GCT: GATED CONTEXTUAL TRANSFORMER FOR SEQUENTIAL AUDIO TAGGING

Yuanbo Hou[1], Yun Wang[2], Wenwu Wang[3] and Dick Botteldooren[1]

[1]WAVES, Ghent University, Belgium [2]Meta AI, USA [3]CVSSP, University of Surrey, UK

1. Introduction

Sequential audio tagging (SAT) means detecting both the class information of audio events, and the order in which they occur within the audio clip. To exploit both forward and backward information of events for SAT tasks, this paper proposes a gated contextual Transformer (GCT) with forward-backward inference.

2. GATED CONTEXTUAL TRANSFORMER (GCT)

2.1. Encoder and Decoder of GCT

Encoder:

There are two ways for the input:
1) the entire spectrogram of the audio clip;
2) the patch sequence by dividing the spectrogram clip into Patches.

Decoder:

With the combined effect of forward and backward mask matrices, the normal and reverse sequence branches will infer the same target at each time step.

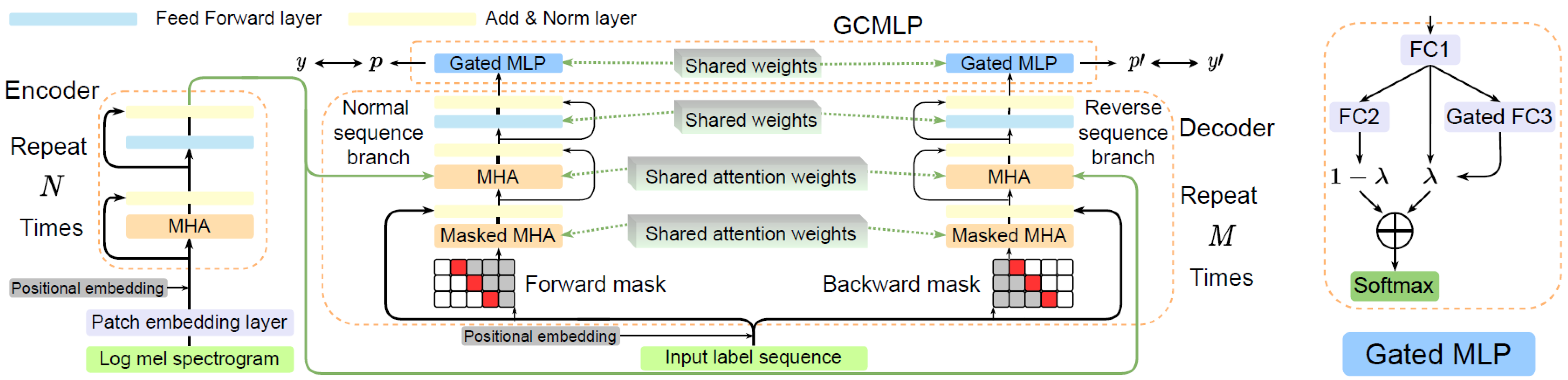

Figure 1: The proposed gated contextual Transformer. In the mask matrices, the red, gray, and white blocks present the positions corresponding to the target to be predicted, the positions of masked data, and the positions of available data.

2.2. Gated contextual multi-layer perceptron (GCMLP)

GCMLP aims to perform the final conditioning of the decoder output based on the gated MLP (gMLP) block and shared weights while considering the contextual information about the target to achieve more accurate predictions.

$$gMLP = Softmax((1-\lambda) \odot F_2 + \lambda \odot F_1)$$

2.3. Forward-backward inference

Algorithm 1 PyTorch pseudo code for the proposed FBI

```
# X: input log mel spectrogram; X′: X reversed along the time axis
E, E′ = Encoder(X), Encoder(X′)                          # output of encoder
I, I′ = <S>, <S′>              # start token of the normal and reverse sequence
for k in range(L − 1):     # L: max length of event sequences; B: batch size
    D = Decoder_normal_branch(E, I)         # D: (B, L, number of tokens)
    p = GCMLP(D[:, -1, :])           # pick the latest target probability vector

    D′ = Decoder_reverse_branch(E′, I′)
    p′ = GCMLP(D′[:, -1, :])     # p′ and p are the same target's predictions

    p_ci = αp + (1 − α)p′  # p_ci: final prediction with contextual informa-
    mation; α: importance factor of the forward information, default to 0.5.
    _, p_et = torch.max(p_ci, dim=1).item()       # p_et: predicted event token
    if p_et == <E>: break                  # <E>: end token of event sequences

    I = torch.cat([I, torch.ones(1, 1).fill_(p_et)], dim=1)
    I′ = torch.cat([I′, torch.ones(1, 1).fill_(p_et)], dim=1)
```

3. Results and analysis

Model structure.

Table 1: AUC of different input modes of GCT with different numbers of encoder and decoder blocks on the *Noiseme* dataset.

#	N	M	Patches	Clip	#	N	M	Patches	Clip
1	1	2	0.575±0.010	0.647±0.012	7	8	6	0.534±0.033	0.557±0.058
2	2	4	0.584±0.009	0.661±0.016	8	8	8	0.614±0.020	0.518±0.063
3	4	4	0.600±0.018	**0.662±0.013**	9	9	5	0.609±0.026	0.512±0.017
4	5	5	0.599±0.046	0.660±0.071	10	9	7	0.604±0.066	0.511±0.013
5	6	6	0.609±0.017	0.596±0.075	11	9	9	0.608±0.027	0.511±0.007
6	7	7	**0.627±0.019**	0.543±0.024	12	10	10	0.606±0.052	0.508±0.032

Ablation study.

Pos emb (#2) slightly outperforms GCMLP (#3). This reveals that when the input is small patches, the position information is valuable for the model to effectively capture the local information of events.

Table 2: Ablation study of GCT {7, 7} component on *Noiseme*.

#	Pos_emb	GCMLP	AT: Acc (%)	AT: AUC	SAT: BLEU
1	✗	✗	92.23±0.61	0.600±0.014	0.297±0.045
2	✓	✗	93.00±0.54	0.616±0.012	0.312±0.019
3	✗	✓	92.55±0.62	0.610±0.009	0.309±0.023
4	✓	✓	**93.21±0.27**	**0.627±0.019**	**0.338±0.012**

FBI plays a more powerful role when coarse-grained clips are input. The reason may be that after the spectrogram is split into patches, the time interval between forward and reverse information is shortened in each patch, equivalent to reducing the range of context that FBI can capture.

Table 3: Ablation study of the inference method on *Noiseme*.

	FBI	Patches	Clip		FBI	Patches	Clip
Acc (%)	✗	93.21±0.27	93.49±0.39	AUC	✗	0.627±0.019	0.662±0.013
	✓	93.57±0.46	**94.01±0.31**		✓	0.635±0.014	**0.685±0.022**

Pretrained weight.

#5 outperforms #4, indicating that the encoder with the ability in acoustic feature extraction is more important than Pos_emb in providing the position information of patches.

Table 4: Effect of transfer learning on GCT on *DCASE*.

#	Pos_emb	Encoder	AT: Acc (%)	SAT: BLEU
1	No Transfer		89.13±0.58	0.435±0.037
2	Fixed	Fixed	**97.68±0.18**	**0.677±0.014**
3	Fine-tuned	Fine-tuned	96.27±0.36	0.645±0.019
4	Fixed	Fine-tuned	93.84±0.85	0.639±0.016
5	Fine-tuned	Fixed	96.45±0.47	0.662±0.015

The fixed mode (#2) is better than fine-tuning the transferred parameters (#3). The reason may be that the part (Pos emb and encoder) containing pretrained weights and the remaining randomly initialized part (decoder and GCMLP) differ greatly in the latent space, finetuning these two disparate parts using the same learning rate will inevitably affect the performance of (Pos emb and encoder) with audio events expertise.

Case study.

The inferred sequences match the corresponding labels consistently, which means that GCT is good at exploiting event context to identify event sequences.

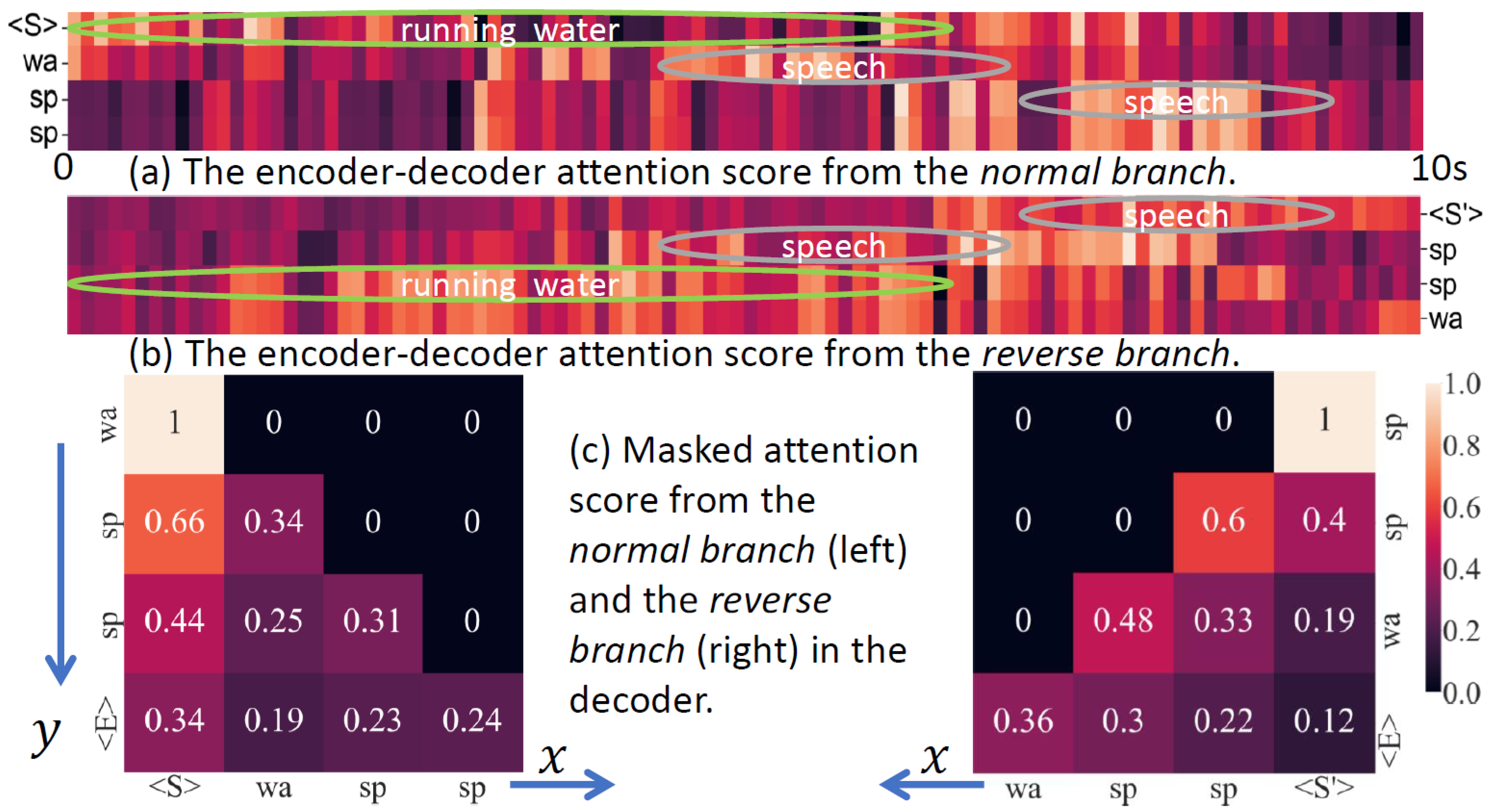

Figure 2: Attention in GCT.
In subgraph (c), the x-axis is each event predicted in an autoregressive way, the y-axis is the reference event.

4. Conclusion

To improve cTransformer in structure and inference, we propose a gated contextual Transformer (GCT) with GCMLP and FBI for SAT.