Interspeech 2022
18-22 September 2022, Incheon, Korea

CT-SAT: Contextual Transformer for Sequential Audio Tagging

Yuanbo Hou[*1], *Zhaoyi Liu*[2], *Bo Kang*[1], *Yun Wang*[3], *Dick Botteldooren*[*1]

[*]WAVES Research Group [1]Ghent University, Belgium [2]KU Leuven, Belgium [3]Meta AI, USA

{Yuanbo.Hou, Bo.Kang, Dick.Botteldooren}@UGent.be
Zhaoyi.Liu@student.kuleuven.be, Maigoakisame@gmail.com

Abstract

Sequential audio event tagging can provide not only the type information of audio events, but also the order information between events and the number of events that occur in an audio clip. Most previous works on audio event sequence analysis rely on connectionist temporal classification (CTC). However, CTC's conditional independence assumption prevents it from effectively learning correlations between diverse audio events. This paper first introduces the Transformer into sequential audio tagging, since Transformers perform well in sequence-related tasks. To better utilize contextual information of audio event sequences, we draw on the idea of bidirectional recurrent neural networks, and propose a contextual Transformer (cTransformer) with a bidirectional decoder that could exploit the forward and backward information of event sequences. Experiments on the real-life polyphonic audio dataset show that, compared to CTC-based methods, the cTransformer can effectively combine the fine-grained acoustic representations from the encoder and coarse-grained audio event cues to exploit contextual information to successfully recognize and predict the audio event sequence in polyphonic audio clips.

Index Terms: Audio tagging, sequential audio tagging, connectionist temporal classification, contextual Transformer

1. Introduction

Audio Tagging (AT) is a multi-label classification task that identifies which target audio events occur in an audio clip. AT only predicts the type of events occurring in an audio clip, not the order between these events nor how many times they occur. As audio events naturally occur sequentially in a sequence, there is often a relationship between the preceding and following events. This paper studies sequential audio tagging (SAT), which aims to learn such relationships between events and predict sequences of audio events in audio clips. SAT can be applied for tasks such as audio classification [1], audio captioning [2], acoustic scene analysis [3], and event anticipation [4].

Previous works related to SAT mostly rely on connectionist temporal classification (CTC) [5] to identify event sequences. Paper [6] explores the possibility of polyphonic SAT using sequential labels and utilizes CTC to train convolutional recurrent neural networks (CRNN) [7] with learnable gated linear units (GLU) [8] to tag event sequences. As audio events often overlap with each other, the order of start and end boundaries of events are used in [6] as sequential labels. For example, the double-boundary sequential label of an audio clip might be "*dishes_start, dishes_end, speech_start, blender_start, speech_end, speech_start, blender_end, speech_end*". Sequential labels do not contain the onset and offset time information of audio events, which avoids the problem of inaccurate annotations of frame-level labels, and reduces the annotation workload. In addition to exploring the feasibility of recognizing audio event sequences in SAT, CTC-based methods have also been proposed for sound event detection (SED), which detects the type, starting time, and ending time of audio events. A bidirectional long short-term memory (LSTM) RNN [9] equipped with CTC (BLSTM-CTC) [10] is used to detect events using double-boundary sequential labels. The results [10] on a very noisy corpus show that BLSTM-CTC is able to locate boundaries of audio events with rough hints about their time positions. Apart from methods using double-boundary labels, another CTC-based SED system [11] uses single-boundary sequential labels (the sequence of start boundaries of events) with unsupervised clustering to detect the type and occurrence time of audio events. CTC redefines the loss function of RNN [5] and allows it to be trained for sequence tasks to keep order information of events in the sequence. However, CTC implicitly assumes that outputs of the network at different time steps are conditionally independent [5], which makes CTC-based approaches unable to effectively learn the contextual information inherent in audio event sequences. This paper introduces the Transformer [12], which has revolutionized the field of natural language processing [13], into SAT. The Transformer [12] does not have the conditional independence assumption in CTC. Compared with RNN-based models, the Transformer can access information at any time step from any other time step, thereby capturing long-term dependencies [14] between audio events. In addition, the training of Transformers can also be efficiently parallelized.

When learning sequence information, the decoder in the Transformer exploits past information to infer the upcoming event. For example, when recognizing audio event sequences "*fire, alarm, run*" and "*fire, crying, sobbing*", the model may be confused between *alarm* and *crying* when forward inferring the next event from *fire*. But if the target event is backward inferred from *run* and *sobbing* respectively, the probability of *alarm* and *crying* is different in different sequences. Contextual information can help the model learn the differences between sequences in detail. To make more comprehensive use of the contextual information in audio event sequences, this paper draws on the idea of bidirectional RNN [15] and proposes a contextual Transformer (cTransformer) to explore bidirectional information in audio event sequences. The cTransformer consists of an encoder and a decoder, the latter of which is the main focus of this paper. The decoder attempts to fuse frame-level representations from the encoder with the clip-level event cues to infer the target by combining the forward and backward information learned from normal and reverse sequences, respectively. Then, the loss between the prediction from the normal sequence branch and the prediction from the reverse sequence branch is calculated and back-propagated to update parameters to learn a more consistent prediction about the same target. In training, partial weights of the normal and reverse sequence branches are shared to learn forward and backward information. With the help of these shared weights, the decoder is able to learn contextual information from both directions simultaneously to more comprehensively and accurately identify event sequences.

10.21437/Interspeech.2022-196

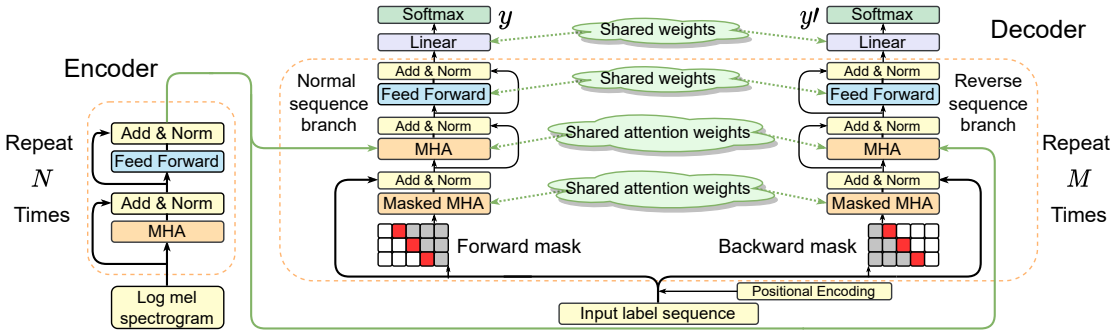

Figure 1: *The proposed contextual Transformer. In the forward and backward mask, the red, gray, and white blocks indicate the masked position of the information to be predicted, the position of the masked information, and the position of the available information.*

The contributions of this paper are: 1) we introduce the Transformer into SAT; 2) We propose the cTransformer that can utilize bidirectional information to better identify audio event sequences in audio clips; 3) To explore the feasibility of SAT based on cTransformer, we manually label sequential labels for a polyphonic audio dataset from real life, and compare the performance of the cTransformer and other CTC-based methods on it. This paper is organized as follows, Section 2 introduces the cTransformer. Section 3 describes the dataset, experimental setup, and analyzes the results. Section 4 gives conclusions.

2. Contextual Transformer

Motivated by the performance of Transformers in sequence tasks [12, 16] and the significance of contextual information in audio tasks [17, 18, 19], this paper proposes the cTransformers for audio event sequence analysis. The cTransformer aims to transform an audio clip to the corresponding sequence of event labels using both global information and rich contextual details.

2.1. Input and output definition

The most common acoustic feature for acoustic event recognition is the log mel spectrogram [20]. We convert every audio clip x into its log mel spectrogram $X(t, f)$ as the model input. Following [11], the sequence of event start boundaries is used as sequential labels. For the normal sequence branch in Figure 1, the label sequence y is "<*S*>, *event*$_1$, *event*$_2$, ..., *event*$_k$, <*E*>", where k is the number of event occurrences, and <*S*> and <*E*> are tokens indicating the start and end of prediction, respectively. For the reverse sequence branch, the label sequence $y\prime$ is "<*S'*>, *event*$_k$, *event*$_{k\text{-}1}$, ..., *event*$_1$, <*E*>", where <*S'*> is the token indicating the start of reverse sequence prediction. Note that we use different start tokens, but the same end token for the two directions. The sequential label y of an audio clip may be "<*S*>, *dishes, speech, speech, blender, speech*, <*E*>"; the corresponding $y\prime$ is "<*S'*>, *speech, blender, speech, speech, dishes*, <*E*>".

2.2. The encoder part of the contextual Transformer

The encoder aims to convert input acoustic features into high-level representations. To consider the audio information globally, this paper does not divide input features into small patches [21], so there is no positional encoding [12] in the encoder. The encoder mainly consists of N identical blocks with multi-head attention layers (MHA) and feed forward layers, which are analogous to the encoder in the Transformer [12]. The attention function in MHA is scaled dot-product attention, whose input consists of queries and keys of dimension d_k, and values of dimension d_v [12]. The attention is calculated on a set of queries, keys, and values packed into matrix $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$, respectively.

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\mathbf{Q}\mathbf{K}^{\mathbf{T}}/\sqrt{d_k})\mathbf{V} \quad (1)$$

Then, MHA is used to allow the model to jointly focus on representations from different subspaces at different positions.

$$MHA(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat(head_1, ..., head_h)\mathbf{W}^O$$
$$where \quad head_i = Attention(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \quad (2)$$

Where $head_i$ represents the output of the i-th attention head for a total number of h heads. $\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, $\mathbf{W}_i^V$ and $\mathbf{W}^O$ are learnable weights. For MHA in the encoder, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ come from the same place, at this point, the attention in MHA is called self-attention [12]. Next, a feed forward layer that consists of two linear transformations with ReLU activation function [22] in between is applied. All the parameters (such as d_k, d_v and h, etc.) follow the default settings of the original Transformer [12].

2.3. The decoder part of the contextual Transformer

The cTransformer is expected to efficiently capture contextual information in audio event sequences without reducing the Transformer's global summarization ability. The global attention in the encoder can attend to the information in all positions. However, to preserve the autoregressive property [12], the masked MHA of the decoder relies only on the forward information to sequentially predict the next event, as shown in the normal sequence branch in Figure 1. To make up for this limitation, we propose a bidirectional sequence decoder that can exploit both forward and backward information, as shown in the decoder of Figure 1. To enhance the ability of the model to capture the contextual information of the target event, the normal and reverse sequence branches predict the same target each time during training (and only the normal sequence branch is used during inference). Since some weights of the two branches are shared, these weights learn both forward and backward information about the target to help the model capture the contextual information about target events more accurately.

The decoder consists of two branches with the same structure. Each branch contains M identical blocks in series, each of which contains a masked MHA layer, an MHA layer, and a feed-forward layer, analogous to the decoder in the original Transformer [12]. The masked MHA layer uses self-attention, i.e. $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ all come from the input sequence of event labels. To preserve the autoregressive property, forward and backward masks are applied to block future and past information, respectively; masked positions in the attention map $\mathbf{Q}\mathbf{K}^T$ are replaced with $-\infty$ before the softmax operation. Next, an MHA layer is used to fuse frame-level acoustic representations from the encoder with clip-level event cues from the previous decoder layer. In this layer, $\mathbf{Q}$ comes from the previous masked MHA layer, while $\mathbf{K}$ and $\mathbf{V}$ come from the output of the encoder. When predicting the t-th target $event_t$, given the input cue for the normal sequence branch is $\overrightarrow{\mathbf{z}}_{t-1}$, and the input cue for the reverse sequence branch is $\overleftarrow{\mathbf{z}}_{t+1}$. Let $\overrightarrow{\mathbf{p}}_t$ and $\overleftarrow{\mathbf{p}}_t$ be

the prediction for $event_t$ from the normal and reverse sequence branches. For the normal sequence branch exploring forward information, $\overrightarrow{\mathbf{p}}_t$ is derived from the output of encoder $\mathbf{O}_{En}$ and $\overrightarrow{\mathbf{z}}_{t-1}$ after forward Masked MHA M_f. For the reverse sequence branch exploring backward information, $\overleftarrow{\mathbf{p}}_t$ is derived from $\mathbf{O}_{En}$ and $\overleftarrow{\mathbf{z}}_{t+1}$ after backward Masked MHA M_b.

$$\begin{aligned}\overrightarrow{\mathbf{p}}_t &= \phi(M_f(\overrightarrow{\mathbf{z}}_{t-1}) + \mathbf{O}_{En}\mathbf{W}^{(f)}_{MHA} + \mathbf{b}^{(f)}) \\ \overleftarrow{\mathbf{p}}_t &= \phi(M_b(\overleftarrow{\mathbf{z}}_{t+1}) + \mathbf{O}_{En}\mathbf{W}^{(b)}_{MHA} + \mathbf{b}^{(b)})\end{aligned} \tag{3}$$

where $\mathbf{b}^{(f)}$ and $\mathbf{b}^{(b)}$ are biases in normal and reverse sequence branches, $\mathbf{W}^{(f)}_{MHA}$ and $\mathbf{W}^{(b)}_{MHA}$ are learnable weights in MHA, ϕ denote the set of mapping functions in each branch of the decoder. The remaining layers and parameters in the decoder are the same as those of the Transformer [12].

2.4. The loss function of the contextual Transformer

The loss function is a summation of the loss over each step in the event label sequences of the training data. Denote $\overrightarrow{\mathbf{p}}_t$ and $\overleftarrow{\mathbf{p}}_t$ as p_t and $p\prime_t$. At each step, p_t and $p\prime_t$ are the distribution of labels predicted by the normal and reverse sequence branches, while the ground-truth labels are y_t and $y\prime_t$. Following the loss function in the original Transformer [12], the cross entropy (CE) loss is computed for both branches:

$$\mathcal{L}_{\text{normal}} = CE(p_t, y_t), \quad \mathcal{L}_{\text{reverse}} = CE(p\prime_t, y\prime_t) \tag{4}$$

Since p_t and $p\prime_t$ are the prediction for the same target, the mean squared error (MSE) loss that performs well in regression tasks [23][24][25] is used as the context loss to measure the distance between p_t and $p\prime_t$ in the latent space.

$$\mathcal{L}_{\text{context}} = MSE(p\prime_t, p_t) \tag{5}$$

To consider both forward and backward information at the same time in the training, the different types of losses of different branches are added together. The final loss of the model is

$$\mathcal{L} = \lambda_n \mathcal{L}_{\text{normal}} + \lambda_r \mathcal{L}_{\text{reverse}} + \lambda_c \mathcal{L}_{\text{context}} \tag{6}$$

where λ adjusts the weights of loss components during training. Each λ defaults to 1. During the training process, the forward prediction p_t and backward prediction $p\prime_t$ will be aligned to capture the rich contextual information around the target event and learn the entire sequence embeddings more accurately.

3. Experiments and results

3.1. Dataset, Baseline, Experiments Setup, and Metrics

Since there is no publicly available polyphonic audio dataset with sequential labels, we manually label the DCASE domestic environment audio dataset [26] with the sequences of event start boundaries as sequential labels following [11]. The audio dataset excerpted from Audioset [27] contains 10 classes of real-life audio events, where the training and test sets consist of 1578 and 288 audio clips, respectively. The number of event occurrences contained in the training and test sets is 3619 and 923. During training, 20% of the training samples are randomly selected and set aside as the validation set.

Since most previous works on audio event sequence analysis rely on CTC, the BLSTM-CTC [10] is used as Baseline in this paper. This paper also compares the cTransformer with CTC-based convolutional bidirectional gated recurrent units (CBGRU-CTC) [28], and CBGRU-CTC equipped with GLU in convolutional layers (CGLU-BGRU-CTC) [6], and in both convolutional and recurrent layers (CBGRU-GLU-CTC) [11].

As the input feature, log mel-band energy with 64 banks [20] is extracted using STFT with Hamming window length of 46 ms and the overlap is $1/3$ between windows, following the settings of [29]. During training, stochastic gradient descent with momentum (SGDM) [30] with an initial learning rate of 1e-3, a batch size of 64, and a momentum value of 0.9 is used to minimize the loss. Dropout [31] and layer normalization [32] are used to prevent overfitting. Systems are trained on a single card Tesla V100-SXM2-32GB for a maximum of 1000 epochs. For more details and the manually labeled dataset with sequential labels, please visit the project homepage (https://github.com/Yuanbo2020/Contextual-Transformer).

The output of SAT consists of the types of audio events plus the order information between them. This paper uses precision (*P*), recall (*R*), F-score (*F*), accuracy (*Acc*) [33], and area under curve (*AUC*) [34] to measure various aspects of the models' performance in recognizing the types of audio events, and adopts the bilingual evaluation understudy (*BLEU*) [35] metric commonly used in sequence tasks to evaluate the models' ability to recognize the order between events. Higher *P*, *R*, *F*, *Acc*, *AUC*, and *BLEU* indicate better performance.

3.2. Results and Analysis

Number of encoder and decoder blocks. The encoder and decoder of the cTransformer consist of N and M identical blocks, respectively. This paper first explores the optimal ratio of blocks of encoder and decoder to determine the final model structure, as shown in Table 1. We choose AUC to represent the models' AT performance, since AUC does not depend on the threshold.

In Table 1, the performance of the model does not increase monotonically with the number of blocks. The best results on the test set are achieved with $\{N, M\} = \{1, 2\}$, which are significantly smaller than the default setting $\{N, M\} = \{6, 6\}$ of the original Transformer [12]. The reason why our best model is smaller may be that the polyphonic audio dataset we have manually labeled is not large enough. In experiments, we have observed serious overfitting as the values of N and M get large. So $\{N, M\} = \{1, 2\}$ will be used in the following experiments.

Weighting factors in the loss. The next step is to optimize the weighting factors λ of different loss terms. The $\mathcal{L}_{\text{normal}}$ and $\mathcal{L}_{\text{reverse}}$ focus on learning task-goal-oriented representations to improve the accuracy of individual event sequence recognition, while the $\mathcal{L}_{\text{context}}$ aims to align the predictions of the normal and reverse sequence branches to make them more consistent.

Table 2 presents an ablation study to demonstrate the necessity of each term in cTransformer's loss function. Model #1 has only the normal sequence branch, and is equivalent to the original Transformer [12]. Conversely, Model #2 has only the reverse sequence branch. Except for the result of # 2 from the reverse sequence branch, the rest are predicted by the normal sequence branch. Model #3, which has both the normal and the reverse sequence branches, outperforms #1 and #2 which only

Table 1: *Performance of the cTransformer with different numbers of encoder and decoder blocks.*

#	$\{N, M\}$	*AUC*	*BLEU*	#	$\{N, M\}$	*AUC*	*BLEU*
1	$\{1, 1\}$	0.771	0.468	7	$\{3, 3\}$	0.784	0.482
2	$\{1, 2\}$	**0.800**	**0.491**	8	$\{3, 6\}$	0.770	0.472
3	$\{2, 2\}$	0.775	0.481	9	$\{4, 2\}$	0.779	0.467
4	$\{2, 4\}$	0.775	0.483	10	$\{4, 4\}$	0.787	0.464
5	$\{2, 5\}$	0.783	0.473	11	$\{5, 5\}$	0.774	0.461
6	$\{3, 1\}$	0.782	0.474	12	$\{6, 6\}$	0.778	0.456

Table 2: *Ablation experiments of the cTransformer on test set.*

#	$\mathcal{L}_{\text{normal}}$	$\mathcal{L}_{\text{reverse}}$	$\mathcal{L}_{\text{context}}$	*F (%)*	*Acc (%)*	*AUC*	*BLEU*
1	✔	✗	✗	66.42	90.41	0.780	0.474
2	✗	✔	✗	64.58	89.79	0.765	0.472
3	✔	✔	✗	67.39	90.66	0.785	0.489
4	✔	✔	✔	**70.42**	**91.63**	**0.800**	**0.491**

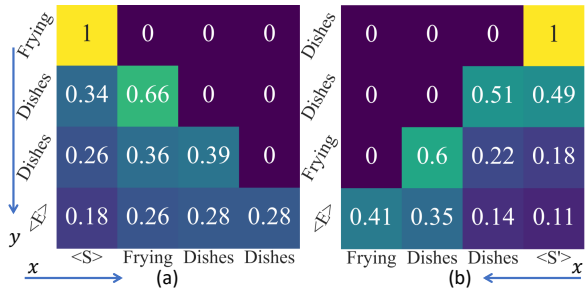

Figure 2: *Attention score from the masked MHA in decoder. Subgraph (a) and (b) are from the normal and reverse sequence branches, respectively. The x-axis is each event predicted in an autoregressive way, y-axis is the corresponding reference event.*

have a single branch. Model #4, which is further equipped with the context loss, performs the best of all. By encouraging both branches to yield consistent predictions, the model is better able to integrate the information contained in both directions to more accurately identify and effectively confirm the target event.

Table 3 further tunes the weights of the different loss terms in a fine-grained manner to find out the optimal combination of coefficients. Table 3 attempts to control variables to compare the performance of models with different combinations of coefficients. Finally, giving the same weight to $\mathcal{L}_{\text{normal}}$ and $\mathcal{L}_{\text{context}}$, and lightening the weight of $\mathcal{L}_{\text{reverse}}$ achieves the best AUC in # 4. This reveals that in the experiments, the cTransformer should focus on capturing the forward and contextual information, while putting the backward information in a secondary position for better event sequence recognition.

Comparison with prior methods. In Table 4, the cTransformer is compared with prior methods for recognizing audio event sequences. All models are retrained on the DCASE dataset with manually labeled event sequences. Since previous works using CTC-based methods did not apply data augmentation, we do not apply data augmentation for a fair comparison. To analyze the ability of different models to identify polyphonic audio events from multiple perspectives, multiple metrics are adopted for AT and the *BLEU* metric is used for SAT.

As shown in Table 4, BLSTM-CTC [10], which uses only LSTM to extract acoustic representations for recognizing polyphonic audio event sequences, has the worst overall performance. The CBGRU-CTC [28] with a composite convolutional recurrent neural network outperforms the BLSTM-CTC on most metrics, which demonstrates the superior ability of convolutional layers in feature extraction. CGLU-BGRU-CTC [6] and CBGRU-GLU-CTC [11] with GLU assembled in convolutional layers and both convolutional and recurrent layers, respectively, do not bring further overall improvement, although they do outperform CBGRU-CTC in some metrics. Table 4 also shows the results of the original Transformer [12] with a 6-layer encoder and decoder. Possibly due to the limited size of the polyphonic audio dataset, the performance of Transformer is close to that of the CTC-based methods. Overall, the cTransformer achieves the best results in both AT and SAT.

Table 3: *Performance of cTransformer, varying the loss weights.*

#	λ_n	λ_r	λ_c	*AUC*	*BLEU*	#	λ_n	λ_r	λ_c	*AUC*	*BLEU*
1	1	0.5	0.1	0.789	0.481	8	0.5	1	1	0.774	0.467
2	1	0.5	0.25	0.803	**0.511**	9	1	1	0.1	0.791	0.485
3	1	0.5	0.5	0.782	0.488	10	1	1	0.25	0.788	0.501
4	1	0.5	1	**0.805**	0.505	11	1	1	0.5	0.783	0.487
5	0.5	1	0.1	0.784	0.479	12	0.1	0.1	1	0.763	0.465
6	0.5	1	0.25	0.788	0.482	13	0.25	0.25	1	0.774	0.466
7	0.5	1	0.5	0.778	0.465	14	0.5	0.5	1	0.785	0.472

Case study. To gain a more intuitive insight into the performance of the model on polyphonic audio event sequences, we conduct a case study on an audio clip where the ground-truth sequence of event start boundaries is "*frying, dishes, dishes*". Figure 2 shows the distribution of attention scores from the masked MHA of the normal and reserve sequence branches. In Figure 2 (a), after inputting <*S*>, the attention value for <*S*> is 1, then combining acoustic representations from the encoder, the model predicts the next event should be *frying* (the event corresponding to the 2nd column of the x-axis). The reference event label is *frying* (the event corresponding to the 1st row of the y-axis). Then, the input is "<*S*>, *frying*", attention values for the two events are 0.34 and 0.66, respectively. The next event is predicted to be *dishes* (the event corresponding to the 3rd column of the x-axis). The reference label is *dishes* (the event corresponding to the 2nd row of the y-axis). Finally, when the input is "<*S*>, *frying, dishes, dishes*", based on acoustic representations, the model judges that the event sequence is complete, and subsequently outputs <*E*> (the event corresponding to the 4th row of the y-axis) to indicate the inference stops. After the autoregressive process, the predicted event sequence p = "*frying, dishes, dishes*" is obtained, the reference label sequence y is "*frying, dishes, dishes*". The match between p and y indicates that the cTransformer successfully fuses frame-level acoustic representations from the encoder with clip-level event cues from the decoder to jointly infer the event sequence. In Figure 2 (b), attention scores from reverse sequence branch for the same audio clip are different from attention scores for forward inference in Figure 2 (a). Guided by <*S'*>, the reverse sequence branch combining audio representations successfully predicts the reverse sequence $p\prime$ = "*dishes, dishes, frying*", the corresponding label $y\prime$ is "*dishes, dishes, frying*". The match of $p\prime$ and $y\prime$ indicates that with the assistance of different prediction cues and mask matrices, the cTransformer effectively infers the event sequence from normal and reverse directions, which implies that the model is effective for modeling contextual information.

4. Conclusions

This paper first introduces the Transformer into SAT. To utilize the context information of audio event sequences, this paper proposes cTransformer with a bidirectional sequence decoder, which can exploit both forward and backward information. The cTransformer automatically assigns different attention scores to available information to effectively model contextual information and infer the event, and then efficiently fuses frame-level acoustic representations and clip-level event cues to identify event sequences implicit in audio clips. Future work will explore the performance of cTransformer using fully bidirectional information to infer audio event sequences on more datasets.

5. ACKNOWLEDGEMENTS

The WAVES Research Group and Bo Kang received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

Table 4: *Comparison of AT and SAT results with prior works on recognizing audio event sequences.*

Method	AT					SAT
	P (%)	*R (%)*	*F (%)*	*Acc (%)*	*AUC*	*BLEU*
BLSTM-CTC [10]	69.73	50.12	58.32	89.47	0.713	0.323
CBGRU-CTC [28]	67.79	63.39	63.23	90.93	0.793	0.475
CGLU-BGRU-CTC [6]	**79.87**	60.99	69.17	90.48	0.786	0.468
CBGRU-GLU-CTC [11]	75.97	64.30	69.65	91.77	0.787	0.463
Transformer [12]	67.24	64.53	65.86	90.17	0.785	0.432
cTransformer	75.66	**67.61**	**71.41**	**92.05**	**0.805**	**0.505**

6. References

[1] A. Mesaros, T. Heittola, E. Benetos, P. Foster, M. Lagrange, T. Virtanen, and M. D. Plumbley, "Detection and classification of acoustic scenes and events: Outcome of the dcase 2016 challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 2, pp. 379–393, 2017.

[2] K. Chen, Y. Wu, Z. Wang, X. Zhang, F. Nian, S. Li, and X. Shao, "Audio captioning based on transformer and pretrained cnn," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 21–25.

[3] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational analysis of sound scenes and events*. Springer, 2018.

[4] D. P. Ellis, "Prediction-driven computational auditory scene analysis," Ph.D. dissertation, Columbia University, 1996.

[5] A. Graves and F. Gomez, "Connectionist temporal classification:labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006, pp. 369–376.

[6] Y. Hou, Q. Kong, J. Wang, and S. Li, "Polyphonic audio tagging with sequentially labelled data using crnn with learnable gated linear units," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2018, pp. 78–82.

[7] E. Cakır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 2017.

[8] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, 2017, pp. 933–941.

[9] Y. Xu, Q. Kong, Q. Huang, W. Wang, and M. D. Plumbley, "Convolutional gated recurrent neural network incorporating spatial features for audio tagging," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3461–3466.

[10] Y. Wang and F. Metze, "A first attempt at polyphonic sound event detection using connectionist temporal classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2986–2990.

[11] Y. Hou, Q. Kong, S. Li, and M. D. Plumbley, "Sound event detection with sequentially labelled data based on connectionist temporal classification and unsupervised clustering," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 46–50.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[13] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 2020, pp. 38–45.

[14] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y. X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[15] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[16] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[17] X. Favory, K. Drossos, T. Virtanen, and X. Serra, "Learning contextual tag embeddings for cross-modal alignment of audio and tags," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 596–600.

[18] Y. Xu, Q. Huang, W. Wang, P. Foster, S. Sigtia, P. J. Jackson, and M. D. Plumbley, "Unsupervised feature learning based on deep models for environmental audio tagging," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1230–1241, 2017.

[19] X. L. Zhang and D. Wang, "Boosting contextual information for deep neural network based voice activity detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 2, pp. 252–264, 2015.

[20] A. Bala, A. Kumar, and N. Birla, "Voice command recognition system based on mfcc and dtw," *International Journal of Engineering Science and Technology*, vol. 2, no. 12, pp. 7335–7342, 2010.

[21] Y. Gong, Y. A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," in *Proc. of INTERSPEECH*, 2021, pp. 571–575.

[22] K. Eckle and J. Schmidt-Hieber, "A comparison of deep networks with relu activation function and linear spline-type methods," *Neural Networks*, vol. 110, pp. 232–242, 2019.

[23] T. Kim, J. Oh, N. Kim, S. Cho, and S.-Y. Yun, "Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation," in *Proc. of IJCAI*, 2021, pp. 2628–2635.

[24] A. Berg, M. Oskarsson, and M. O'Connor, "Deep ordinal regression with label diversity," in *25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 2740–2747.

[25] H. Phan, L. Pham, P. Koch, N. Q. Duong, I. McLoughlin, and A. Mertins, "On multitask loss function for audio event detection and localization," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 160–164.

[26] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. P. Shah, "Large-scale weakly labeled semi-supervised sound event detection in domestic environments," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2018, pp. 19–23.

[27] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[28] Y. Hou, Q. Kong, and S. Li, "Audio tagging with connectionist temporal classification model using sequentially labelled data," in *International Conference in Communications, Signal Processing, and Systems*, 2018, pp. 955–964.

[29] Q. Kong, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "DCASE 2018 challenge surrey cross-task convolutional neural network baseline," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2018, pp. 217–221.

[30] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[31] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[32] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, "On layer normalization in the transformer architecture," in *International Conference on Machine Learning*, 2020, pp. 10 524–10 533.

[33] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[34] J. Huang and C. X. Ling, "Using auc and accuracy in evaluating learning algorithms," *IEEE Transactions on knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005.

[35] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.