supervised and supervised manners.

There remains much research exploring the potential of CL. To maintain the current negative candidate pool and adopt momentum update strategy passing parameters between positive and negative samples, He et al. proposed the MoCo [6] method, a mechanism for building dynamic dictionaries for CL. The SimCLR [10] framework introduced learnable nonlinear transformations and two data augmentation methods to train an encoder and compared the effects of data augmentation, batch sizes, and training steps on the training results. Common cropping and rotation transformations are no longer applicable to serialized data such as texts and audio. A universal unsupervised learning approach CPC [8] architecture that extracts features from high-dimensional data and uses autoregessive models to predict the future in latent space was proposed.

Beyond self-supervised classification, SCL [1] is an extension of traditional CL. It pulls the same class closer using SupCon loss by leveraging label information. SsCL [11] combined the CL loss branch with the cross-entropy loss branch in semisupervised learning and introduced a cocalibration mechanism to interchange predictions between the two branches. RoCL [12] trained a robust neural network without label information by using a contrastive self-supervised learning framework. Also, the CosG [9] model is a graph-based CL method for fact verification including label-supervised and unsupervised graph-contract. The former helps the model learn discriminative representation for items of various classes, and the latter trains a graph convolutional encoder for reducing the loss of sole node features in graph propagation.

2.2. Adversarial Example Generation. In NLP systems, studies have shown that the model is highly dependent on the input data and have used this discovery to conduct robust training of the model through adversarial samples. Belinkov and Bisk [13] found that even with spell checking, existing models are still sensitive to the spelling order of words in machine translation tasks. For character-level attacks, perturbation can be approximated by the number of character edits. For word-level attacks, perturbation can be achieved by substitution. EDA [14], an easy data augmentation method, consists of four simple operations: synonym replacement, random insertion, random swap, and random deletion. Similarly, there are some methods based on word semantic similarity. Kobayashi proposed a contextual augmentation [15] strategy, which uses a bidirectional language model to generate a diverse alternative vocabulary according to the context. Exploring the broadening of the scope of application, round-trip translation constructs data through translation, which is suitable for words, phrases, sentences, and texts.

Recent adversarial attack example generation algorithms can be generally categorized into two groups: white-box or black-box methods. In the white-box setting, an adversary has access to the model, model parameters, and feature set of inputs. For perturbation at the embedding level, TextTricker [16] calculated the gradient magnitude of each input unit and added the scores of each dimension in the embedding space as the word-level importance score when identifying key words from input. Perturbating at the grade level, HotFlip [17] relies on an atomic flip operation to generate white-box adversarial examples that trick character-level and word-level neural models. It swaps one token for another based on the gradients of one-hot input vectors. Papernot et al. proposed FGSM [18], a fast gradient sign method widely applied in the image domain [19], which could be solved by linearizing the cost function of the model around its input and selecting perturbation using the gradient of the cost function with respect to the input itself. GBDA [20], a gradient-based distributional text attack against a transformer framework, defines a parameterized distribution of adversarial examples and uses Gumbel-softmax approximation to derive a smooth estimate of the gradient. In the black-box setting, an adversary is only allowed to query the target classifier and does not know the details of learned models or the feature representations of inputs. In [21], AEG was proposed, a reinforcement-learning-based approach, to generate adversarial examples in black-box settings.

Intuitively, the generative adversarial network (GAN) generator captured the data distribution and made the discriminant equal to ½ everywhere in the space of its arbitrary function. After continuous research on GAN, WGAN [22] was developed by using the directed Wasserstein distance to produce a value function to solve the problem regarding original GAN training instability. Facing weight flipping caused by poor samples in WGAN, Gulrajani proposed a gradient penalty (WGAN-GP) algorithm [23] which can penalize the norm of the gradient of the critic with respect to its input. Moreover, in adversarial examples producing high quality and efficiency, Xiao produced AdvGAN [24] to generate adversarial examples with GANs in both semiwhite-box and black-box attack settings.

2.3. Deep Learning Models for Text Classification. The main basic components of the deep learning model in the text classification task are the recurrent neural network (RNN) [25], CNN [26, 27], attention mechanism [28, 29], and graph neural network (GNN) [30]. These basic components have tended to develop towards a fusion model. In the joint model, the advantages of local information, context information, and feature screening are combined. Huang and Liu proposed GCNN [31] and used CNN to compensate for the shortcoming of gated recurrent units (GRUs) in the extraction of long sentence information. Yang and Tang [32] tested the effect of the attention mechanism on different positions of the CNN-based model and performed experimental verification in ATCNN-1, ATCNN-2, and ATCNN-3 which superimposes formers combining feature weighting and feature selecting. To strengthen the feature extraction ability, Wenzhen et al. proposed the C-BiGRU-ATT [33] method to reduce the impact of text representation on the classification results and increase the features of the text at the vocabulary level and character level to improve classification accuracy. Also, in the face of semantic ambiguity, Liu and Guo proposed AC-BiLSTM [34], which is oriented to high-dimensional, discrete, and complex semantic phenomena for use on text classification tasks. This entailed the use of bidirectional long short-term memory (BiLSTM), an attention mechanism, and a convolutional layer.

The success of TextGCN [30] turned the serialized text model to the graph-structured text model. Wu et al. proposed SGC [35] to accelerate GCN by removing nonlinearities and collapsing weight matrices between consecutive layers. Zhu and Koniusz proposed SSGC to improve the performance of GCN and capture the global and local context of each node using simple spectral graph convolution. With the enrichment of the corpus and the development of increased computing power, a series of transformer models represented by Bert emerged. Lin et al. proposed that BertGCN [36] adopt transductive learning, learning representations for both training data and unlabeled test data, by propagating label influence through graph convolution. Zhu and Koniusz proposed SSGC [37] to solve transition smoothing and reduce computing and storage costs.

Inspired by large-scale pretraining models [38], researchers have begun to use external information [39, 40] to improve the classification accuracy of the model. Pan et al. [41] proposed SWEMs to transfer source domain knowledge to unseen text sequences. Abid et al. proposed BiGRU-CNN [28] based on pretrained GloVe embedding which retains domain knowledge. There are also models that do not rely on external knowledge. In the sparse classification of short text data, Liu et al. [42] proposed a bilevel attention model that does not rely on external knowledge to capture word-level information which is presented to explore the topic-word association and sequence-level information which is used to extract the relationship between local and global sentiment expression.

So far, we have discussed the development of CL, adversarial sample generation strategies, and deep models used in text classification tasks. Obviously, there is still potential for improvement in CL's robust representation. The adversarial sample generation technique based on the entire feature embedding is not addressed. The majority of deep models are focused on enhancing feature acquisition while ignoring model stability. To make the best of the text feature and robust representation content, a new neural network significantly improving the model classification robustness is proposed in this paper. The proposed model can efficiently represent enhanced features and learn the stable representation by input attacks.

3. Proposed Model

Here, we introduce the structure of the proposed model, main components, and realization of data-processing flow in detail. The model combines the powerful instance representation ability of contrastive learning and robustness brought about by GP. In the feature vector extraction part, the joint model of CNN, RNN, and the attention mechanism is used. The SCL projection method is adopted to obtain better representation capabilities. Finally, we add the gradient penalty to the SCL loss to obtain the robustness of the model to perturbation of the feature vector. The model processes the input data in a two-step process. In the first step, the embedding layer is used to vectorize the input. Next, CNNs, BiGRU, an attention mechanism, and a projector are integrated to construct an encoder. Then, the dataset is utilized to train the encoder to achieve a better representation of the input text. In the second step, the frozen encoder with the classifier layer realizes text classification. The overall structure of PCRA-GP is shown in Figure 1. The pseudocode of PCRA-GP is presented in Algorithm 1. The key points of our approach are described as follows.

The following is a summary of the implementation process of the PCRA-GP model for the text classification task:

Step 1: Embedding is implemented on training data T_{train}, and a joint feature extraction model is established. Next, equations (2), (3), and (4) are employed to capture the feature representation of training items. Then, GP perturbation is added to the feature representation. Finally, a robustness projector is trained under the loss function, as shown in equation (6).

Step 2: The trained encoder in Step 1 is used to extract the training sample feature vectors. Besides, the cross-entropy function with the Adam optimizer is adopted to train the classifier, and the parameters are updated according to the results.

Step 3: The trained encoder and the classifier of the PCRA-GP model are applied to predict the label of testing data T_{test}. The comparison between the predicted value and the true value reveals the model classification accuracy.

3.1. CNN Layer. CNNs are widely used for image tasks and are gradually being applied to NLP. They can extract local correlation of spatial or temporal structures and can also be used to capture sequence information and reduce the number of input dimensions. TextCNN [43] added a convolutional layer after the word vector to build a simple model and achieved good sentence classification results. The CNN layer used is the same as TextCNN. The structure of TextCNN is shown in Figure 2.

The input text of the CNN is denoted as $T = (x_1, x_2, \ldots, x_n)$, $T \in R^{n\times d}$. The text length is n, and each word is represented as a vector x_n. After the embedding layer, the word is represented as a vector of the dimension d. Because the input text matrix must be of the same length, we cut off lengths greater than n and pad those lengths less than n. For the selection of n-gram features, the height of convolution kernels is set to 3, 4, and 5. The step size is set to 1. After one-dimensional convolutional operation, the m^{th} local features encoding f_m are calculated using equation (1). Regarding unilateral suppression operation, the ReLU function is used to change negative values to zero, all positive values and zero remain unchanged, and a pooling layer is adopted to retain the main features while reducing parameters and computation. Finally, the feature sequence is derived as $L_f = [f_1, f_2, \ldots, f_m]$. The output of the three-layer CNN is concatenated as the downstream input (as given by equation (2)):

$$f_m = r(W_t \bullet x_t + b), \tag{1}$$

$$\begin{aligned}\mathrm{F} = [F_1, F_2, \ldots, F_m] &= L_f^{(3)} \oplus L_f^{(4)} \oplus L_f^{(5)}\\ &= \left[f_1^{(3)} \oplus f_1^{(4)} \oplus f_1^{(5)}, \ldots, f_m^{(3)} \oplus f_m^{(4)} \oplus f_m^{(5)}\right],\end{aligned} \tag{2}$$

where $r(\bullet)$ is the ReLU function, W_t and b represent the filter weight and bias term in a window of the word x_t, m denotes the total number of filters, F represents the output through the CNN layer, $\bullet^{(\bullet)}$ represents the feature under different kernel sizes, and $\oplus$ represents the concatenate operator. The length of F is equal to m.

3.2. Bi-GRU Layer. RNNs have a recurrent hidden state throughout the computational flow which can maintain continuous sequence information and has a good effect on processing sequence data. Therefore, in addition to text local features, we also pay attention to sequence global information. A GRU is a type of RNN architecture and has become the popular structure. It performs in a manner similar to long short-term memory and is less computationally onerous. In our architecture, BiGRU is used to encode the forward and backward information of the sentence. The structure of the BiGRU is shown in Figure 3.

The forward GRU ($\overrightarrow{\text{GRU}}$) calculates each timestep hidden state from 1 to m. In the backward GRU ($\overleftarrow{\text{GRU}}$) layer, the reverse calculation is performed from m to 1. Finally, the bidirectional output is merged as the result of BiGRU. The whole calculation process can be described using the following formulae:

$$\begin{aligned} Y_m &= \overrightarrow{\text{GRU}}(F) = f(w_1 F_m + w_2 Y_{m-1}),\\ Y'_m &= \overleftarrow{\text{GRU}}(F) = f(w_3 F_m + w_4 Y'_{m-1}),\\ o_m &= \text{BiGRU}(w_5 Y_m + w_6 Y'_m),\end{aligned} \tag{3}$$

where Y_m and Y'_m are the forward and backward output, o_m represents the output results of BiGRU at time m, the final output of the BiGRU network is $O = [o_1, o_2, \ldots, o_m]$, and $w_{(\bullet)}$ denotes the weights corresponding to the forward and reverse hidden states.

3.3. Attention Layer. An attention mechanism is a feature selection method, which focuses on the strength and weakness of local information and obtains dependent information for correct judgment. In practice, the attention function is commonly divided into two categories: dot-product attention and additive attention [44].

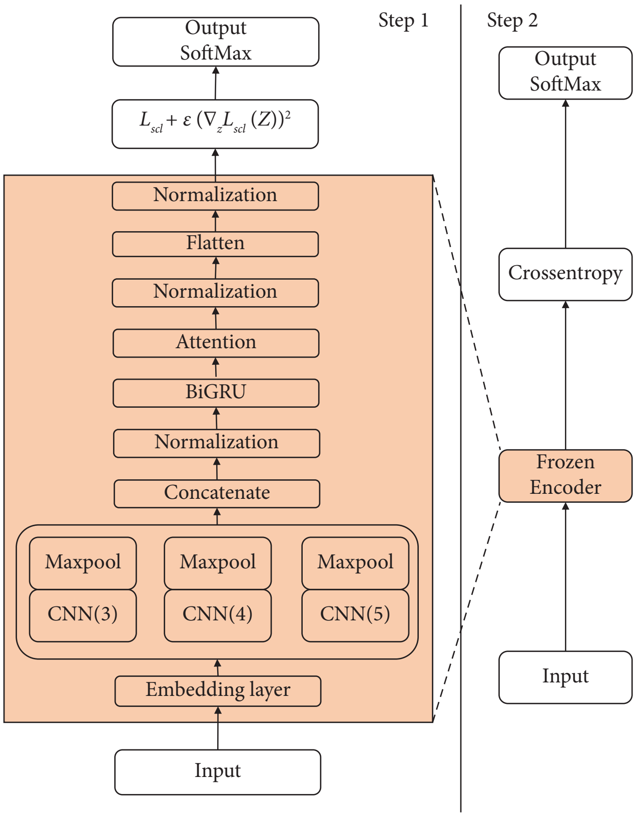


FIGURE 1: The framework of the PCRA-GP model.

```
Input: a set of text $T$
Output: Accuracy
    #Step 1: training an encoder()
    $T_{\text{train}} \in T$;
    for $t \in T_{\text{train}}$ do
        $F \longleftarrow$ concat $(\text{CNN}(t, 3), \text{CNN}(t, 4), \text{CNN}(t, 5))$;
        $O \longleftarrow \text{BiGRU}(F)$;
        $c \longleftarrow \text{Att}(O)$;
        $z \longleftarrow \text{projector}(c)$;
        $\text{loss}_{\text{train}} \longleftarrow$ equation (6);
        update $\theta_{train\_e}$ to minimize $\text{loss}_{\text{train\_}e}$;
    #Step 2: training a model PCRA-GP() with a frozen encoder encoder()
    for $t \in T_{\text{train}}$ do
        $z \longleftarrow \text{encoder}(t)$;
        $\hat{y} \longleftarrow \text{classifier}(z)$
        $\text{loss}_{\text{train\_}c} \longleftarrow \text{crossentropy}(y, \hat{y})$;
        update $\theta_{\text{train\_}c}$ to minimize $\text{loss}_{\text{train\_}c}$;
    #Step 3: testing the PCRA-GP() model
    $T_{\text{test}} \in T$;
    for $t \in T_{\text{test}}$ do
    return accuracy $\longleftarrow$ PCRA-GP (t). evaluate();
```

ALGORITHM 1: Pseudocode for PCRA-GP.

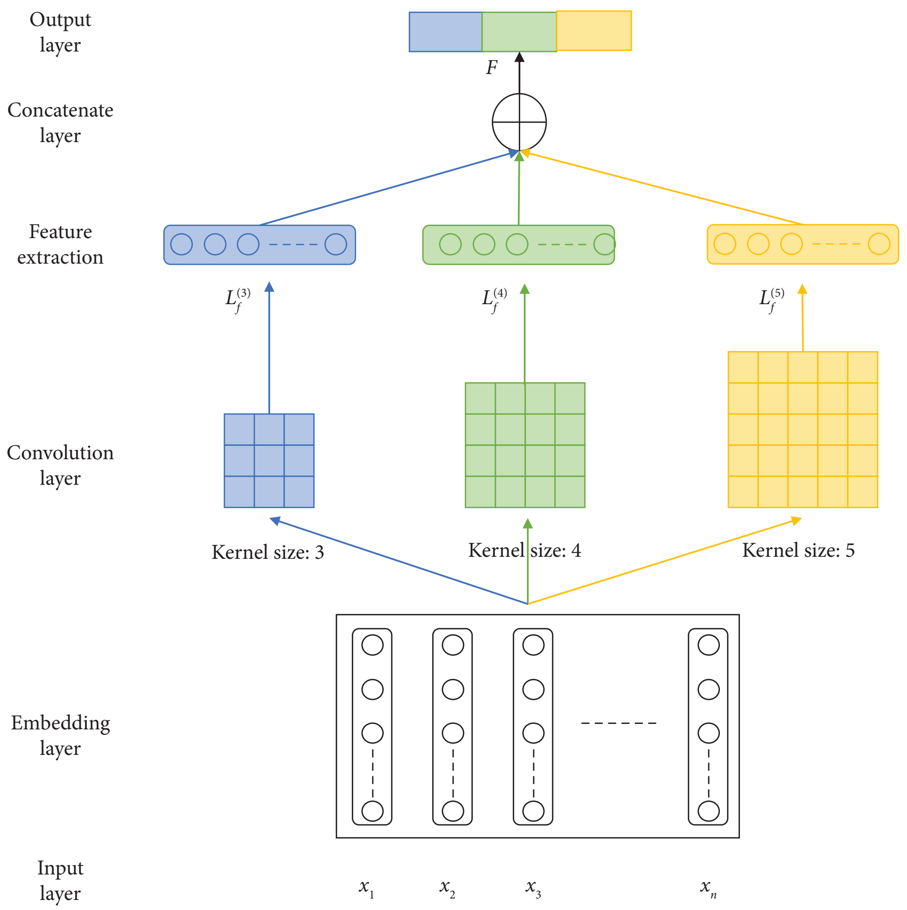


Figure 2: Structural diagram of the TextCNN process.

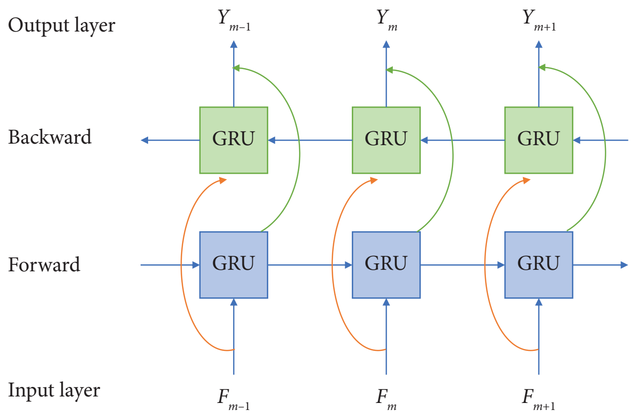


Figure 3: Structural diagram of the BiGRU.

Dot-product attention is much faster and more space-efficient and uses a highly optimized matrix multiplication code, while additive attention is committed to dealing with long-term memory problems and applied to longer and wide varying sequences. The schematic of the attention mechanism is illustrated in Figure 4.

The attention mechanism can be simplified as follows:

$$\begin{aligned} e_t &= a(O), \\ \alpha_t &= \frac{\exp\ (e_t)}{\sum_{k=1}^{m} \exp(e_k)}, \\ c &= \sum_{t=1}^{m} \alpha_t o_t, \end{aligned} \tag{4}$$

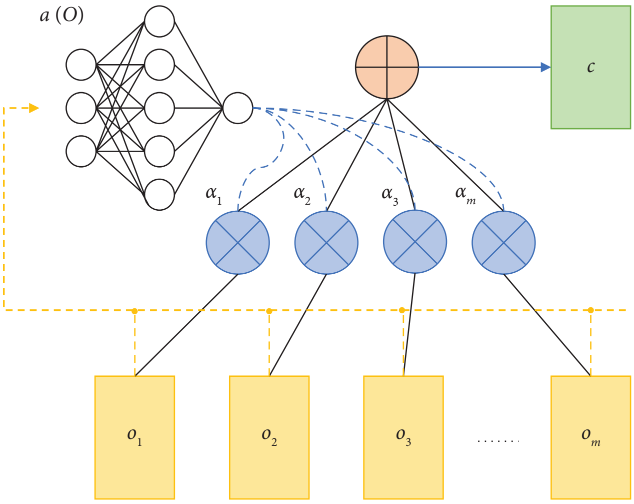


Figure 4: Schematic of the attention mechanism.

where $a(\bullet)$ is a learnable function dependent only on O, c denotes the attention weight embedding sequence that matches the product of the input hidden state o and the weight α, and the parameters t and k indicate different positions.

3.4. SCL Method. SCL adopts a fully supervised method to further reveal label information. The SCL method [1] comprises three parts: data augmentation, an encoder network, and a projection network. In our work, the feature extraction network consists of a CNN layer (Section 3.1), a RNN layer (Section 3.2), and attention (Section 3.3) components. Besides, we do not directly use the data augmentation part in our model but indirectly adopt the embedding penalty gradient (Section 3.5). In this section, the project network and supervised contrastive loss function are introduced.

The projection network is instantiated as a dense layer. First, this layer maps the encoder network output c to a feature vector $z = project(c) \in R^{D_p}$, where $D_p = 128$ and z is normalized to place the output on the unit hypersphere; then, an inner product is used to measure the output in the projection space.

Khosla et al. [1] argued that the SCL loss is equivalent to the N-pair loss when more than one negative item is present. The N-pair loss [45] is a metric-learning objective, which accelerates convergence through strengthening the interaction between positive classes and negative classes in each update. The contrastive loss is shown as follows:

$$L_{\text{scl}} = \frac{1}{N}\sum_{i=1}^{N}\log\left(1 + \sum_{j\neq i}\exp\left(\frac{z_i^T z_j^+ - z_i^T z_i^+}{\tau}\right)\right), \quad (5)$$

where N is the total number of sample data categories and $i, j \in N$, $(\bullet)^T$ denotes the transpose operation, z_i^+ represents the positive example representation while z_j^+, $i \neq j$ denotes the negative example representation, and τ is the temperature parameter.

3.5. Gradient Penalty. Commonly, text classification mainly relies on text representation and feature extraction to obtain better classification results. However, the model trained by this method is vulnerable to attacks, which leads to the model having poor robustness. In the field of NLP, data augmentation is a way to improve the robustness of the model, but it has its own limitations. Text enhancement methods are usually suitable for the character level, word level, and sentence level and are rarely applied to document-level materials. The GP method adds perturbation to the loss function, realizing a universal adversarial method. As a regularization method, the gradient penalty reduces overfitting and improves model generalizability.

In our method, the disturbance part is calculated by the differential of the contrastive loss L_{scl} to the feature vector z, defined as $\nabla_z L_{\text{scl}}(z)$. A hyperparameter ε is also introduced to control the extent of the penalty applied. Then, loss function equation (5) is updated to equation (6):

$$L_{\text{total}} = L_{\text{scl}} + \varepsilon\left(\nabla_z L_{\text{scl}}(z)\right)^2. \quad (6)$$

In PCRA-GP, the pooling layer, dropout layer, and normalizing layer are used to avoid overfitting. The softmax layer is used to generate the probability distribution of the text label class, and the Adam optimizer is selected to optimize the loss function of the network.

4. Experiments

In this section, the experimental set-up and comparative experimental design are demonstrated. The experimental setup is about the presence of datasets and the initial settings of model parameters used in the experiment. In the comparative test part, the results are compared with the deep learning baselines and the state-of-the-art text classification methods. Finally, the model classification accuracy changes under the two main components of the model, and hyperparameter settings are evaluated. The detailed experimental data are recorded in Appendix A (Tables 1–8). In addition, as a verification of the versatility of our idea, the GP method is also used in image classification (Appendix B) (Figure 5).

4.1. Experimental Setup

4.1.1. Datasets. Our model is used to solve bilingual text classification tasks (including sentiment and news category) at the chapter level and paragraph level in English and Chinese (Table 9). The datasets used in this article are as follows:

(i) IMDB: it is a dataset of 50,000 movie reviews in English, labeled by sentiment (positive/negative).

(ii) ChnSentiCorp-Htl (https://github.com/SophonPlus/ChineseNlpCorpus): it is a corpus including more than 7,000 hotel reviews in Chinese.

(iii) 20news: The 20 Newsgroups (https://qwone.com/~jason/20Newsgroups/) dataset is a collection of approximately 20,000 English newsgroup

Table 1: IMDB with various epsilon (temperature=0.05) on RTX2080.

IMDB with various epsilon (temperature=0.05) on RTX2080									
epoch	batch_size	norm	epsilon						PCRA
			0.01	0.02	0.031	0.04	0.05	0.06	
200	256	L	89.34%	89.38%	89.38%	89.26%	89.75%	89.55%	89.52%
		B	89.22%	88.51%	89.36%	78.07%	74.67%	52.30%	88.61%
	128	L	89.47%	89.76%	89.44%	49.47%	87.58%	49.47%	89.38%
		B	89.38%	89.13%	88.95%	88.77%	88.88%	87.37%	89.16%
	64	L	88.75%	49.47%	49.47%	50.53%	49.47%	49.47%	89.21%
		B	89.42%	89.37%	89.04%	89.18%	52.03%	50.73%	89.00%
100	256	L	89.69%	89.64%	50.80%	89.71%	89.24%	88.90%	89.72%
		B	88.78%	88.67%	89.06%	88.94%	89.23%	86.89%	88.89%
	128	L	88.93%	49.47%	89.91%	87.60%	50.10%	49.47%	89.46%
		B	89.19%	88.47%	89.43%	87.81%	82.84%	89.12%	86.48%
	64	L	89.15%	89.24%	50.53%	49.47%	49.47%	49.47%	89.24%
		B	89.29%	88.76%	88.92%	89.30%	50.32%	49.46%	89.05%

Bold values are the better performances with different epsilon. Values with color red are the best accuracy scores.

Table 2: IMDB with various epsilon (temperature=0.05) on K80.

IMDB with various epsilon (temperature=0.05) on K80									
epoch	batch_size	norm	epsilon						PCRA
			0.01	0.02	0.031	0.04	0.05	0.06	
200	256	L	90.18%	89.83%	50.83%	50.56%	50.48%	50.80%	89.82%
		B	89.94%	89.86%	90.23%	88.81%	89.47%	88.59%	90.26%
	128	L	89.97%	50.69%	90.16%	89.94%	89.60%	49.47%	89.40%
		B	89.90%	90.34%	89.23%	89.86%	90.18%	52.20%	90.23%
	64	L	88.85%	49.47%	49.47%	50.53%	49.47%	49.47%	89.03%
		B	89.80%	89.35%	55.62%	50.82%	50.64%	49.50%	89.39%
100	256	L	89.89%	51.09%	50.71%	50.86%	50.80%	50.78%	89.94%
		B	89.23%	51.59%	89.51%	50.78%	90.13%	50.80%	88.08%
	128	L	89.68%	50.57%	50.77%	49.47%	49.47%	50.53%	89.66%
		B	90.31%	89.71%	90.09%	90.01%	90.76%	51.04%	90.14%
	64	L	89.90%	90.24%	49.47%	49.47%	49.47%	49.47%	89.78%
		B	89.70%	89.54%	52.95%	50.89%	50.47%	51.11%	89.57%

Bold values are the better performances with different epsilon. Values with color red are the best accuracy scores.

documents, partitioned evenly across 20 different newsgroups. It has become a popular dataset for experiments in text applications of machine learning techniques, such as text classification and text clustering.

(iv) Cnews: It is a subset of the THUCnews (https://github.com/thunlp/THUCTC) dataset produced by NLP and Computational Social Science Lab, Tsinghua University. The THUCnews dataset mainly collects Sohu news, and Cnews contains 10 categories and 65,000 Chinese texts.

4.1.2. Baseline Methods. BALB: it is a backdoor attack against the LSTM-based text classification system [46].

GCNN: it is a model that introduces contextual information obtained by the RNN and local information captured by the CNN and was proposed by Huang and Liu [31].

AEG: it is a reinforcement-learning-based approach to generating adversarial examples in black-box settings [21].

AC-BiLSTM: it is a novel and unified architecture that contains the bidirectional LSTM (BiLSTM), attention mechanism, and convolutional layer [34].

BiGRU-CNN: A model composed of GloVe, CNN, RNN and max-pooling layer was proposed by Abid et al. [28].

BAM: it is a bilevel attention model that does not rely on external knowledge and was proposed by Liu et al. [42].

Text GCN: in this model, the first task is to build the whole corpus into a heterogeneous graph and use GNN to

Table 3: ChnsentiCorp-Htl with various epsilon (temperature=0.05) on RTX2080.

ChnsentiCorp-Htl with various epsilon (temperature=0.05) on RTX2080									
epoch	batch_size	norm	epsilon						PCRA
			0.01	0.02	0.031	0.04	0.05	0.06	
200	256	L	88.42%	90.67%	90.08%	90.17%	89.42%	55.33%	91.08%
		B	89.92%	89.75%	89.50%	90.75%	89.67%	90.08%	89.67%
	128	L	89.50%	89.25%	89.17%	89.17%	89.75%	56.42%	90.25%
		B	90.33%	89.58%	90.08%	90.00%	89.00%	89.08%	90.58%
	64	L	89.00%	88.75%	49.42%	57.25%	49.42%	49.42%	88.92%
		B	91.42%	89.00%	90.08%	88.75%	90.17%	56.92%	89.92%
100	256	L	90.25%	90.17%	90.83%	72.25%	58.25%	56.42%	90.50%
		B	90.00%	89.17%	90.42%	90.83%	89.50%	89.75%	90.83%
	128	L	89.42%	90.58%	56.67%	56.75%	56.58%	56.75%	90.08%
		B	89.67%	89.83%	90.17%	89.25%	90.17%	91.42%	90.25%
	64	L	89.42%	88.50%	56.08%	57.08%	61.42%	57.50%	91.00%
		B	89.58%	89.75%	89.83%	57.00%	89.67%	56.17%	89.83%

Bold values are the better performances with different epsilon. Values with color red are the best accuracy scores among PCRA-GP and PCRA.

Table 4: ChnsentiCorp-Htl with various epsilon (temperature=0.05) on K80.

ChnsentiCorp-Htl with various epsilon (temperature=0.05) on K80									
epoch	batch_size	norm	epsilon						PCRA
			0.01	0.02	0.031	0.04	0.05	0.06	
200	256	L	90.00%	89.17%	90.42%	90.25%	91.08%	57.33%	91.75%
		B	90.42%	90.58%	89.75%	89.17%	90.92%	89.58%	89.42%
	128	L	89.67%	90.08%	89.33%	91.42%	59.08%	59.08%	90.92%
		B	89.42%	88.42%	89.67%	89.83%	90.42%	90.58%	89.33%
	64	L	91.25%	89.17%	48.92%	57.50%	48.92%	48.92%	90.17%
		B	90.42%	90.42%	88.58%	90.00%	57.92%	56.42%	90.25%
100	256	L	89.25%	89.58%	77.25%	58.83%	59.83%	57.58%	90.42%
		B	91.67%	89.50%	91.25%	88.58%	90.67%	90.33%	89.25%
	128	L	91.08%	90.33%	58.83%	58.25%	59.08%	58.17%	90.75%
		B	87.92%	89.92%	91.00%	90.25%	91.00%	88.92%	89.00%
	64	L	91.75%	89.67%	58.33%	58.67%	57.50%	59.83%	88.92%
		B	90.83%	90.42%	90.25%	90.33%	90.58%	55.50%	89.25%

Bold values are the better performances with different epsilon. Values with color red are the best accuracy scores among PCRA-GP and PCRA.

jointly learn word and document embeddings, as performed by Yao et al. [30].

SGC: it is a method aimed to simplify single linear transformation and was proposed by Wu et al. [35].

SSGC: this model is a variant model derived from Markov diffusion kernels, deals with degradation caused by the increased depth, and was proposed by Zhu and Koniusz [37].

C-CNN: this model can realize centrality convolution based on a separate encoding function and was proposed by Dong et al. [26].

BertGCN: it is a model that combines Bert's large-scale pretraining knowledge and the transferability of the GCN and was proposed by Lin et al. [36].

RoBERTaGCN: this method entails joint RoBERTa and GCN as a comparative experiment and was proposed by Lin et al. [36].

SWEM: this method uses a modified hierarchical pooling strategy in few-shot transfer learning and was proposed by Pan et al. [41].

C-BiGRU-ATT: this model uses the CNN, attention mechanism, and BiGRU, extracting contextual and local information at the character and vocabulary levels, and was proposed by Wenzhen et al. [33].

ATCNN-3: this method extracts and filters features by adding attention mechanisms in different positions and was proposed by Tang and Yang. [32].

Table 5: 20news with various epsilon (temperature=0.05) on RTX2080.

20news with various epsilon (temperature=0.05) on RTX2080									
epoch	batch_size	norm	epsilon						PCRA
			0.01	0.02	0.031	0.04	0.05	0.06	
200	256	L	87.65%	84.07%	88.93%	88.87%	11.92%	10.89%	87.49%
		B	87.52%	86.67%	81.65%	86.43%	86.22%	83.30%	88.48%
	128	L	88.79%	87.44%	12.80%	14.21%	12.13%	12.80%	87.87%
		B	86.94%	86.22%	87.33%	84.63%	78.23%	71.93%	87.31%
	64	L	87.87%	12.06%	13.99%	13.91%	12.98%	15.53%	86.80%
		B	86.67%	85.82%	78.57%	85.98%	82.77%	15.03%	86.43%
100	256	L	88.93%	89.17%	88.40%	87.60%	11.87%	11.15%	88.40%
		B	86.62%	85.90%	80.48%	78.20%	72.09%	69.25%	88.10%
	128	L	87.97%	73.58%	12.64%	12.27%	10.62%	13.12%	87.33%
		B	84.71%	82.95%	78.60%	86.72%	62.00%	62.27%	86.72%
	64	L	87.89%	13.54%	13.20%	13.14%	12.80%	12.56%	87.63%
		B	86.80%	86.51%	78.92%	61.13%	12.51%	13.78%	86.78%

Bold values are the better performances with different epsilon. Values with color red are the best accuracy scores among PCRA-GP and PCRA.

Table 6: 20news with various epsilon (temperature=0.05) on K80.

20news with various epsilon (temperature=0.05) on K80									
epoch	batch_size	norm	epsilon						PCRA
			0.01	0.02	0.031	0.04	0.05	0.06	
200	256	L	88.48%	88.53%	88.08%	89.86%	89.09%	88.79%	88.42%
		B	87.39%	87.84%	87.92%	77.83%	85.90%	86.86%	88.42%
	128	L	83.59%	86.83%	13.09%	13.78%	14.07%	14.74%	86.99%
		B	87.31%	86.19%	87.44%	80.91%	87.36%	86.17%	87.55%
	64	L	88.02%	12.80%	13.73%	12.77%	13.86%	12.03%	86.59%
		B	86.51%	86.70%	86.48%	13.70%	85.34%	86.25%	86.25%
100	256	L	86.99%	89.27%	88.37%	87.20%	83.64%	88.58%	88.26%
		B	87.04%	87.41%	87.92%	68.61%	85.58%	87.28%	87.87%
	128	L	88.82%	11.82%	11.74%	12.11%	10.46%	10.73%	88.02%
		B	87.52%	86.35%	82.90%	78.07%	84.94%	85.10%	88.37%
	64	L	87.12%	12.37%	12.83%	12.13%	12.16%	14.87%	87.68%
		B	86.99%	82.45%	76.10%	82.10%	79.05%	77.80%	86.38%

Bold values are the better performances with different epsilon. Values with color red are the best accuracy scores among PCRA-GP and PCRA.

4.1.3. Parameter Settings. The PCRA-GP model uses accuracy as the evaluation metric to estimate the overall classification performance. The first layer of PCRA-GP is an embedding layer which converts words into 300-dimensional vectors. The kernels of multichannel CNNs are set to 3, 4, and 5. The number of hidden units of BiGRUs and CNNs is set to 256 which is suitable for document classification and 128 which is suitable for paragraph classification. The training batch sizes are set to 256, 128, and 64. The dropout rate is 0.5. To better avoid model overfitting, we add dropout behind the embedding layer and before the classification dense layer and set it to 0.2 when solving IMDB and ChnSentiCorp-Htl models. The two hyperparameters (epsilon and temperature) are set to 0.01–0.06. The backpropagation algorithm based on the Adam stochastic optimization method is used to train the network through time, and the learning rates are 0.001 and 0.0001, which are used for different datasets. The normalization layer involves batch normalization and layer normalization. Each corpus is split into training and testing sets in the proportion of 8 : 2. All experiments are conducted with Python 3.7.9 and Keras 2.4.3 on GeForce RTX 2080 Ti and NVIDIA Tesla K80 systems. The specific parameters of the model are listed in Table 10.

Table 7: Cnews with various epsilon (temperature=0.05) on RTX2080.

Cnews with various epsilon (temperature=0.05) on RTX2080									
epoch	batch_size	norm	epsilon						PCRA
			0.01	0.02	0.031	0.04	0.05	0.06	
200	256	L	96.79%	96.99%	96.92%	97.08%	97.08%	93.59%	97.18%
		B	93.48%	97.01%	97.14%	96.97%	97.02%	97.13%	97.08%
	128	L	96.57%	97.09%	96.82%	57.18%	93.15%	62.88%	96.60%
		B	96.62%	96.92%	93.56%	67.17%	64.64%	71.28%	93.64%
	64	L	96.93%	56.23%	97.05%	53.88%	45.96%	97.07%	96.52%
		B	97.02%	93.20%	57.51%	57.96%	65.53%	57.29%	92.80%
100	256	L	96.80%	96.88%	97.01%	97.18%	97.09%	72.39%	97.18%
		B	93.32%	97.05%	96.95%	97.10%	97.09%	97.04%	93.60%
	128	L	97.13%	90.56%	96.99%	50.32%	50.53%	96.95%	97.16%
		B	96.90%	97.11%	93.76%	71.48%	66.04%	78.09%	92.99%
	64	L	97.15%	55.15%	58.15%	46.34%	53.30%	48.79%	96.54%
		B	96.94%	97.21%	97.07%	96.97%	56.91%	38.40%	92.93%

Bold values are the better performances with different epsilon. Values with color red are the best accuracy scores among PCRA-GP and PCRA.

Table 8: Cnews with various epsilon (temperature=0.05) on K80.

Cnews with various epsilon (temperature=0.05) on K80									
epoch	batch_size	norm	epsilon						PCRA
			0.01	0.02	0.031	0.04	0.05	0.06	
200	256	L	97.01%	96.99%	97.37%	97.12%	97.15%	63.26%	96.78%
		B	96.84%	97.15%	96.98%	97.19%	97.15%	97.15%	92.85%
	128	L	97.12%	97.25%	58.61%	67.05%	96.70%	59.00%	96.88%
		B	97.18%	96.85%	97.37%	96.87%	71.18%	66.63%	96.86%
	64	L	96.91%	61.41%	59.38%	55.95%	70.95%	97.22%	96.69%
		B	97.01%	97.18%	69.25%	97.03%	71.02%	71.55%	96.57%
100	256	L	96.88%	96.78%	97.12%	97.22%	97.25%	67.02%	96.90%
		B	97.26%	97.22%	97.45%	96.98%	97.17%	97.19%	93.14%
	128	L	97.04%	96.89%	50.22%	60.03%	60.78%	95.42%	96.92%
		B	96.98%	97.08%	97.42%	87.72%	67.95%	65.45%	93.82%
	64	L	97.15%	60.97%	58.25%	53.68%	71.65%	56.35%	96.65%
		B	97.02%	96.91%	70.54%	65.58%	97.06%	69.87%	96.35%

Bold values are the better performances with different epsilon. Values with color red are the best accuracy scores among PCRA-GP and PCRA.

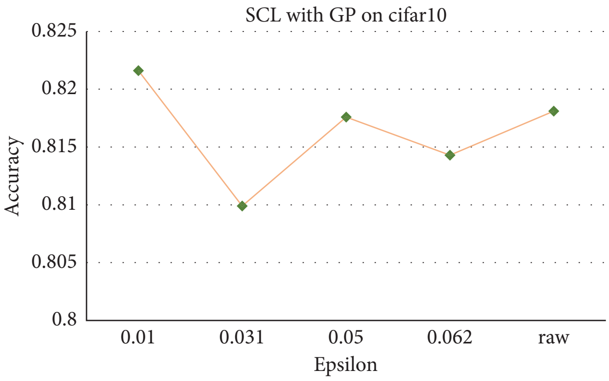


Figure 5: Result of SCL with GP on the cifar10 dataset.

Table 9: Dataset statistics.

Corpus	Classes	Instance	Avg length	Train/test	Language
IMDB	2	50,000	231	25,000/25,000	English
ChnSentiCorp-Htl	2	7767	128	3000/3000	Chinese
20news	20	18,846	221.26	11,314/7532	English
Cnews	10	65,000	530	50,000/15,000	Chinese

Table 10: The specific parameters of PCRA-GP.

Component	Parameter
Word embedding	300
CNN	[3, 4, 5], [256, 128]
BiGRU	[256, 128]
Batch size	[256, 128, 64, 32]
Dropout	[0.2, 0.5]
Dense	128
Epsilon	[0.01, 0.02, 0.031, 0.04, 0.05, 0.06]
Temperature	[0.01, 0.02, 0.03, 0.04, 0.05, 0.06]
Learning rate	[0.001, 0.0001]
Normalization	Batch normalization, layer normalization
Python	3.7.9
Keras	2.4.3
GPU	GeForce RTX 2080 Ti, NVIDIA Tesla K80

4.2. Experiments

4.2.1. Overall Comparison. In this section, all the experimental data are summarized and the best classification accuracy is selected for comparisons with the baseline model accuracy. The results recorded in Table 11 involve the classification of sentiment and news texts, and the text languages are Chinese and English. The baseline results are all reported in the cited papers.

The results for the IMDB dataset show that the PCRA is higher than GCNN by 0.26%, while the PCRA-GP is 0.34% higher. In comparison with backdoor attack models, our model classification accuracy improvement is significantly ahead of theirs by 5.42% and 5.34%. The black-box setting AEG model is better than backdoor attack models, but it is weaker than our model by 0.39% and 0.31%. The results indicate that perturbation on the model loss function is an effective way to improve accuracy.

The models applied to the ChnsentiCorp-Htl dataset show that BiGRU-CNN lacks an attention layer compared to AC-BiLSTM, and AC-BiLSTM lacks projection compared to the PCRA model. The model using only bilevel attention returns the lowest result. Due to the small amount of data, the simple model BiGRU-CNN is better than the complex point model AC-BiLSTM, and the addition of the GP method results in the highest accuracy of classification. The results for the Cnews dataset indicate that PCRA and PCRA-GP are, respectively, 1.37% and 1.9% better than C-BiGRU-ATT that does not use a projector and GP components. Compared with ATCNN-3, PCRA is 0.34% higher after increasing feature extraction ability. Compared with SWEM, which is a few-shot model, the overall training method can maintain a high level of expressiveness. The results show that a projector under the SCL training process brings improved results with the GP constraint.

Since TextGCN was proposed, GNN has demonstrated high-quality performance and gradually replaced the traditional deep model based on CNNs and RNNs. The results for the 20 Newsgroups dataset show that the classification accuracies of GNN-based models (TextGCN, SGC, and SSGC) are gradually improving. After fusing the pretrained Bert model, the performance of the GNN model is further improved. Our proposed model is 0.36% better than RoBERTaGCN and achieved SOTA results. The results indicate that our model improves the performance of traditional deep models based on the CNN, RNN, and attention mechanism.

Combined with the performance of our proposed model for four public datasets, our results are found to be consistently better than the baseline results recorded in most published papers. In view of the above discussion, the PCRA-GP model that we proposed is highly competitive.

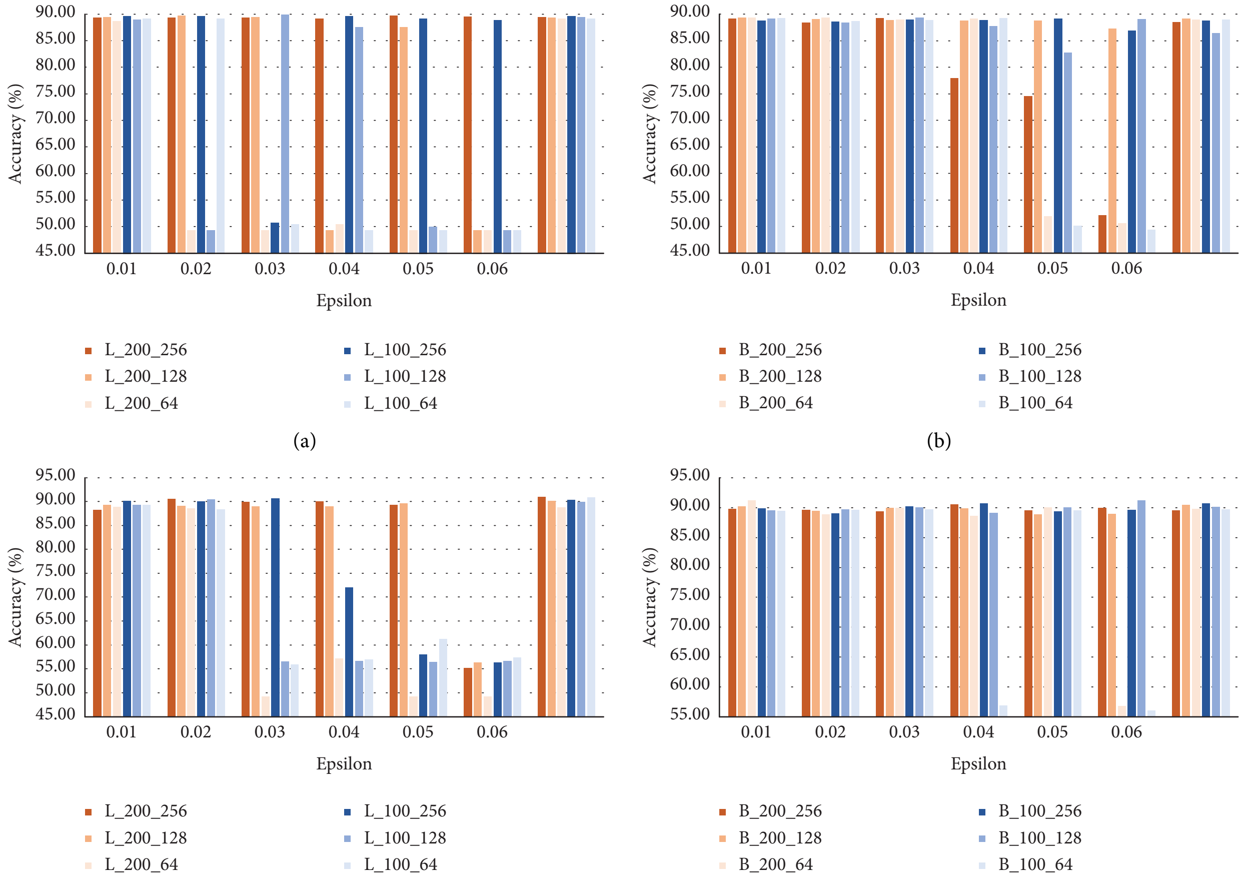
4.2.2. Effect of Components of PCRA-GP. In Figure 6, the classification accuracies are plotted for all datasets, in which the temperature hyperparameter is set to 0.05 and comes from the experiment using the RTX 2080Ti system. Since the experimental results are sensitive to parameter settings, our experimental process mainly focuses on the type of normalization, batch size, epoch, and epsilon.

Overall, as shown in Figure 6, the performance of the PCRA model is relatively stable and less affected. Batch normalization could withstand larger epsilon disturbances than layer normalization. When the batch size is set to 256, the model classification accuracy can be maintained in response to variations in other factors. The various epoch

Table 11: Accuracy of the models.

Dataset	Models	Accuracy (%)
IMDB	BALB [46]	84.92
	GCNN [31]	90.0
	AEG [21]	89.95
	PCRA (ours)	90.26
	PCRA-GP (ours)	**90.34**
ChnSentiCorp-Htl	AC-BiLSTM [34]	90.60
	BiGRU-CNN [28]	91.60
	BAM [42]	90.0
	PCRA (ours)	**91.75**
	PCRA-GP (ours)	**91.75**
20news	TextGCN [30]	86.34
	SGC [35]	88.50
	SSGC [37]	88.60
	C-CNN [26]	82.59
	BertGCN [36]	89.3
	RoBERTaGCN [36]	89.5
	PCRA (ours)	88.42
	PCRA-GP (ours)	**89.86**
Cnews	SWEMs [41]	90.07
	C-BiGRU-ATT [33]	95.55
	ATCNN-3 [32]	96.58
	PCRA (ours)	96.92
	PCRA-GP (ours)	**97.45**

Bold values are the best accuracy scores.



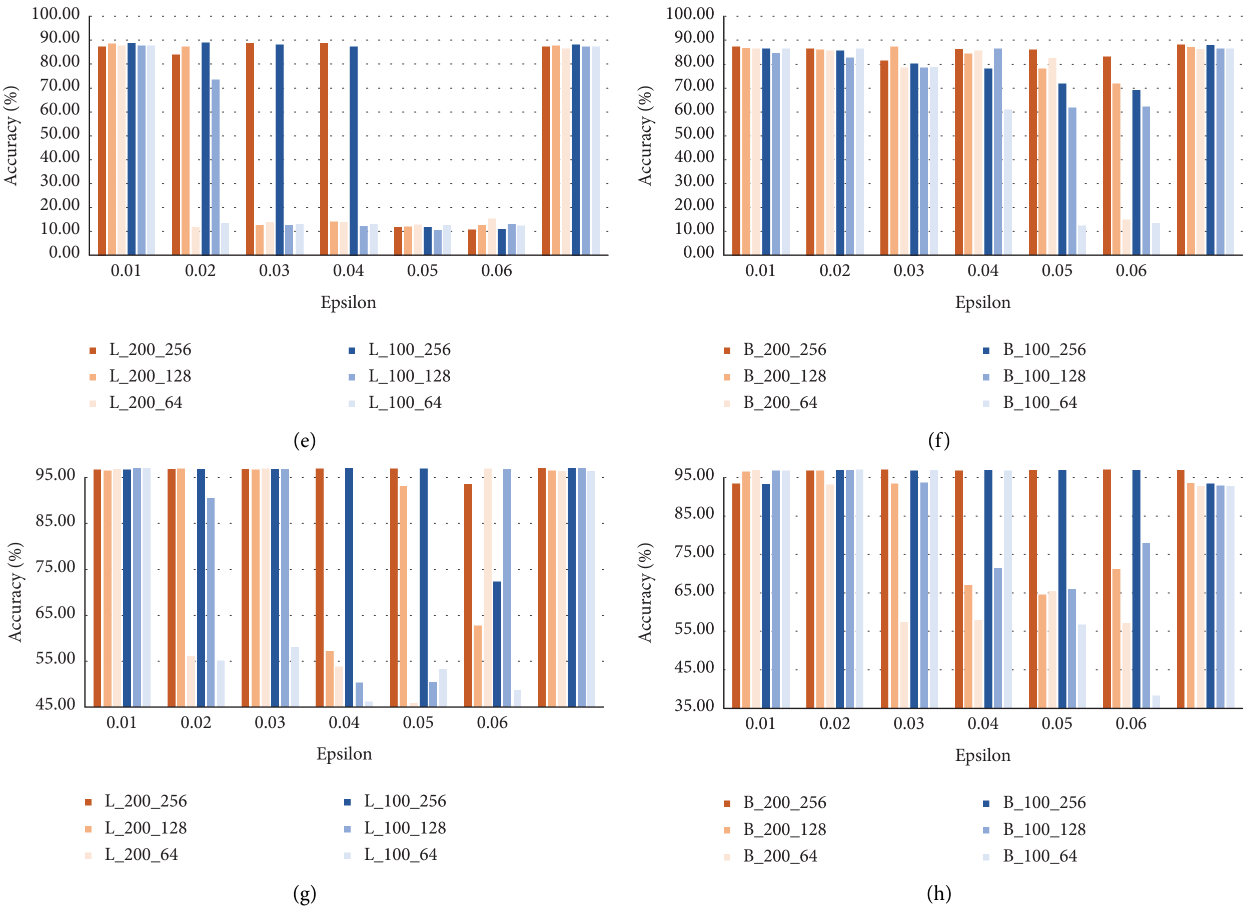
(a)

(b)

(c)

(d)

Figure 6: Continued.



(e)

(f)

(g)

(h)

Figure 6: The accuracy of datasets with the type of normalizations, epoch, and batch size on RTX2080 Ti. The symbol *L*/*B*_XXX_YYY means layer (*L*) and batch (*B*) normalizations, epsilon, epoch, and batch size. The PCRA column represents contrastive learning without the gradient penalty. (a) The PCRA-GP's accuracy on IMDB, trained with *L*. (b) The PCRA-GP's accuracy on IMDB, trained with *B*. (c) The PCRA-GP's accuracy on ChnSentiCorp-Htl, trained with *L*. (d) The PCRA-GP's accuracy on ChnSentiCorp-Htl, trained with *B*. (e) The PCRA-GP's accuracy on 20news, trained with *L*. (f) The PCRA-GP's accuracy on 20news, trained with *B*. (g) The PCRA-GP's accuracy on Cnews, trained with *L*. (h) The PCRA-GP's accuracy on Cnews, trained with *B*.

settings have little effect on the performance. When epsilon is greater than 0.04, the classification accuracy decreases significantly.

In Table 12, the maximum value is counted under different parameter settings, which ignores epsilon, and the data are selected from Figure 6. Different from the trend analysis of the data, we pay more attention to specific accuracy value information. By summarizing the classification results of all datasets, the layer normalization classification results are found to be 33 times higher than those of batch normalization among 48 groups, and the best accuracy occurrence ratio is 5 : 3. Moreover, eight locations, where the minimum value is obtained, are counted, five of them appear at a batch size of 64, and the epoch is 200. The maximum value appears 5 times when the batch size is 256.

In summary, batch normalization methods are stable, while layer normalization has high accuracy and often leads to the optimal classification. A large batch size can acquire good accuracy and offset the impact of different epochs. In terms of overall results, the result is best balanced when epsilon is 0.01, and the second best choice is 0.02.

4.2.3. Ablation Experiment. We tested the performance of different combinations, and the results are shown in Table 13. The experimental hyperparameters are fixed and executed on RTX 2080Ti. We set the epoch size to 100, batch size to 256, and temperature to 0.05. Besides, epsilon is set to 0.01 during adversarial training. Considering PC as the baseline, the introduction of *R* reduces the classification effect and the integration of A brings a huge improvement in the results. The adversarial training with GP also improves the classification accuracy of the model. 20news is not well trained in methods PC and PCR. Additionally, we visualize the loss value and accuracy during the training process to understand the results. The training process data are shown in Figure 7. During the training stage of the encoder and classifier, the loss values of PC and PCR do not drop significantly, resulting in low accuracy. However, metrics for PCRA and PCRA-GP perform well. In addition, during the training of the encoder, the introduction of GP

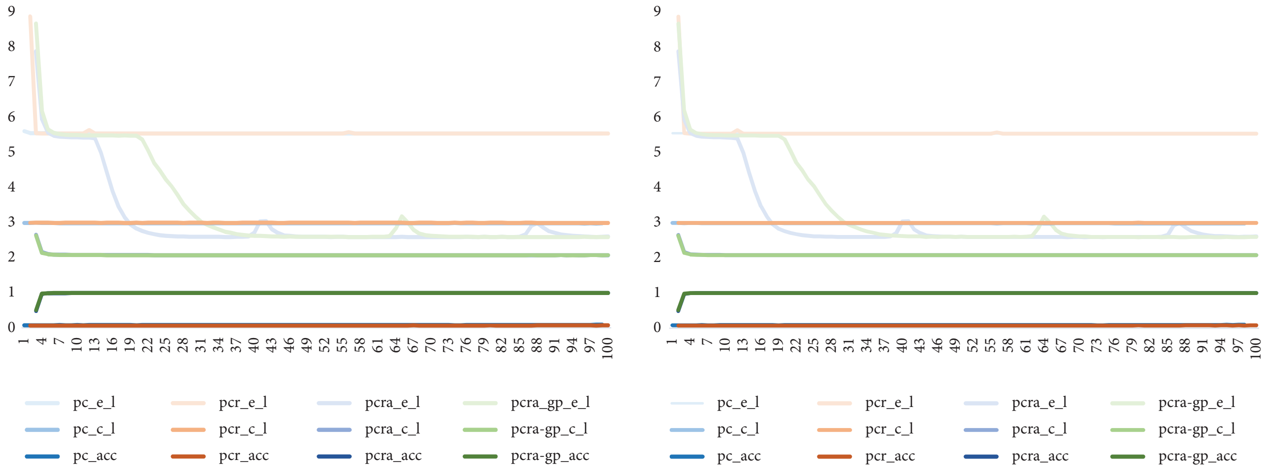
Table 12: Best results obtained from different parameters without epsilon. The minimum and maximum values are in bold in the table, and the maximum values are marked in red.

Epoch	Batch Size	Norm	Accuracy							
			IMDB		ChnsentiCorp-Htl		20news		Cnews	
			PCRA-gp	PCRA	PCRA-gp	PCRA	PCRA-gp	PCRA	PCRA-gp	PCRA
200	256	L	89.55%	89.52%	90.67%	91.08%	88.93%	87.49%	97.08%	97.18%
		B	89.36%	88.61%	90.75%	89.67%	87.52%	88.48%	97.14%	97.08%
	128	L	89.76%	89.38%	89.75%	90.25%	88.79%	87.87%	97.09%	96.60%
		B	89.38%	89.16%	90.33%	90.58%	87.33%	87.31%	96.92%	93.64%
	64	L	88.75%	89.21%	89.00%	88.92%	87.87%	86.80%	97.07%	96.52%
		B	89.42%	89.00%	91.42%	89.92%	86.67%	86.43%	97.02%	92.80%
100	256	L	89.71%	89.72%	90.83%	90.50%	89.17%	88.40%	97.18%	97.18%
		B	89.23%	88.89%	90.83%	90.83%	86.62%	88.10%	97.10%	93.60%
	128	L	89.91%	89.46%	90.58%	90.08%	87.97%	87.33%	97.13%	97.16%
		B	89.43%	86.48%	91.42%	90.25%	86.72%	86.72%	97.11%	92.99%
	64	L	89.24%	89.24%	89.42%	91.00%	87.89%	87.63%	97.15%	96.54%
		B	89.30%	89.05%	89.83%	89.83%	86.80%	86.78%	97.21%	92.93%

Table 13: Experimental results of different component combinations.

Model	IMDB	ChnSentiCorp-Htl	20news	Cnews
PC	64.35	67.33	8.98	61.69
PCR	59.57	57.58	7.04	54.85
PCRA	89.47	**90.33**	88.34	**96.91**
PCRA-GP	**89.73**	89.33	**89.33**	96.83

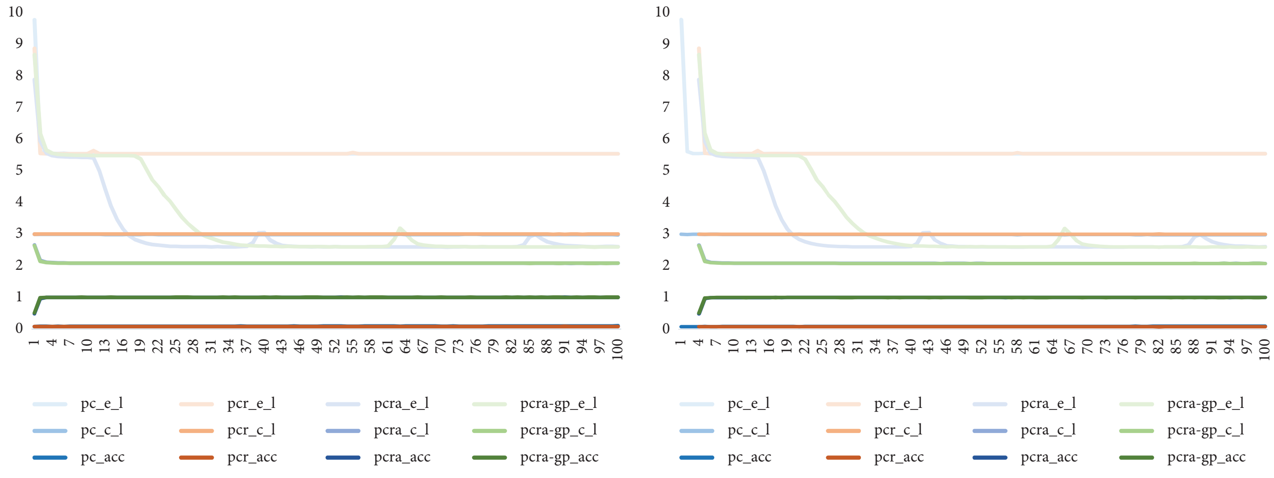
Bold values are the best accuracy scores.



(a)

(b)

Figure 7: Continued.



(c)

(d)

Figure 7: The training process data of different datasets. e_l means the encoder loss, c_l denotes the classifier loss, and acc means accuracy. (a) The training process data of IMDB. (b) The training process data of ChnSentiCorp-Htl. (c) The training process data of 20news. (d) The training process data of Cnews.

Table 14: Classification results under different temperature parameters. The temperature is adjusted for the maximum value in Table 12, which is trained at a fixed temperature of 0.05.

	Accuracy							
Temperature	IMDB		ChnSentiCorp-Htl		20news		Cnews	
	PCRA-GP (%)	PCRA (%)	PCRA-GP (%)	PCRA (%)	PCRA-GP (%)	PCRA (%)	PCRA-GP (%)	PCRA (%)
0.01	51.43	89.23	59.42	90.50	12.98	84.55	66.34	97.35
0.02	49.50	87.71	88.83	89.58	12.37	82.10	68.92	**97.39**
0.03	50.10	81.27	89.17	90.08	13.04	86.86	56.46	97.18
0.04	88.80	**89.84**	88.92	89.25	88.10	86.91	96.50	96.92
0.05	**89.91**	89.72	**91.42**	**91.08**	**89.17**	**88.48**	97.21	97.18
0.06	89.56	89.65	89.83	89.25	87.76	87.97	**97.23**	97.08

Bold values are the best accuracy scores.

Table 15: Model training time comparison (ms/step).

Model	IMDB		ChnSentiCorp-Htl		20news		Cnews	
	Encoder	Classifier	Encoder	Classifier	Encoder	Classifier	Encoder	Classifier
PC	95.77	27.98	51.38	11.94	138.71	35.78	211.87	58.18
PCR	**99.10**	**35.39**	57.31	16.52	**138.22**	**51.47**	**215.81**	**80.77**
PCRA	94.31	35.2	**57.70**	**16.8**	130.24	47.01	199.34	77
PCRA-GP	94.78	34.81	49.27	15	131.12	47	213.37	79.67

Bold values are the most time-consuming results.

gives a delayed effect to the loss compared to PCRA. The results indicate that joint features enhance the performance of the model and that adversarial training is an effective way for the classification task.

4.2.4. Tuning of the Temperature Parameter. In addition to the changes brought about by the above hyperparameter changes, temperature is also a hyperparameter introduced by SimCLR [10], which is an adjustable value to help learn hard negatives well. Table 14 lists the accuracy under a change in temperature. Obviously, PCRA-GP is easier to be affected than PCRA, and only three results are improved by adjusting temperature. In the IMDB dataset, the PCRA method is improved by 0.12% at a temperature of 0.04. Besides, the PCRA method improved by 0.21% at a temperature of 0.02 in the Cnews dataset. In addition, only PCRA is higher than PCRA-GP in the Cnews dataset. The results indicate that the best value of temperature is 0.05.

4.2.5. Model Training Efficiency. The training time is also recorded in Table 15 to observe the effect of different components on the training time. The experimental tests were performed on RTX 2080Ti. The results show that R increases the training time per step. The attention mechanism reduces the training time and verifies the effect of automatic feature selection. However, the role of the attention mechanism fails for ChnSentiCorp-Htl. This is mainly due to the small size of the dataset and the short text length. Furthermore, the lower time reduction during the classification stage is due to the use of the frozen encoder. The results indicate that the training time can be reduced by the attention mechanism in the multilevel feature fusion strategy.

5. Conclusion

In this research, PCRA-GP that takes the best advantages from both feature extraction and representation is proposed for text classification. First, the CNN, BiGRU, and attention mechanism are combined to obtain text features. Second, contrastive learning is used to train a projector which enhances feature vector embedding. Finally, a gradient penalty is used to generate feature vector disturbance by improving the robustness of PCRA. Experiments are performed on Chinese and English texts, involving sentiment classification and news text classification. The comparisons with some state-of-the-art baseline models demonstrate that the PCRA-GP model is more effective, efficient, and adaptable in terms of the classification quality and domain in most cases. In subsequent research, we will optimize the details of the model. The feature extractor will be adjusted to reduce the dimensionality appropriately to accelerate training speed. In addition, this work will be useful in other domain classification tasks.

Appendix

A The Detailed Results on K80 and RTX2080 Ti

The list of detailed results related to each dataset about text classification is shown in.

B Applied GP to Image Classification

Our proposed model is regarded as a pretrained encoder combining the SCL loss with GP. To show that the proposed training process remains applicable to image classification tasks after changing the encoder, the GP method is added to SCL [1] work to improve classification accuracy. The encoder is a ResNet50 system, with a batch size of 256, an epoch size of 50, a projection unit size of 128, a temperature of 0.05, and an Adam optimizer on the Tesla K80 system. The experimental results are shown in Figure 5. The raw value of epsilon means accuracy without the GP method. The classification accuracy is the largest when epsilon is 0.01, which is 0.35% higher than the accuracy of raw data. An epsilon value of 0.031 returns the lowest accuracy, and accuracy is improved with increasing epsilon.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Key R&D Project under Grant 2019YFB1705503, the Key-Area Research and Development Program of Guangdong Province under Grant 2019B010153002, the Key Program of NSFC-Guangdong Joint Funds under GrantU1801263, the National Natural Science Foundation of China under Grant 62002071, the Guangdong Provincial Key Laboratory of Cyber-Physical System under Grant 2020B1212060069, and the Guangdong Natural Science Foundation Project under Grant 2020A1515011409.

References

[1] P. Khosla, P. Teterwak, and C. Wang, "Supervised Contrastive Learning," 2021, http://arxiv.org/abs/2004.11362.

[2] J. Deng, L. Cheng, and Z. Wang, "Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classi fi cation," *Computer Speech & Language*, vol. 68, Article ID 101182, 2021.

[3] A. Jati, C. C. Hsu, M. Pal, R. Peri, W. AbdAlmageed, and S. Narayanan, "Adversarial attack and defense strategies for deep speaker recognition systems," *Computer Speech & Language*, vol. 68, Article ID 101199, 2021.

[4] C. Szegedy, W. Zaremba, and I. Sutskever, "Intriguing properties of neural networks," 2021, http://arxiv.org/abs/1312.6199.

[5] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive representation learning: a framework and review," *IEEE Access*, vol. 8, pp. 193907–193934, 2020.

[6] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2020, http://arxiv.org/abs/1911.05722.

[7] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6706–6716, Seattle, WA, USA, July 2020.

[8] A. van den Oord, Y. Li, and O. Vinyals, "Representation Learning with Contrastive Predictive Coding," 2020, http://arxiv.org/abs/1807.03748.

[9] C. Chen, J. Zheng, and H. Chen, "CosG: a graph-based contrastive learning method for fact verification," *Sensors*, vol. 21, no. 10, p. 3471, 2021.

[10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2021, http://arxiv.org/abs/2002.05709.

[11] Y. Zhang, X. Zhang, R. C. Qiu, J. Li, H. Xu, and Q. Tian, "Semi-supervised contrastive learning with similarity Co-calibration," 2021, http://arxiv.org/abs/2105.07387.

[12] M. Kim, J. Tack, and S. J. Hwang, "Adversarial Self-Supervised Contrastive Learning," 2021, http://arxiv.org/abs/2006.07589.

[13] Y. Belinkov and Y. Bisk, "Synthetic and natural noise both break neural machine translation," in *Proceedings of the Sixth International Conference on Learning Representations*, Vancouver, BC, Canada, May 2018.

[14] J. Wei and K. Zou, "Easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the 2019 Conference on Empirical Methods in*

Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6382–6388, Association for Computational Linguistics, Hong Kong, China, November 2019.

[15] S. Kobayashi, "Contextual augmentation: data augmentation by words with paradigmatic relations," 2021, http://arxiv.org/abs/1805.06201.

[16] J. Xu and Q. Du, "TextTricker: loss-based and gradient-based adversarial attacks on text classification models," *Engineering Applications of Artificial Intelligence*, vol. 92, Article ID 103641, 2020.

[17] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "HotFlip: white-box adversarial examples for text classification," 2021, http://arxiv.org/abs/1712.06751.

[18] N. Papernot, P. McDaniel, A. Swami, and R. Harang, "Crafting adversarial input sequences for recurrent neural networks," in *Proceedings of the MILCOM 2016 - 2016 IEEE Military Communications Conference*, pp. 49–54, Baltimore, Maryland, USA, November 2016.

[19] X. Sun and S. Sun, "Adversarial robustness and attacks for multi-view deep models," *Engineering Applications of Artificial Intelligence*, vol. 97, Article ID 104085, 2021.

[20] C. Guo, A. Sablayrolles, H. Jégou, and D. Kiela, "Gradient-based adversarial attacks against text transformers," 2021, http://arxiv.org/abs/2104.13733.

[21] P. Vijayaraghavan and D. Roy, "Generating black-box adversarial examples for text classifiers using a deep reinforced model," in *Machine Learning And Knowledge Discovery In Databases*, U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, and C. Robardet, Eds., pp. 711–726, Springer International Publishing, Midtown Manhattan, New York City, 2020.

[22] M. Arjovsky, S. Chintala, L. Bottou, and W. Gan, "Wasserstein GAN," 2021, https://arxiv.org/abs/1701.07875v3.

[23] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5769–5779, NIPS'17 Curran Associates Inc, Long Beach, CA,USA, December 2017.

[24] C. Xiao, B. Li, J. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pp. 3905–3911, International Joint Conferences on Artificial Intelligence Organization, Stockholm, Sweden, July 2018.

[25] M. Tezgider, B. Yildiz, and G. Aydin, "Text classification using improved bidirectional transformer," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 9, Article ID e6486, 2022.

[26] W. Dong, J. Wu, Z. Bai, W. Li, and W. Qiao, "Design of affinity-aware encoding by embedding graph centrality for graph classification," *Neurocomputing*, vol. 387, pp. 321–333, 2020.

[27] H. Wang, J. He, X. Zhang, and S. Liu, "A short text classification method based on N-gram and CNN," *Chinese Journal of Electronics*, vol. 29, no. 2, pp. 248–254, 2020.

[28] F. Abid, M. Alam, M. Yasir, and C. Li, "Sentiment analysis through recurrent variants latterly on convolutional neural network of Twitter," *Future Generation Computer Systems*, vol. 95, pp. 292–308, 2019.

[29] Y. Liu, P. Li, and X. Hu, "Combining context-relevant features with multi-stage attention network for short text classification," *Computer Speech & Language*, vol. 71, Article ID 101268, 2022.

[30] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the Thirty-Third Aaai Conference on Artificial Intelligence/Thirty-First Innovative Applications of Artificial Intelligence Conference/Ninth Aaai Symposium on Educational Advances in Artificial Intelligence*, pp. 7370–7377, Assoc Advancement Artificial Intelligence, Honolulu Hawaii, January 2019, https://www.webofscience.com/wos/alldb/full-record/WOS:000486572501112.

[31] C. Huang and G. Liu, "Sentiment analysis of network comments based on GCNN," in *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence - CSAI '18*, pp. 409–413, ACM Press, Shenzhen China, December 2018.

[32] S. Yang and Y. Tang, "Text classification based on convolutional neural network and attention model," in *Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pp. 67–73, Long Beach, CA,USA, September 2020.

[33] J. Wenzhen, Z. Hong, and Y. Guocai, "An efficient character-level and word-level feature fusion method for Chinese text classification," *Journal of Physics Conference Series*, vol. 1229, no. 1, Article ID 012057, 2019.

[34] G. Liu and J. Guo, "Bidirectional LSTM with attention mechanism and convolutional layer for text classification," *Neurocomputing*, vol. 337, pp. 325–338, 2019.

[35] F. Wu, T. Zhang, A. H. Souza, C. Fifty, T. Yu, and K. Q. Weinberger, "Simplifying graph convolutional networks," 2021, http://arxiv.org/abs/1902.07153.

[36] Y. Lin, Y. Meng, and X. Sun, "BertGCN: transductive text classification by combining GCN and BERT," 2021, http://arxiv.org/abs/2105.05727.

[37] H. Zhu and P. Koniusz, "Simple spectral graph convolution," in *Proceedings of the International Conference on Learning Representations*, New Orleans, LA, USA, May 2021, https://openreview.net/forum?id=CYO5T-YjWZV.

[38] Ö Köksal and E. H. Yılmaz, "Improving automated Turkish text classification with learning-based algorithms," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 11, Article ID e6874, 2022.

[39] D. Grießhaber, N. T. Vu, and J. Maucher, "Low-resource text classification using domain-adversarial learning," *Computer Speech & Language*, vol. 62, Article ID 101056, 2020.

[40] B. Skrlj, M. Martinc, J. Kralj, N. Lavrac, and S. Pollak, "tax2vec: constructing interpretable features from taxonomies for short text classification," *Computer Speech & Language*, vol. 65, Article ID 101104, 2021.

[41] C. Pan, J. Huang, J. Gong, and X. Yuan, "Few-shot transfer learning for text classification with lightweight word embedding based models," *IEEE Access*, vol. 7, pp. 53296–53304, 2019.

[42] W. Liu, G. Cao, and J. Yin, "Bi-level attention model for sentiment analysis of short texts," *IEEE Access*, vol. 7, pp. 119813–119822, 2019.

[43] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Association for Computational Linguistics, Doha, Qatar, September 2014.

[44] C. Raffel and D. P. W. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," 2021, http://arxiv.org/abs/1512.08756.

[45] K. Sohn, "Improved deep metric learning with multi-classN-pair loss objective," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*,

pp. 1857–1865, Curran Associates Inc, Canada, in Barcelona, October 2016.

[46] J. Dai, C. Chen, and Y. Li, “A backdoor attack against LSTM-based text classification systems,” *IEEE Access*, vol. 7, pp. 138872–138878, 2019.