Inceptive Transformer: Augmenting Transformer Models with Multi-Scale Feature Learning for Generalized Cross-Domain Text Classification

Anonymous ACL submission

Abstract

In this work we introduce *Inceptive Transformer*, an architecture designed to enhance transformer based models by incorporating a multi-scale feature extraction module inspired by inception networks. Unlike conventional transformers, which compress the information from all tokens into a single `[CLS]` token to capture the global context of the sequence, our model balances local and global dependencies by dynamically weighting token interactions, enriching their representations for downstream tasks. We propose a generalizable framework that can be integrated into both domain-specific pre-trained models (e.g., BERTweet, BioBERT, CT-BERT) and general-purpose models like RoBERTa. We evaluate our models on a diverse range of text classification tasks, including emotion recognition, irony detection, disease identification, and anti-COVID vaccine tweets classification, covering both multi-class and multi-label settings. Results show that our models consistently outperform baseline transformers by 1% to 9% while maintaining efficiency, highlighting the versatility and generalization capabilities of Inceptive Transformers across diverse domains and applications.

1 Introduction

Since its introduction, the transformer architecture (Vaswani et al., 2017) has revolutionized the field of natural language processing (NLP), thanks to an innovative self-attention mechanism capable of capturing complex contextual relationships across tokens. Transformer-based models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), Electra (Clark et al., 2020), and XL-Net (Yang et al., 2019) have demonstrated impressive performance across a wide range of NLP tasks, including text classification, question answering, and named entity recognition. However, in practice, we often encounter domain-specific text—be it medical, scientific, business, legal, or social media content. These texts come with their own unique language and nuanced stylistic patterns, which are difficult for general purpose models like BERT or RoBERTa to capture. To address this, domain-specific BERT-based models like BioBERT (Lee et al., 2019), SciBERT (Beltagy et al., 2019), Legal-BERT (Chalkidis et al., 2020), BERTweet (Nguyen et al., 2020) have emerged, which have been further pre-trained on domain-specific corpora to capture the unique language, terminology, and stylistic features of various specialized fields.

Despite their success, transformer models still have limitations, particularly in capturing short-range dependencies between tokens (Guo et al., 2019; Li et al., 2021) that are often important for classification. A significant issue we observed in our research is their reliance on the `[CLS]` token for text classification, where the model aggregates all token embeddings into a single representation. Although convenient, we found that this approach can lead to information loss, as the single `[CLS]` token is insufficient to capture fine-grained contextual nuances or localized cues critical for tasks like emotion recognition or irony detection. This limitation is especially problematic for multi-label tasks, which require token-level attention rather than a single sequence-level summary.

To address these limitations of traditional transformer models, we propose Inceptive Transformers, which aim to enhance both general-purpose and domain-specific transformer models by using convolutional filters. These filters are designed to recognize key phrases or word combinations that are indicative of specific classifications. Our model uses an initial transformer layer to capture the global context and long-range dependencies within the input sequence. Following this, we introduce a multi-scale convolutional module with varying kernel sizes to extract local dependencies and patterns, complementing the global representations learned by the transformer layers. These enriched features

are then processed by a self-attention mechanism, which dynamically assigns weights to tokens based on their task-specific contribution, thus allowing the model to effectively prioritize relevant tokens.

Our experiments show that Inceptive Transformers consistently outperform baseline transformer models across both general-purpose (e.g., RoBERTa) and domain-specific (e.g., BERTweet, BioBERT) architectures in text-classification. Evaluated on four distinct tasks across three diverse domains, our models achieved moderate **(1%)** to significant **(9%)** improvements in key metrics like accuracy and F1-score. Notably, in disease identification, our model *InceptiveRoBERTa* outperformed the domain-specific pre-trained model BioBERT-base, while *InceptiveBioBERT* performed at a similar level to BioBERT-large despite requiring one-third of inference time.

The major contributions of our work are as follows.

- We introduce the *Inceptive Transformer* architecture, designed to capture both global context and local features effectively while identifying and prioritizing the most important tokens across the entire input sequence— thus alleviating the limitations of standard transformer models.
- We propose a generalizable framework that can enhance both general-purpose models like RoBERTa and domain-specific pre-trained models. Through comprehensive evaluation, we show that our inceptive models perform strongly across diverse datasets while maintaining efficiency.
- We demonstrate the effectiveness of our models through extensive experiments and comparisons, ablation studies, statistical significance testing, and interpretations of the findings.

2 Related Work

There are a number of text classification methods, ranging from traditional machine learning approaches like decision trees (Law and Ghosh, 2022), support vector machines (SVM), k-nearest neighbors (KNN) (Hanifelou et al., 2018), and ensemble learning (Zhu et al., 2023; Wu et al., 2016), to more advanced deep learning techniques like RNN and LSTM (Lai et al., 2015; Onan, 2022). Convolutional networks have also been been used (Conneau et al., 2017; Choi et al., 2019; Yao et al., 2019; Soni et al., 2022), but they often struggle with capturing long-range dependencies in text.

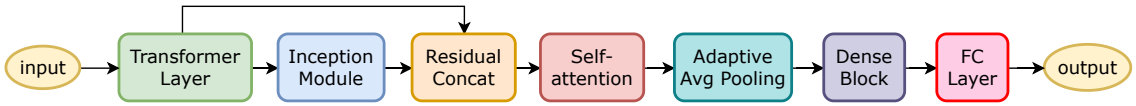
After the transformer architecture (Vaswani et al., 2017) was introduced, many works have combined convolution with transformers, but these works mostly focus on vision related tasks (Fang et al., 2022; Si et al., 2022; Yuan et al., 2023). Application on NLP domain remains limited to a few works (Zheng and Yang, 2019; Wan and Li, 2022; Chen et al., 2022; Wu et al., 2024) — which mostly focus on improving a particular transformer model, like BERT or XLNet. In comparison, we provide a general architecture capable of improving different types of transformer models, both domain-specific and general purpose.

3 Inceptive Transformer

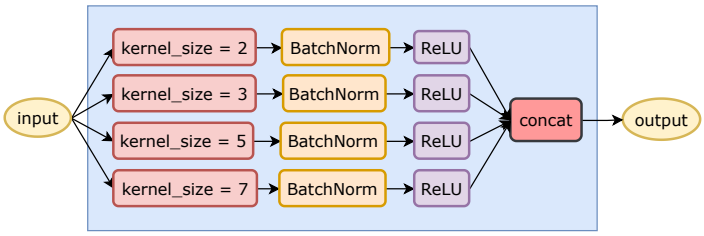
3.1 Motivation

Transformer-based models like RoBERTa, and domain-specific pre-trained variants such as BioBERT, BERTweet, CT-BERT, and SciBERT, rely on token-level embeddings derived primarily from self-attention layers to capture global dependencies and context within text sequences. In our experiment, we visualized the attention maps of these models, which show a strong bias in attention towards the `[CLS]` token, while intermediate tokens often receive comparatively lower attention. The `[CLS]` token is a weighted aggregation of all token embeddings in the sequence, which the model relies on to represent the entire sequence for classification tasks. This bias suggests an underutilization of contextual and local dependencies, potentially limiting the model's ability to effectively capture fine-grained patterns and hierarchical structures present in textual data.

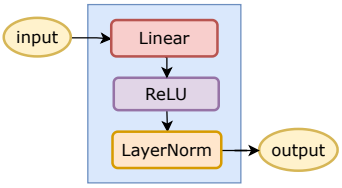
Our model is designed to address this gap by incorporating convolutional operations, which excel at capturing local patterns and hierarchical structures in data (Gu et al., 2018; Li et al., 2022). CNNs are typically not used on textual data due to their inability to capture long-range dependencies. However, using convolution makes sense in our model because it operates on embeddings generated by a transformer— not on raw text. This allows the convolutional operations to refine the already globally contextualized embeddings by emphasizing fine-grained, local features that might otherwise be overlooked. Furthermore, instead of using a single convolution layer with a fixed kernel size, we use an inception module (Szegedy et al., 2015) to apply convolutions with multiple kernel sizes to learn features at different levels of granularity. This enables



(a) Full workflow of an inceptive transformer model



(b) Inception module



(c) Dense block

Figure 1: Inceptive Transformer model architecture

multi-scale feature extraction, allowing the model to simultaneously capture both token-level patterns and phrase-level dependencies.

The applicability of our model is not limited to general-purpose transformers like RoBERTa. Domain-specific pre-trained models such as BioBERT, CT-BERT, or BERTweet show similar attention biases as BERT and RoBERTa, leading to challenges in capturing local and hierarchical dependencies. By integrating our model's multi-scale feature extraction approach, these domain-specific variants can also be enhanced, improving their ability to represent diverse patterns within specialized input data. This versatility makes our model a robust addition to any transformer-based architecture.

3.2 Model Architecture

In this section we describe the end-to-end workflow of our model. Fig.1 illustrates the full model architecture.

3.2.1 Input Preparation

The input to our model is pre-processed text data, which need to be tokenized using an appropriate pre-trained tokenizer corresponding to the chosen transformer model. Mathematically, the input can be represented as:

$$X = [x_1, x_2, \ldots, x_L]$$

where L is the sequence length, and each x_i corresponds to a token from the text. X is passed to the transformer layer.

3.2.2 Transformer Layer

The first layer of our architecture is a transformer-based model, such as RoBERTa, BioBERT, BERTweet, or CT-BERT. Given the input X, the transformer layer generates a sequence of hidden states:

$$H = [h_1, h_2, \ldots, h_L]$$

where $H \in \mathbb{R}^{B \times L \times d}$, B is the batch size, L is the sequence length, and d is the hidden state dimension. Each $h_i \in \mathbb{R}^d$ represents the contextual embedding for token x_i. A dropout layer is applied to H to prevent overfitting.

3.2.3 Inception Module

The primary task of this layer is to extract multi-scale local features. The inception module receives contextual embeddings H generated by the transformer and applies parallel convolutional layers with small kernel sizes k (e.g., $k = 2, 3, 5, 7$) to learn features at different granularities. Smaller kernels ($k = 2$ or 3) capture fine-grained token-level relationships, such as modifiers or word pair dependencies, whereas larger kernels ($k = 5$ or 7) capture slightly broader local patterns, such as syntactic or semantic relationships over small phrases.

Given an input tensor H, each branch of the inception module applies a 1D convolution along the sequence dimension. For a convolution with kernel size k, the output at position i is computed as:

$$Y_i = \sum_{j=0}^{k-1} W_j \cdot H_{i+j} + b$$

where W is the filter, b is the bias, and Y represents the extracted feature map. The convolution is performed using multiple filters simultaneously, resulting in c output channels in each branch (here c

is a tunable hyperparameter). To maintain the original sequence length, we apply manual padding: for a kernel of size 2, we right-pad by 1, and for kernels of sizes 3, 5, and 7, we apply symmetric (left and right) padding.

After the convolution, each branch further processes its output using batch normalization to stabilize and accelerate the training process, followed by a ReLU activation to introduce non-linearity. Finally, the outputs from all four branches are concatenated along the channel dimension to form a combined feature map $C \in \mathbb{R}^{B\times L\times(4\cdot c)}$. To preserve information from the original transformer output, a residual connection from H is added with C to form $R \in \mathbb{R}^{B\times L\times(d+4\cdot c)}$. This residual connection ensures that the original features are retained alongside the multi-scale features. This combined representation, enriched with both global and multi-scale local features, is then passed to the self-attention layer for further processing.

3.2.4 Self-Attention

While the transformer layer uses self-attention to contextualize token embeddings, these mechanisms are applied early in the model flow. After the inception module extracts multi-scale features, an additional self-attention mechanism is necessary to capture dependencies and relationships across the enriched feature space R. This ensures that tokens contributing the most to the task are effectively prioritized and selected, thus allowing the model to focus on the most relevant features.

Given $R \in \mathbb{R}^{B\times L\times d_R}$, the attention mechanism maps it to query Q, key K, and value V:

$$Q = RW_Q, \quad K = RW_K, \quad V = RW_V$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d_R\times d_A}$, d_R is the enriched feature space dimension, and d_A is the attention head dimension. The attention scores are computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_A}}\right)V$$

Since we use multi-headed attention, the outputs of multiple attention heads are concatenated and projected back to the original feature space:

$$A = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W_O$$

where $W_O \in \mathbb{R}^{(h\cdot d_A)\times d_R}$ is a learnable projection matrix and h is the number of attention heads, another tunable hyperparameter. The attention output $A \in \mathbb{R}^{B\times L\times d_R}$ captures refined dependencies across both token positions and feature scales.

3.2.5 Adaptive Average Pooling

To reduce the sequence-level representation A to a fixed-size vector suitable for classification, global average pooling is applied across the sequence length. Given the attention output $A \in \mathbb{R}^{B\times L\times d_R}$, we first permute it to $\mathbb{R}^{B\times d_R\times L}$. Afterwards, adaptive average pooling computes the average over the entire sequence for each feature channel, regardless of the input length, by dynamically adjusting the pooling region. Mathematically:

$$P_{b,i} = \frac{1}{L}\sum_{j=1}^{L} a_{b,i,j}$$

where $a_{b,i,j}$ is the value of the ith feature channel at the jth position for the bth example. This produces a tensor $P \in \mathbb{R}^{B\times d_R\times 1}$, which is then squeezed to yield a final pooled representation $P \in \mathbb{R}^{B\times d_R}$.

3.2.6 Dense Block

The pooled representation P is passed through a dense block to further refine features and enhance non-linear interactions. This block consists of three sublayers. First, a fully connected layer is used to reduce the dimensionality:

$$D_1 = PW_d + b_d$$

where $W_d \in \mathbb{R}^{d_R\times d_D}$, $b_d \in \mathbb{R}^{d_D}$, and d_D is the target dimensionality (e.g., 512). Next, ReLU activation is used to introduce non-linearity, and layer normalization is used to stabilize the output. The output of the dense block $D \in \mathbb{R}^{B\times d_D}$ represents a compact and refined feature set ready for classification.

3.2.7 Final Classification

The output D is passed to a linear classifier, which computes logits for each class:

$$O = DW_f + b_f$$

where $W_f \in \mathbb{R}^{d_D\times C}$ and $b_f \in \mathbb{R}^C$. The logits $O \in \mathbb{R}^{B\times C}$ are interpreted based on the task.

4 Experimental Setup

In this section we discuss the datasets, model training and evaluation procedures, and hyperparameters used.

4.1 Datasets

We have selected three datasets from diverse domains that cover both multi-class and multi-label settings. The TweetEval dataset (Barbieri et al., 2020) is a benchmark for seven diverse Twitter-specific classification tasks, from which we have selected two: emotion recognition (Mohammad et al., 2018) and irony detection (Van Hee et al., 2018). The first one is a multi-class problem while the latter is binary classification. For multi-label, we have chosen two datasets: OHSUMED [1] from biomedical domain, which is a collection of abstracts of medical journal articles; and CAVES (Poddar et al., 2022) for anti-covid vaccine concerns, such as concerns about the vaccine ingredients, side-effects of vaccines, monetary motivations of the pharmaceutical companies, political and geographic issues, etc.

Table 1: Dataset statistics. C : number of classes or labels; $\overline{C}$: average number of labels per instance (for multi-label); and $\overline{L}$: average token length of each text.

Dataset	**#Texts**	**C**	$\overline{C}$	$\overline{L}$
Emotion	5,052	4	–	24.35
Irony	4,601	2	–	21.54
OHSUMED	13,929	23	1.66	289.51
CAVES	9,921	11	1.16	58.35

4.2 Model Training and Evaluation

Each input sequence was tokenized using a model-specific tokenizer and then passed through the model to generate logits. For multi-class classification, the model predicts mutually exclusive class probabilities using softmax activation and cross-entropy loss, whereas for binary and multi-label tasks, it outputs non-exclusive probabilities with sigmoid activation and binary cross entropy with logits loss. During backpropagation, gradients were clipped to a maximum norm of 1.0 to ensure numerical stability. The AdamW optimizer (Kingma and Ba, 2014) with weight decay was used to update the model weights.

The training process was conducted iteratively over multiple epochs, with a Cosine Annealing learning rate scheduler. At the end of each epoch, the model was evaluated on the validation dataset to monitor key metrics, including accuracy, F1-score, AUC-ROC (multi-class), AUPR (multi-label), and inference time. The best model was selected based on accuracy for binary and multi-class classification tasks, and F1-score for multi-label tasks. Each model was run 10 times on each dataset. The models were trained and evaluated using Google Colab Pro+ (40GB A100 GPU). However, all of our models can be run on 16 GB GPUs (e.g. Kaggle P100). We used the `transformer` version 4.48.3.

[1] OHSUMED-link

4.3 Hyperparameters

Table 2: Hyperparameters.

Hyperparameter	Value
Sequence Length	128, 512 (ohsumed)
Batch Size	32
Epochs	12
Learning Rate	1e-5
Weight Decay	1e-3, 1e-4 (ohsumed, caves)
Attention Heads	4, 8
Sigmoid threshold	0.5

The hyperparameters used in this experiment are shown in Table 2. These values were determined empirically. Two hyperparameters related to our model architecture are the number of output channels in convolution branches, and the number of attention heads. We found that 4 attention heads work well with 16 output channels, while 8 heads work better with higher number of channels. Number of output channel affects performance the most. A detailed comparison of various output channels for each model can be found in appendix C.

5 Results

5.1 Comparative Performance

In this section, we compare the performance of the inception-enhanced models with that of the transformer-based models. For each data set, we selected two transformer models: RoBERTa as a general-purpose model and a domain-specific pre-trained model. Multi-class performance comparison (in terms of accuracy) is shown in Table 3, while multi-label comparison (F1-score) is shown in Table 4. A detailed comparison can be found in appendix A. We ran each model in each dataset 10 times and reported the average metric. Performance in each run can be found in appendix B. It should be noted here that *i*BERTweet-32 means it is an Inceptive BERTweet model with 32 output channels in each convolution layer. This number

was determined through extensive hyperparameter tuning.

Table 3: Multi-class performance comparison in test set

Model	Avg Accuracy	Inference Time (s)
Emotion Recognition		
BERTweet	83.29	2.83
*i*BERTweet-64	**84.11**	2.93
RoBERTa	81.69	2.88
*i*RoBERTa-16	**82.22**	3.00
Irony Detection		
BERTweet	82.69	1.59
*i*BERTweet-16	**84.51**	1.62
RoBERTa	75.15	1.60
*i*RoBERTa-32	**76.86**	1.66

Table 4: Multi-label performance comparison in test set

Model	Avg F1-score	Inference Time (s)
OHSUMED		
BioBERT	67.16	53.26
*i*BioBERT-128	**73.32**	57.97
BioBERT-Large	74.30	154.00
RoBERTa	64.08	61.22
*i*RoBERTa-128	**68.98**	65.12
RoBERTa-Large	71.75	159.01
CAVES		
CT-BERT	74.24	10.27
*i*CTBERT-16	**74.86**	10.56
RoBERTa	71.11	4.67
*i*RoBERTa-32	**72.11**	4.78

Multi-class Performance

In the task of emotion recognition, Inceptive BERTweet-32 achieved an accuracy of 84.02, which is a **0.98%** improvement over BERTweet (83.29). InceptiveRoBERTa-16 (82.22) improved on RoBERTa (81.69) by **0.65%**. However, in the binary classification task of irony detection, InceptiveBERTweet-16 improved on BERTweet by a higher margin of **2.20%** (84.51 vs 82.69). InceptiveRoBERTa-32 also improved on RoBERTa by a similar margin of **2.28%**.

Multi-label Performance

In OHSUMED disease identification, our Inceptive BioBERT model performed superbly; achieving an average F1 score of 73.32, which is a **9.16%** improvement on BioBERT (67.16). Inceptive RoBERTa (68.98) also offered a significant performance uplift of **7.65%** over RoBERTa (64.08). There are two interesting observations here. First, Inceptive RoBERTa achieved a higher F1-score (68.98) than BioBERT (67.16), which is pre-trained on biomedical literature. This shows the generalization capability of our inception mechanism. Second, Inceptive BioBERT performed at a similar level as BioBERT-large, despite the latter taking almost thrice as much to run and requiring significantly more compute power. This observation highlights our models' ability to achieve significant performance improvement while maintaining efficiency.

Finally, in CAVES dataset, the integration of inception module resulted in improvements of **0.84%** over the domain-specific model CT-BERT, and **1.41%** over RoBERTa.

5.2 Statistical Significance Testing

Table 5: Wilcoxon Signed-Rank Test Results and Performance Gain.

Dataset	Models	Gain	p-value
Emotion	BT, iBT-64	**+0.98%**	0.001953
Irony	BT, iBT-16	**+2.20%**	0.005859
OHSUMED	BioBERT, iBioBERT-128	**+9.16%**	0.001953
CAVES	RoBERTa, iRoBERTa-32	**+1.41%**	0.001953

For statistical significance testing, we performed the Wilcoxon signed-rank test, which is a non-parametric test and suitable for paired comparison on the same test set. Each model was run 10 times, and the average accuracy or F1-score was recorded

for statistical analysis. As shown in Table 5, the p-value in each test is below the 0.05 significance threshold. Therefore, we conclude that the gain achieved are statiscillay significant.

5.3 Performance Interpretation

The attention maps for the baseline transformers (BERTweet, BioBERT, CT-BERT), plotted in Fig. 2, show that the attention weights are heavily skewed toward the initial `[CLS]` token, while the rest of the tokens receive negligible attention. This is a typical behaviour for transformer models. However, this single token can be inadequate for tasks that require nuanced understanding of token-level dependencies. For example, in tasks like irony detection, where localized cues or specific tokens (e.g., sarcasm markers) are crucial, over-reliance on the `[CLS]` token can lead to information loss. Similarly, multi-label tasks like disease identification often demand token-level attention rather than a single sequence-level summary. In such cases, the `[CLS]` token may fail to represent the sequence adequately.

On the contrary, the attention maps presented in Fig. 3 highlight a more balanced distribution of attention weights across the sequence. Tokens that were overlooked by transformer-based models, particularly those in the middle of the sequence, now receive higher attention, reflecting their contextual importance. This improvement is a direct result of the architectural enhancements introduced in our models. The inceptive transformers first capture the global context using the initial transformer layer, and then apply parallel convolutional layers with small kernel sizes to capture short-range dependencies between neighboring tokens. The outputs of the convolutional layers are concatenated with the original embeddings from the transformer layer via a residual connection, ensuring that the original token representations are not lost. This enriched representation is then passed to the self-attention layer. Since each token embedding now contains both global and local features, tokens across the sequence compete more effectively for attention. This allows the self-attention mechanism to dynamically assign weights to the tokens based on their contribution to the task, as evident from the attention maps.

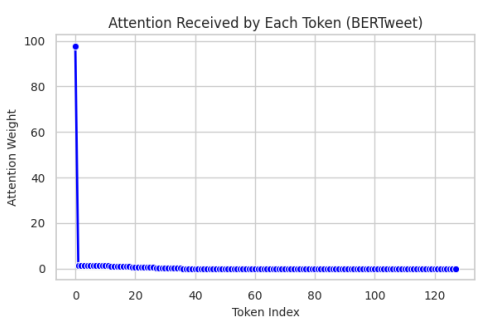
Our inceptive transformer models are able to adapt their attention patterns to suit the specific requirements of each task. For tasks like emotion recognition and irony detection, the input data often contains localized cues that are highly indicative of the target class. For example: In emotion recognition, key emotional expressions such as "happy," "sad," or "angry" are often concentrated in a few specific words or phrases within the sentence. Similarly, in irony detection, sarcasm or irony is usually conveyed through specific linguistic patterns or markers like exaggeration or contrasting terms, which are localized to certain parts of the sequence. As a result, the model's attention tends to focus sharply on these critical tokens while assigning less importance to the rest of the sequence, as shown in Fig. 3a. In contrast, the OHSUMED dataset, used for disease identification, involves longer, more complex sequences such as medical abstracts or documents. Here, relevant information is often dispersed throughout the text rather than being localized. For example, mentions of symptoms, treatments, or diagnoses may appear in different parts of the text, each contributing to the prediction of a specific disease label. Since the relevant features are distributed across the sequence, the model must maintain a more balanced and diffuse attention pattern. This behavior is evident in the attention maps for disease identification (Fig. 3b), where attention is spread across the sequence to capture multiple independent or overlapping features.

5.4 Ablation Study

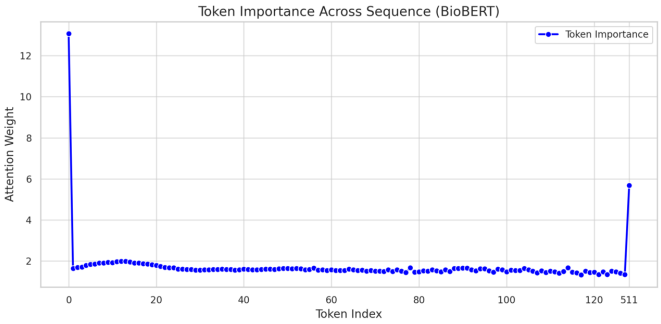
Table 6: Ablation study results. BT: BERTweet, Attn.: Attention.

Model	Full	No Attn.	No Dense
*i*BT (emotion)	84.11	83.63	83.51
*i*BT (irony)	84.51	82.61	82.48
*i*BioBERT	73.32	71.54	69.00
*i*RoBERTa (OHSUMED)	68.98	68.57	68.57
*i*RoBERTa (CAVES)	72.11	71.31	71.38

The results of the ablation study in Table 6 show that both the self-attention mechanism and the dense block positively contribute to the model's performance. Without self-attention, the models' performances fall by 0.5% - 2.4%. The removal of the dense block reduces the performance by 0.6% - 5.9%. The differences are most pronounced in the OHSUMED dataset, where our inception models achieve the most significant improvement. On the

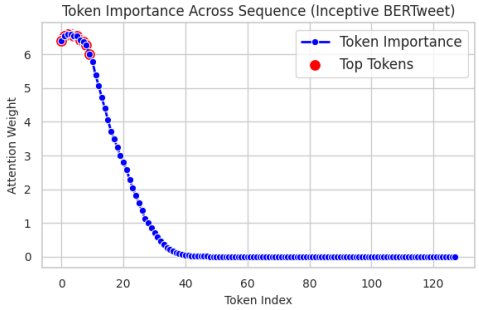


(a) BERTweet (Irony detection)

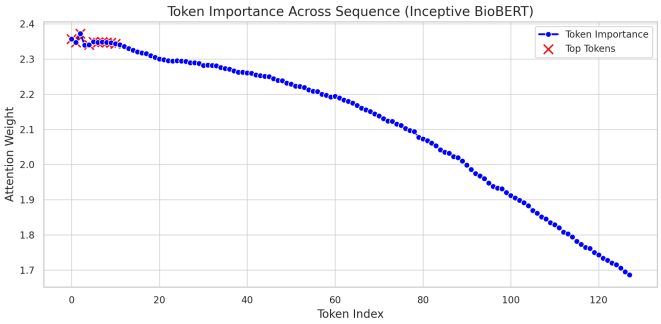


(b) BioBERT (OHSUEMD)

Figure 2: Attention received by each token in transformer models. The spike at the end in BioBERT is due to the `[SEP]` token, used to separate sequences.



(a) Inceptive BERTweet (Irony detection)



(b) Inceptive BioBERT (OHSUMED)

Figure 3: Attention received by each token in inceptive transformer models. Models were run on OHSUMED dataset with 512 tokens, but we show the first 128 tokens only for better visualization.

contrary, these components contribute least to the multi-class emotion recognition dataset.

6 Conclusion

In this paper we presented *Inceptive Transformer*, a general convolution-based framework that enhances the performance of both general-purpose transformer models like RoBERTa and domain-specific pre-trained language models such as BERTweet, BioBERT, and CT-BERT. Our experiments show statistically significant performance gains ranging from 1% to 9%. Moreover, our approach consistently delivers strong results across diverse domains while maintaining computational efficiency. In future work, we plan to evaluate our model on larger-scale datasets and explore additional techniques to further boost performance.

7 Limitations

The main limitation of our work is that we could not develop a single version of the inception mechanism that works uniformly across all baseline models. For example, while an inception module with 128 output channels works best on BioBERT, 16 (for irony detection) and 32 or 64 (for emotion recognition) output channels are more suitable for BERTweet. Another limitation is that we did not explore additional performance enhancement techniques, as our primary focus was on providing a fair comparison between the models.

8 Acknowledgment

While writing the paper, we used AI assistance (chatGPT) for polishing a few sentences. We also used AI assistance (chatGPT, copilot) for some minor debugging of the code. The authors remain fully responsible for the text in the manuscript as well as the code.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. 2020. TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scib-

ert: Pretrained language model for scientific text. In *EMNLP*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Xinying Chen, Peimin Cong, and Shuo Lv. 2022. A long-text classification method of chinese news based on bert and cnn. *IEEE Access*, 10:34046–34057.

Byung-Ju Choi, Jun-Hyung Park, and SangKeun Lee. 2019. Adaptive convolution for text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2475–2485.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1107–1116. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186.

Jinsheng Fang, Hanjiang Lin, Xinyu Chen, and Kun Zeng. 2022. A hybrid network of cnn and transformer for lightweight image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1103–1112.

Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. 2018. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377.

Maosheng Guo, Yu Zhang, and Ting Liu. 2019. Gaussian transformer: A lightweight approach for natural language inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6489–6496.

Zahra Hanifelou, Peyman Adibi, Sayyed Amirhassan Monadjemi, and Hossein Karshenas. 2018. Knn-based multi-label twin support vector machine with priority of labels. *Neurocomputing*, 322:177–186.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29.

Anwesha Law and Ashish Ghosh. 2022. Multi-label classification using binary tree of classifiers. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(3):677–689.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Pengfei Li, Peixiang Zhong, Kezhi Mao, Dongzhe Wang, Xuefeng Yang, Yunfeng Liu, Jianxiong Yin, and Simon See. 2021. Act: an attentive convolutional transformer for efficient text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13261–13269.

Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2022. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 33.

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. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4221–4231. Association for Computational Linguistics.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14. Association for Computational Linguistics.

Aytuğ Onan. 2022. Bidirectional convolutional recurrent neural network architecture with group-wise enhancement mechanism for text sentiment classification. *J. King Saud Univ. Comput. Inf. Sci.*, 34:2098–2117.

Soham Poddar, Azlaan Mustafa Samad, Rajdeep Mukherjee, Niloy Ganguly, and Saptarshi Ghosh. 2022. Caves: A dataset to facilitate explainable classification and summarization of concerns towards

covid vaccines. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. 2022. Inception transformer. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22.

Sanskar Soni, Satyendra Singh Chouhan, and Santosh Singh Rathore. 2022. Textconvonet: a convolutional neural network based architecture for text classification. *Applied Intelligence (Dordrecht, Netherlands)*, 53:14249 – 14268.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50.

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, pages 5998–6008. Curran Associates, Inc.

C. X. Wan and B. Li. 2022. Financial causal sentence recognition based on bert-cnn text classification. *Journal of Supercomputing*, 78:6503–6527.

D. Wu, Z. Wang, and W. Zhao. 2024. Xlnet-cnn-gru dual-channel aspect-level review text sentiment classification method. *Multimed Tools Appl*, 83:5871–5892.

Qingyao Wu, Mingkui Tan, Hengjie Song, Jian Chen, and Michael K. Ng. 2016. Ml-forest: A multi-label tree ensemble method for multi-label classification. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2665–2680.

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*, pages 5754–5764.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph convolutional networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:7370–7377.

Feiniu Yuan, Zhengxiao Zhang, and Zhijun Fang. 2023. An effective cnn and transformer complementary network for medical image segmentation. *Pattern Recognition*, 136.

Shaomin Zheng and Meng Yang. 2019. A new method of improving bert for text classification. In *Intelligence Science and Big Data Engineering. Big Data and Machine Learning*, pages 442–452. Springer International Publishing.

Xiaoyan Zhu, Jiaxuan Li, Jingtao Ren, Jiayin Wang, and Guangtao Wang. 2023. Dynamic ensemble learning for multi-label classification. *Information Sciences*, 623:94–111.

A Full Performance Comparison

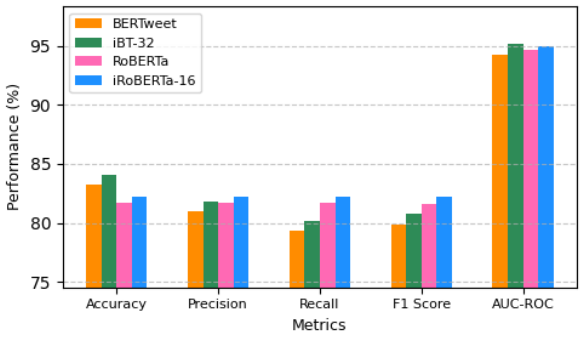


Figure 4: Performance comparison in Emotion Recognition

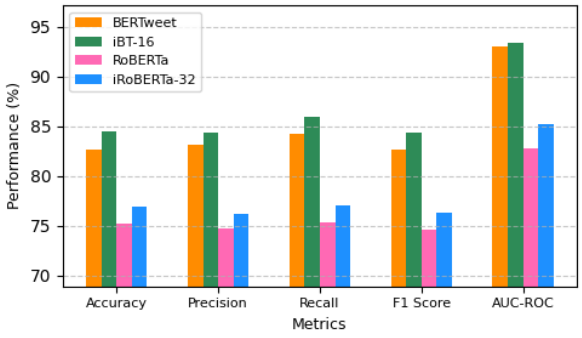


Figure 5: Performance comparison in Irony Detection

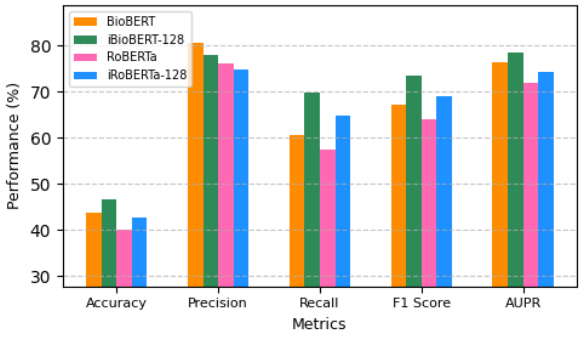


Figure 6: Performance comparison in OHSUMED

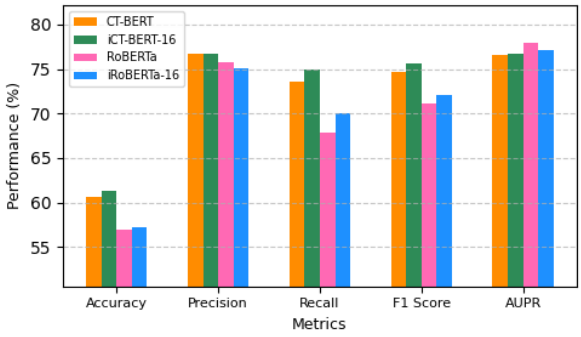


Figure 7: Performance comparison in CAVES

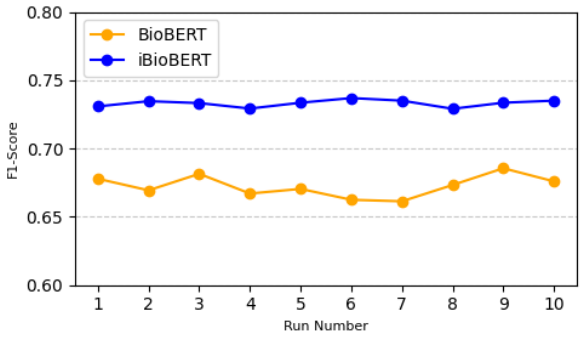


Figure 10: Comparison across all runs in OHSUMED

B Comparison across All Runs

Fig. 8, 9, 10, and 11 show the comparison of baseline pretrained models (BERTweet, BioBERT, RoBERTa) against the inception models across all 10 runs.

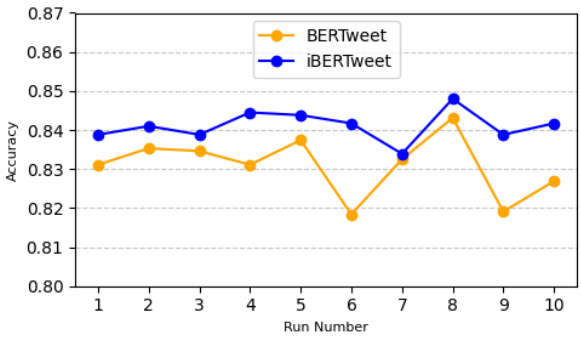


Figure 8: Comparison across all runs in Emotion Recognition

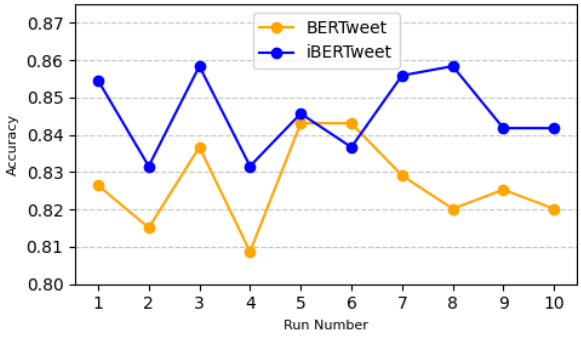


Figure 9: Comparison across all runs in Irony Detection

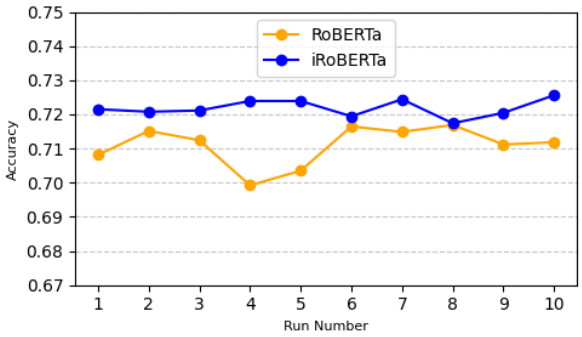


Figure 11: Comparison across all runs in CAVES

C Effect of Convolution Output Channels

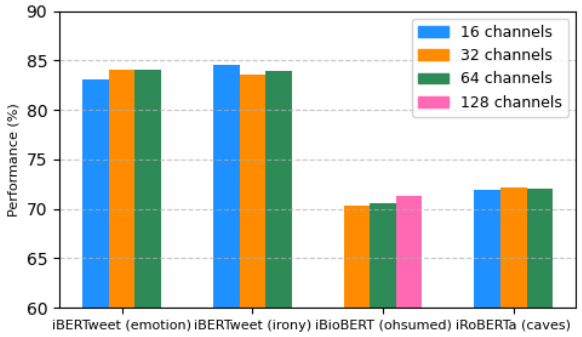


Figure 12: Conv. output channels vs performance (accuracy for multi-class, F1-score for multi label)