






# Local Information Assisted Attention-Free Decoder for Audio Captioning

Feiyang Xiao , *Graduate Student Member, IEEE*, Jian Guan , *Member, IEEE*, Haiyan Lan ,  
Qiaoxi Zhu , *Member, IEEE*, and Wenwu Wang , *Senior Member, IEEE*

**Abstract**—Automated audio captioning aims to describe audio data with captions using natural language. Existing methods often employ an encoder-decoder structure, where the attention-based decoder (e.g., Transformer decoder) is widely used and achieves state-of-the-art performance. Although this method effectively captures global information within audio data via the self-attention mechanism, it may ignore the event with short time duration, due to its limitation in capturing local information in an audio signal, leading to inaccurate prediction of captions. To address this issue, we propose a method using the pretrained audio neural networks (PANNs) as the encoder and local information assisted attention-free Transformer (LocalAFT) as the decoder. The novelty of our method is in the proposal of the LocalAFT decoder, which allows local information within an audio signal to be captured while retaining the global information. This enables the events of different duration, including short duration, to be captured for more precise caption generation. Experiments show that our method outperforms the state-of-the-art methods in Task 6 of the DCASE 2021 Challenge with the standard attention-based decoder for caption generation.

**Index Terms**—Automated audio captioning, local information, attention-free transformer.

## I. INTRODUCTION

**A**UTOMATED audio captioning (AAC) is an intermodal translation task that converts input audio into a text description, i.e., caption, using natural language [1], [9]. It benefits various applications, such as intelligent and content oriented machine-to-machine interaction and automatic content description [2]–[5]. Existing AAC methods usually employ an encoder-decoder structure, where the encoder is used to obtain an audio

Manuscript received 12 April 2022; revised 26 June 2022; accepted 2 July 2022. Date of publication 8 July 2022; date of current version 22 July 2022. This work was supported in part by the Natural Science Foundation of Heilongjiang Province under Grant YQ2020F010 and in part by Newton Institutional Links Award from the British Council under Grant 623805725. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Rosangela Coelho. (*Corresponding author: Jian Guan.*)

Feiyang Xiao, Jian Guan, and Haiyan Lan are with the Group of Intelligent Signal Processing (GISP), College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China (e-mail: xiaofeiyang128@gmail.com; j.guan@hrbeu.edu.cn; lanhaiyan@hrbeu.edu.cn).

Qiaoxi Zhu is with the Centre for Audio, Acoustics and Vibration, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: qiaoxi.zhu@gmail.com).

Wenwu Wang is with the Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH Guildford, U.K. (e-mail: w.wang@surrey.ac.uk).

Digital Object Identifier 10.1109/LSP.2022.3189536

TABLE I  
EXAMPLES SHOW HOW THE LOSS OF LOCAL INFORMATION AND IMPRECISE CAPTION PREDICTION DEGRADE THE QUALITY OF CAPTIONS

Example	Method	Caption
1. Loss of local information	Ground Truth	Wind chimes are playing while <i>people are talking</i> in the background
	P-Transformer [12]	Wind chimes chime wind chimes chime wind chimes clang wind chimes tinkle wind chimes and wind chimes tinkle wind chimes chime
2. Imprecise caption prediction	Ground Truth	<i>A dog barks</i> a few times and two men talk to each other
	P-Transformer [12]	People are talking while <i>birds are chirping</i> in the background

feature of the audio input, while the decoder is used to generate texts from the audio feature [6]–[10]. The pretrained audio neural networks (PANNs) [11] module has been widely used in the encoder to extract audio features from the audio signal, while the attention-based Transformer has been used in the decoder to model the global information within audio features with a self-attention mechanism [7]. With the PANNs encoder and the attention-based decoder, several methods [2], [12]–[14] have been developed, achieving state-of-the-art performance in Task 6 of the DCASE 2021 Challenge.

The standard self-attention mechanism [7], however, does not explicitly consider the local information within audio features and may ignore the event with short duration. Moreover, with the softmax function used in the self-attention strategy, the frames that contain predicted events get high weights, while those corresponding to rare events may get low weights. As a result, the prediction of captions for describing such events may be inaccurate. Table I shows examples of generated captions using such methods, assuming the ground truth captions are correct. For instance, P-Transformer [12] may miss local information about some events, e.g., “*people are talking*” in example 1, or generate inaccurate descriptions, e.g., “*A dog barks*” is wrongly interpreted as “*birds are chirping*” in example 2.

To address the above issues, we propose a new decoder by leveraging the attention-free Transformer (AFT) method and its variant AFT-local [15]. AFT-local was initially proposed for modeling data within a local region in image processing and character-level language modeling [15]. Although AFT-local has the ability to capture locality while maintaining global connectivity within features, it does not have a decoder structure to model the sequential information for text description, which, nevertheless, is required in the audio captioning task. Other studies, e.g., [16] for image classification and [17] for speaker verification have also considered local self-attention, but they cannot be used directly to model the sequential information for generating text description.

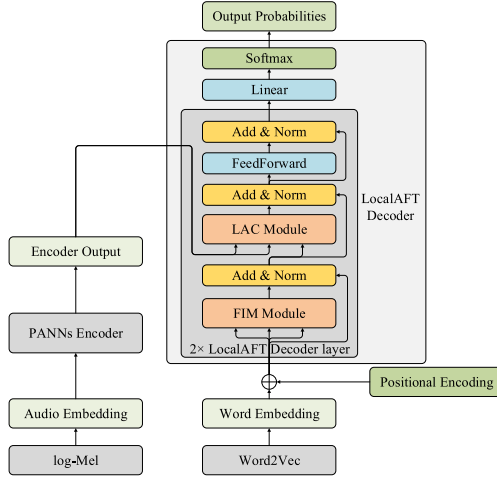


Fig. 1. The framework of the proposed automated audio captioning method.

In this letter, we present a LocalAFT decoding method based on [15] for the AAC task. More specifically, our LocalAFT method has two key modules, namely, the future interference masking (FIM) module and the local information assisted captioning (LAC) module. The FIM module is designed to model the sequential word features, which can avoid interference from future words and use element-wise multiplication to capture the global information of the input word features. In the LAC module, we introduce a window function and a learnable weight matrix, where the window function is used to constrain the weights in the learnable weight matrix. With the element-wise multiplication and the constrained weight matrix, our LAC module can capture the local information while maintaining the global information of audio features, to predict latent features for captions, thereby achieving the inter-modal integration between text and audio information. Thus, the proposed decoder can capture the short-time event, along with events of longer duration, and generate more precise captions.

In our work, the PANNs algorithm based encoder is employed to extract high-quality audio features as the input of the LocalAFT decoder. Therefore, the overall captioning method including both encoder and decoder is denoted as P-LocalAFT. The proposed method is evaluated and compared with the state-of-the-art methods that use PANNs as the encoder and the standard attention mechanism as the decoder [2], [12]–[14]. Experiments are conducted on the AudioCaps dataset [18] for pretraining the model and the Clotho-v2 dataset [1] for fine-tuning the model. Results show that the proposed method achieves better performance than the state-of-the-art methods.

## II. PROPOSED METHOD

The proposed P-LocalAFT uses a PANNs encoder to extract audio features and a LocalAFT decoder to incorporate the local information about audio events to guide caption prediction. Fig. 1 shows the framework of the P-LocalAFT method.

### A. PANNs Encoder

The encoder of the proposed method is used to extract audio features, which are the input of the decoder for caption prediction. The input to the PANNs encoder is the log-Mel spectrogram

$\mathbf{X} \in \mathbb{R}^{T \times F}$ , where  $T$  denotes the number of time frames and  $F$  denotes the dimension of Mel band.

To extract the audio feature, we choose the 10-layer CNN module from PANNs [11] which is pretrained on AudioSet [19]. The 10-layer CNN module has four convolutional blocks and two linear layers. Each convolutional block has two  $3 \times 3$  convolutional layers with ReLU activation function and batch normalization. The number of channels in the convolutional blocks is 64, 128, 256, and 512, respectively. There are  $2 \times 2$  average pooling layers between convolutional blocks. After the convolutional blocks, global pooling is employed on the Mel band dimension. Then, two linear layers are used to output the audio feature  $\mathbf{H} \in \mathbb{R}^{L \times D}$ , as the input to the LocalAFT decoder, where  $L$  and  $D$  represent the number of time frames and the dimension of the spectral feature at each frame, respectively. Here  $D$  is set as 128.

### B. LocalAFT Decoder

Inspired by [15], in our LocalAFT decoder, local information about audio events is incorporated to guide caption prediction. The LocalAFT decoder has two inputs. One is the output of the PANNs encoder  $\mathbf{H} \in \mathbb{R}^{L \times D}$ , the other is the word embedding  $\mathbf{W} = [\mathbf{w}_0, \dots, \mathbf{w}_{N-1}]^T \in \mathbb{R}^{N \times D}$  from a pretrained word2vec model, where  $N$  denotes the length of the target caption and  $\top$  denotes matrix transpose. Note that,  $\mathbf{w}_0$  is the token  $\langle \text{sos} \rangle$  representing the start of a sequence, and it is not from a caption. To predict the  $n$ -th word  $\mathbf{w}_n$  of the caption, the posterior probability is calculated as

$$p(\mathbf{w}_n | \mathbf{H}, \mathbf{W}_{pre}) = \text{Decoder}(\mathbf{H}, \mathbf{W}_{pre}), \quad (1)$$

where  $\mathbf{W}_{pre} = [\mathbf{w}_0, \dots, \mathbf{w}_{n-1}]^T$  are the previously predicted words. The LocalAFT decoder has two key modules, future interference masking (FIM) and local information assisted captioning (LAC), as shown in Fig. 1. FIM is developed to deal with sequential word embedding and avoid the interference from the future words after the  $(n-1)$ -th word, i.e.,  $[\mathbf{w}_n, \dots, \mathbf{w}_{N-1}]^T$ . LAC is developed to capture the local information while maintaining the global information about audio events.

1) *Future Interference Masking*: This module processes the input word feature  $\mathbf{Y}$  (i.e., the word embedding with positional encoding), where  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n, \dots, \mathbf{y}_N]^T \in \mathbb{R}^{N \times D}$ . To predict the  $n$ -th word, the input feature  $\mathbf{Y}$  should only contain the features of the previously generated words (i.e.,  $[\mathbf{y}_1, \dots, \mathbf{y}_n]^T$ ), and the feature after  $\mathbf{y}_n$  should be ignored. To this end, we design a mask matrix  $\mathbf{M} \in \mathbb{R}^{N \times N}$  of learnable weights in the FIM module, defined as

$$\mathbf{M} = \begin{bmatrix} m_{1,1} & -\infty & \cdots & -\infty \\ m_{2,1} & m_{2,2} & \ddots & \vdots \\ \vdots & \vdots & \ddots & -\infty \\ m_{N,1} & m_{N,2} & \cdots & m_{N,N} \end{bmatrix}, \quad (2)$$

where “ $-\infty$ ” is the negative infinity and  $\exp(-\infty) = 0$ . As a result,  $\exp(\mathbf{M})$  becomes a lower triangular matrix. This helps reduce the interference from subsequent words, i.e., words after the current words.

Following the above masking, we define the future interference masking function  $FIM(\cdot)$  to remove the future interference in  $\mathbf{Y}$ , as follows

$$\begin{aligned} \mathbf{y}_n^{FIM} &= FIM(\mathbf{Y}) \\ &= \sigma(\mathbf{Q}_n) \odot \frac{\sum_{i=1}^I \exp(m_{n,i}) \odot \exp(\mathbf{K}_i) \odot \mathbf{V}_i}{\sum_{i=1}^I \exp(m_{n,i}) \odot \exp(\mathbf{K}_i)}, \end{aligned} \quad (3)$$

where  $\mathbf{y}_n^{FIM}$  denotes the output feature of the FIM module corresponding to the  $n$ -th word  $\mathbf{w}_n$ ,  $m_{n,i}$  is an element of matrix  $\mathbf{M}$ ,  $\sigma(\cdot)$  denotes the sigmoid function,  $\odot$  denotes element-wise multiplication, and  $I$  is the length of the previously generated caption. Here,  $\mathbf{Q}$ ,  $\mathbf{K}$  and  $\mathbf{V}$  are the linear mapping results of  $\mathbf{Y}$ , which are calculated following [15].  $\mathbf{Q}_n$  is the  $n$ -th row vector of  $\mathbf{Q}$ .  $\mathbf{K}_i$  and  $\mathbf{V}_i$  are the  $i$ -th row vectors of  $\mathbf{K}$  and  $\mathbf{V}$ , respectively.

According to (3), the FIM module is free from the words information after the previously generated caption, since  $\exp(m_{n,i})=0$  is used to mask future interference when  $i > n$ . The mask matrix  $\mathbf{M}$  can act on all captions of the dataset and adjust the weight on its lower triangular part. Instead of implementing self-attention through matrix multiplication operation, the FIM applies element-wise multiplication operation, following [15]. Therefore, the FIM module involves global information similar to self-attention for each caption feature, by applying  $\mathbf{M}$  over the whole dataset.

Then, we can obtain the output feature  $\hat{\mathbf{y}}_n^{FIM}$  of the  $n$ -th word, as follows

$$\hat{\mathbf{y}}_n^{FIM} = LN(\mathbf{y}_n^{FIM} + \mathbf{y}_n), \quad (4)$$

where  $\mathbf{y}_n$  is the  $n$ -th vector of the input feature  $\mathbf{Y}$ , corresponding to the  $(n-1)$ -th word  $\mathbf{w}_{n-1}$ , and  $LN(\cdot)$  denotes the layer normalization function.

2) *Local Information Assisted Captioning*: The local information assisted captioning (LAC) module uses both  $\hat{\mathbf{y}}_n^{FIM}$  and audio feature  $\mathbf{H}$  as inputs, aiming to learn the semantic information from the audio feature and generate latent caption feature. To learn the global information from the whole dataset, a weight matrix  $\mathbf{Z} \in \mathbb{R}^{N \times L}$  is defined and learned in LAC. Meanwhile, similar as [15], a window function  $f_{\mathcal{L}}(\cdot)$  is used to constrain the weight matrix  $\mathbf{Z}$ , as follows

$$f_{\mathcal{L}}(z_{n,l}) = \begin{cases} z_{n,l}, & \text{if } (l-n) < s, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

where  $z_{n,l} \in \mathbf{Z}$  with  $1 \leq n \leq N$  and  $1 \leq l \leq L$ . In (5),  $s$  is the size of the local region, which helps  $\mathbf{Z}$  attend local information in audio feature  $\mathbf{H}$ . Different from [15], the local area defined in (5) is not lower bounded with respect to  $s$  and  $n$ . Specifically,  $\mathbf{Z}$  correlates the caption with the audio feature, such that the local information corresponding to the  $n$ -th word information can be adaptively learned from the 1-st to the  $(s+n)$ -th audio feature vectors.

The output feature  $\mathbf{y}_n^{LAC}$  corresponding to the  $n$ -th word  $\mathbf{w}_n$  is calculated as

$$\begin{aligned} \mathbf{y}_n^{LAC} &= LAC(\hat{\mathbf{y}}_n^{FIM}, \mathbf{H}) \\ &= \sigma(\bar{\mathbf{y}}_n) \odot \frac{\sum_{l=1}^L \exp(f_{\mathcal{L}}(z_{n,l})) \odot \exp(\mathbf{H}_l^K) \odot \mathbf{H}_l^V}{\sum_{l=1}^L \exp(f_{\mathcal{L}}(z_{n,l})) \odot \exp(\mathbf{H}_l^K)}, \end{aligned} \quad (6)$$

where  $\bar{\mathbf{y}}_n$  is the linear mapping result of  $\hat{\mathbf{y}}_n^{FIM}$ ,  $\mathbf{H}^K$  and  $\mathbf{H}^V$  are the linear mapping results of  $\mathbf{H}$ . Here,  $\mathbf{H}_l^K$  and  $\mathbf{H}_l^V$  are the  $l$ -th row vector of  $\mathbf{H}^K$  and  $\mathbf{H}^V$ , respectively.

Note that, when  $z_{n,l}$  is not within the window for the local region,  $\exp(f_{\mathcal{L}}(z_{n,l}))$  has the value as 1 (i.e.,  $\exp(0) = 1$ ). Therefore,  $\mathbf{Z}$  can learn local information (i.e.,  $f_{\mathcal{L}}(z_{n,l}) = z_{n,l}$ ), while allowing the LAC module to retain the global information by the Transformer (i.e.,  $f_{\mathcal{L}}(z_{n,l}) = 0$ ). In summary, the LAC module can model both the global and local information for caption prediction. Finally, the output feature of LAC is calculated as follows

$$\hat{\mathbf{y}}_n^{LAC} = LN(\mathbf{y}_n^{LAC} + \hat{\mathbf{y}}_n^{FIM}). \quad (7)$$

To get a more effective word embedding, we pretrain a word2vec language model [20] by combining the captions of both Clotho-v2 [1] and an external AAC dataset (i.e., AudioCaps [18]). We also use the AudioCaps dataset to pretrain the whole P-LocalAFT method, and then fine-tune it on Clotho-v2 dataset, with the same training strategy as [12].

### III. EXPERIMENTS

#### A. Dataset

We use two datasets in our experiments, i.e., Clotho-v2 [1] and AudioCaps [18]. Clotho-v2 was released for Task 6 of the DCASE 2021 Challenge, including 3839, 1045 and 1045 audio clips for the development, validation and evaluation splits, respectively. Each audio clip has five captions. The audio clips have a varying duration ranging from 15 to 30 seconds, with captions of 8 to 20 words. In our experiments, the development and validation splits are combined together to form a new training set of 4884 audio clips. The evaluation split is used for the performance evaluation.

AudioCaps [18] is another AAC dataset whose audio signals are from AudioSet [19], including 49838, 495 and 975 audio clips for the development, validation and evaluation splits, respectively, in the initial version. However, because of the rules of YouTube, the dataset available in this experiment includes 44366, 458 and 905 audio clips for the development, validation and evaluation, respectively. Each audio clip has the duration of 10 seconds. We combine the development and the validation splits to pretrain our P-LocalAFT method.

#### B. Experimental Setup

The local region window size  $s$  in (5) was set as 80 based on empirical tests. In practice, this hyper-parameter needs to be chosen and tuned in terms of the data to be processed. Positional encoding and padding mask were used in the LocalAFT decoder. For the whole P-LocalAFT method, SpecAugment and mix-up strategies were introduced to improve generalization following [12]. The cross-entropy loss with label smooth [21] was used with Adam optimizer [22] to optimize the network. The batch size was set as 16, and the learning rate was set as 0.0001. The decoder used a teacher forcing strategy in training and beam search strategy with the beam size of 5 in evaluation.

Following DCASE 2021 Challenge, all the methods are evaluated by machine translation metrics (i.e., BLEU<sub>n</sub>, ROUGE<sub>l</sub> and METEOR) and captioning metrics (CIDE<sub>r</sub>, SPICE and SPIDE<sub>r</sub>). BLEU<sub>n</sub> [23] measures a modified n-gram precision. ROUGE<sub>l</sub> [24] is a score based on the longest common sub-sequence. METEOR [25] is a harmonic mean of weighted unigram precision and recall. CIDE<sub>r</sub> [26] is a weighted cosine similarity of n-grams, which reflects the grammar and fluency properties of captions ignored in METEOR and SPICE.

TABLE II  
PERFORMANCE COMPARISON ON THE EVALUATION SPLIT OF THE CLOTHO-V2 DATASET

Method	BLEU <sub>1</sub>	BLEU <sub>2</sub>	BLEU <sub>3</sub>	BLEU <sub>4</sub>	ROUGE <sub>l</sub>	METEOR	CIDE <sub>r</sub>	SPICE	SPIDE <sub>r</sub>
P-Transformer [12]	0.565	0.372	0.253	0.171	0.377	0.172	0.413	<b>0.123</b>	0.268
P-Conformer [13]	0.546	0.356	0.243	0.165	0.369	0.163	0.381	0.110	0.246
P-Meshed Memory [14]	0.563	0.367	0.244	0.158	0.371	0.170	0.406	0.119	0.262
P-Temporal Attention [2]	0.576	0.377	0.252	0.164	0.382	<b>0.178</b>	0.421	0.122	0.271
<b>P-GlobalAFT</b>	0.563	0.374	0.255	0.171	0.374	0.171	0.414	0.120	0.267
<b>P-LocalAFT</b>	<b>0.578</b>	<b>0.387</b>	<b>0.267</b>	<b>0.179</b>	<b>0.390</b>	0.177	<b>0.434</b>	0.122	<b>0.278</b>

SPICE [27] is the F-score of semantic propositions extracted from caption and reference. SPIDE<sub>r</sub> [28] is the mean score between CIDE<sub>r</sub> and SPICE.

### C. Performance Comparison

The proposed P-LocalAFT is compared with state-of-the-art methods [2], [12]–[14], which all use PANNs encoder and standard attention decoders, i.e., P-Transformer [12], P-Conformer [13], P-Meshed Memory [14], and P-Temporal Attention [2]. Noting that, our proposed method does not consider the influence of reinforcement learning, so the reinforcement learning processes in both P-Transformer and P-Temporal Attention are not involved. Moreover, the P-Transformer method has the same training strategy as the proposed P-LocalAFT. The results of P-Conformer, P-Meshed Memory, and P-Temporal Attention are those from the results of DCASE 2021 Challenge.

The results are shown in Table II. Our proposed P-LocalAFT method outperforms other state-of-the-art methods (i.e., P-Transformer, P-Conformer, P-Meshed Memory and P-Temporal Attention), in terms of BLEU<sub>n</sub>, ROUGE<sub>l</sub>, CIDE<sub>r</sub> and SPIDE<sub>r</sub>. More specifically, apart from the SPICE metric, our P-LocalAFT method performs better than P-Transformer in all the other metrics, showing that our LocalAFT decoder outperforms the standard Transformer decoder, especially in terms of grammar and fluency. The proposed method has slightly lower performance in SPICE and METEOR as compared to the best baselines. This could be because the window size is fixed and not optimized for capturing the semantic information of the audio events or scenes with varying duration.

### D. Ablation Study

To demonstrate the effectiveness of using local information for audio captioning, a P-LocalAFT variant without using the local region (i.e., P-GlobalAFT) is evaluated in our ablation study. Specifically, P-GlobalAFT does not use the local region window function (i.e., (5)) to constrain the learnable weight matrix  $\mathbf{Z}$  in the LAC module. The result is also given in Table II. It can be observed from Table II that, without incorporating the local information, P-LocalAFT degenerates to P-GlobalAFT, resulting in similar captioning performance to the standard Transformer-based method, i.e., P-Transformer. This result indicates that the proposed P-LocalAFT improves performance in caption prediction, due to the effective use of the local information from the audio events and caption contents, along with the use of the global information.

For better understanding the effect of local information for audio captioning, we present two examples in Table III. In example 1, the audio event “*people talk*” is described by our P-LocalAFT

TABLE III  
ILLUSTRATION FOR AUDIO CAPTION USING THE METHODS WITH AND WITHOUT LOCAL INFORMATION

Example	Method	Caption
1. Loss of local information	Ground Truth	Wind chimes are playing while <i>people are talking</i> in the background
	P-Transformer [12]	Wind chimes chime wind chimes chime wind chimes clang wind chimes tinkle wind chimes and wind chimes tinkle wind chimes chime
	P-GlobalAFT	Wind chimes clang together in the wind as the wind blows
	<b>P-LocalAFT</b>	Wind chimes clang together as <i>people talk</i> in the background
2. Imprecise caption prediction	Ground Truth	<i>A dog barks</i> a few times and <i>two men talk</i> to each other
	P-Transformer [12]	<i>People are talking</i> while <i>birds are chirping</i> in the background
	P-GlobalAFT	<i>A dog barks</i> while <i>birds chirp</i> in the background
	<b>P-LocalAFT</b>	<i>A dog barks</i> while <i>people talk</i> in the background

method but lost in the caption generated by P-Transformer and P-GlobalAFT. This demonstrates that our LocalAFT decoder can capture the audio event of short duration, overcoming the loss of local information in the standard Transformer decoder. Meanwhile, in example 2, the audio event “*a dog barks*” is wrongly interpreted as “*birds are chirping*” by the P-Transformer method and the audio event “*two men talk*” is wrongly interpreted as “*birds chirp*” by the P-GlobalAFT method, which shows that the lack of the effective use of the local information may also degrade the quality of the prediction of the acoustic events. In contrast, the proposed P-LocalAFT method precisely describes the events “*a dog barks*” and “*people talk*” in the generated caption, verifying that the proposed LocalAFT decoder can achieve more precise caption prediction. The source codes and more examples with audio clips are available.<sup>1</sup>

## IV. CONCLUSION

We have presented a novel automated audio captioning method that employs a PANNs encoder to extract audio features and exploits a LocalAFT decoder to incorporate local and global information from audio data. Experimental results show that the proposed method improves captioning performance as compared to state-of-the-art methods by incorporating local information, and overcomes the limitation of these attention-based methods. In a future work, we may explore the choice of adaptive window size to capture local information from audio clips with acoustic events and scenes of different duration.

<sup>1</sup>[Online]. Available: <https://github.com/LittleFlyingSheep/P-LocalAFT/tree/main/examples>

## REFERENCES

- [1] K. Drossos, S. Lipping, and T. Virtanen, "Clotho An audio captioning dataset," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Barcelona, Spain, 2020, pp. 736–740.
- [2] X. Xu, Z. Xie, M. Wu, and K. Yu, "The SJTU system for DCASE2021 challenge task 6: Audio captioning based on encoder pre-training and reinforcement learning," DCASE2021 Challenge, Tech. Rep., Jul. 2021.
- [3] X. Mei, X. Liu, J. Sun, M. Plumbley, and W. Wang, "Diverse audio captioning via adversarial training," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 8882–8886.
- [4] X. Liu *et al.*, "Leverage pre-trained BERT for audio captioning," in *Proc. 30th Eur. Signal Process. Conf.*, Belgrade, Serbia, 2022.
- [5] X. Liu *et al.*, "CLAAC: A contrastive loss for audio captioning," in *Proc. DCASE Workshop*, 2021, pp. 196–200.
- [6] K. Drossos, S. Adavanne, and T. Virtanen, "Automated audio captioning with recurrent neural networks," in *Proc. IEEE Workshop Appl. Signal Process Audio Acoust.*, New Paltz, NY, USA, 2017, pp. 374–378.
- [7] X. Mei, X. Liu, Q. Huang, M. Plumbley, and W. Wang, "Audio captioning transformer," in *Proc. DCASE Workshop*, 2021, pp. 211–215.
- [8] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, and S. Saito, "A transformer-based audio captioning model with keyword estimation," in *Proc. INTERSPEECH*, Shanghai, China, 2020, pp. 1977–1981.
- [9] X. Mei, X. Liu, M. Plumbley, and W. Wang, "Automated audio captioning: An overview of recent progress and new challenges," 2022, *arXiv:2205.05949*.
- [10] X. Xu, M. Wu, and K. Yu, "A comprehensive survey of automated audio captioning," 2022, *arXiv:2205.05357*.
- [11] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2880–2894, 2020.
- [12] X. Mei *et al.*, "An encoder-decoder based audio captioning system with transfer and reinforcement learning," in *Proc. DCASE Workshop*, 2021, pp. 206–210.
- [13] C. Narisetty, T. Hayashi, R. Ishizaki, S. Watanabe, and K. Takeda, "Leveraging state-of-the-art ASR techniques to audio captioning," in *Proc. DCASE Workshop*, 2021, pp. 160–164.
- [14] Z. Chen, D. Zhang, J. Wang, and F. Deng, "Audio captioning with meshed-memory transformer," DCASE2021 Challenge, Tech. Rep., Jul. 2021.
- [15] S. Zhai *et al.*, "An attention free transformer," 2021, *arXiv:2105.14103*.
- [16] Z. Liu *et al.*, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 10012–10022.
- [17] B. Han, Z. Chen, and Y. Qian, "Local information modeling with self-attention for speaker verification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Singapore, 2022, pp. 6727–6731.
- [18] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist. Human Lang. Technol.*, Minneapolis, MN, USA, 2019, pp. 119–132.
- [19] J. F. Gemmeke *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, New Orleans, LA, USA, 2017, pp. 776–780.
- [20] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Representations*, Scottsdale, AZ, USA, 2013.
- [21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 2818–2826.
- [22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, San Diego, CA, USA, 2015.
- [23] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. Annu. Meeting Assoc. Comput. Linguist.*, 2002, pp. 311–318.
- [24] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. Text Summarization Branches Out*, 2004, pp. 74–81.
- [25] A. Lavie and A. Agarwal, "METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments," in *Proc. Annu. Meet. Assoc. Comput. Linguist.*, Prague, Czech Republic, 2007, pp. 228–231.
- [26] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 4566–4575.
- [27] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic propositional image caption evaluation," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 382–398.
- [28] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved image captioning via policy gradient optimization of SPIDer," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, 2017, pp. 873–881.