INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

# Towards Effective and Compact Contextual Representation for Conformer Transducer Speech Recognition Systems

*Mingyu Cui*[1], *Jiawen Kang*[1], *Jiajun Deng*[1], *Xi Yin*[3], *Yutao Xie*[3], *Xie Chen*[2,3], *Xunying Liu*[1]

[1]The Chinese University of Hong Kong, Hong Kong SAR, China
[2] MoE Key Lab of Artificial Intelligence, AI Institute,
X-LANCE Lab, Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, China
[3] Peng Cheng Laboratory, Shenzhen, China

## Abstract

Current ASR systems are mainly trained and evaluated at the utterance level. Long range cross utterance context can be incorporated. A key task is to derive a suitable compact representation of the most relevant history contexts. In contrast to previous researches based on either LSTM-RNN encoded histories that attenuate the information from longer range contexts, or frame level concatenation of transformer context embeddings, in this paper compact low-dimensional cross utterance contextual features are learned in the Conformer-Transducer Encoder using specially designed attention pooling layers that are applied over efficiently cached preceding utterances' history vectors. Experiments on the 1000-hr Gigaspeech corpus demonstrate that the proposed contextualized streaming Conformer-Transducers outperform the baseline using utterance internal context only with statistically significant WER reductions of 0.7% to 0.5% absolute (4.3% to 3.1% relative) on the dev and test data.

**Index Terms**: Speech Recognition, Conformer-Transducer, Contextual Representation

## 1. Introduction

End-to-end (E2E) automatic speech recognition (ASR) technologies have achieved great success in recent years. A series of representative models, such as connectionist temporal classification (CTC) [1, 2], listen-attend-spell (LAS) [3], transformer [4–6], convolution-augmented transformer (Conformer) [7, 8] and recurrent neural network transducer (RNN-T) [9–12] have been developed. Among these, transformer based models, in particular those based on Conformer Encoder architectures [13–16], have demonstrated performance improvements over RNN based models.

It is well known that context plays an important role in human communication. A rich taxonomy of contextual cues across neighbouring speech utterances at acoustic-phonetic, prosodic, lexical, semantic and discourse level are used to determine what is likely to be said in a conversation. However, the majority of current ASR systems are trained and evaluated at the utterance level. To this end, the incorporation of long range, cross utterance contexts in E2E ASR systems provides a powerful solution. For this reason, contextual ASR models are attracting increasing research interest [17–23]. For example, the benefit from incorporating such cross-utterance information has been widely demonstrated on language modelling [24–32]. In contrast, limited prior researches in this direction have been conducted for Transformer or Conformer models [33–37].

A key task in modelling cross utterance contexts for E2E ASR systems in general, including Transformers and Conformers, is to derive a suitable representation of the most relevant portion of history contexts to improve the prediction of current outputs, while incurring minimal computational overhead. In LSTM-RNN based neural transducers [22], the recurrent hidden vectors are used to encode preceding utterance histories, while attenuating the contribution from longer range contexts with an undesirable diminishing effect.

Although the convolutional and attention mechanisms used in the Conformer architectures can capture both local and global feature patterns within a single utterance, there has no principled and well established solution when using these to model cross utterance contexts. A common practice [36, 37] is to utilize the outputs of Transformer or Conformer in each frame time step before being concatenated and serving as the long span context representation. However, there is a lack of mechanistic approaches to locate the most relevant portion of history contexts over time. In addition, this leads to computational efficiency and latency issues due to the high dimensionality of the frame level concatenated contextual representation that is dependent on the lengths of preceding utterances.

To this end, compact low-dimensional cross utterance contextual features are learned in the Encoder module of Conformer-Transducer (C-T) [7, 38] ASR systems in this paper using specially designed attention pooling layers applied over preceding utterances' history vectors to auto-configure the context weighting at different time steps. Similar to the attentive pooling [39] and attentive speaker embedding [40] used in speaker recognition [41] tasks, such attention based pooling compresses variable length hidden states into fixed length context representations. Inspired by Transformer-XL [27] language models, the preceding utterances' history vectors are efficiently cached prior to the pooling to improve computational efficiency. Cross utterance contexts are also incorporated into the Predictor. Experiments on the 1000-hr Gigaspeech corpus demonstrate that the proposed cross utterance context conditioned Conformer-Transducer system outperforms the baseline using utterance internal context only with statistically significant word error rate (WER) reductions of 0.7% to 0.5% absolute (4.3% to 3.1% relative) on the dev and test data, while incurring a moderate processing latency increase by 7.5% during cross utterance context fusion.

The main contributions of this paper are summarized as follows. First, to the best of our knowledge, this is the first work to efficiently model attention pooling compressed low-dimensional cross utterance contexts in Conformer-Transducer systems. In contrast, related prior researches focused on other architectures based on, for example, Transformer and Conformer-transformer AED [33, 35–37] models. Second, the compact low-dimensional cross utterance contextual features of in this paper enhances the computational efficiency and practical deployment of Conformer-Transducer and other Transformer Encoder based E2E systems. Their benefit from cross utterance context modelling can be exploited to process stream-


10.21437/Interspeech.2023-552

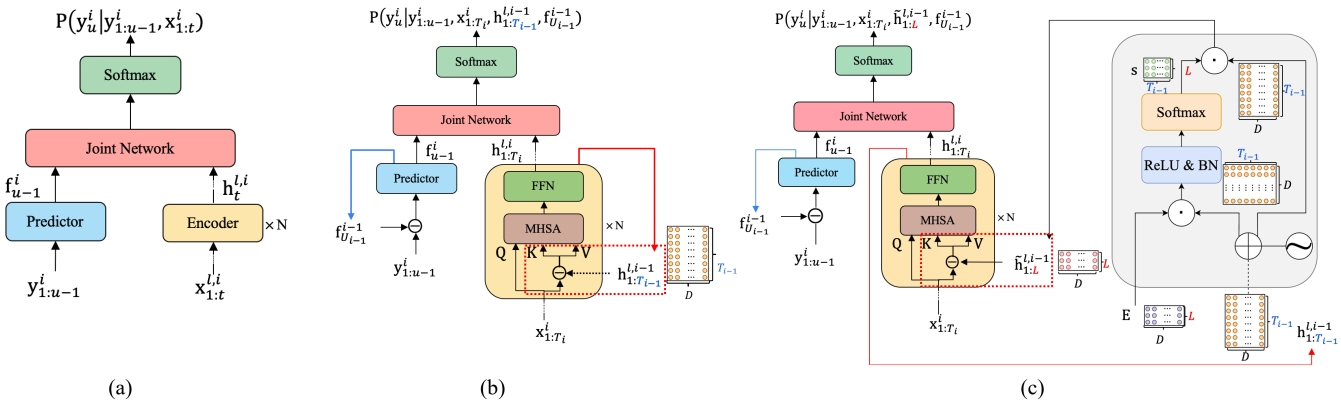


Figure 1: *Examples of: a) Standard Conformer-Transducer models using utterance internal context only; (b) C-T models using cross utterance context of the most recent preceding $(i-1)^{\text{th}}$ utterance $\mathbf{x}^{i-1}_{1:T_{i-1}}$ of $T_{i-1}$ frames in both the Encoder (connected via red line) and Predictor (via blue line encoding $(u-1)$ history words of current $i^{\text{th}}$ utterance $\mathbf{x}^{i}_{1:T_i}$), where frame level concatenated Encoder contextual matrix $\mathrm{h}^{l,i-1}_{1:T_{i-1}} \in \mathbb{R}^{T_{i-1}\times D}$ are used; and (c) C-T models using the most recent preceding utterance's context in both Encoder and Predictor, where low-dimensional Encoder contextual representations $\tilde{\mathrm{h}}^{l,i-1}_{1:L} \in \mathbb{R}^{L\times D}, L \ll T_{i-1}$ are compressed using attention pooling (circled in grey box, right) and $D$ is the Encoder output dimensionality. Grey dotted connections only used during forward passes in training. $\ominus$, $\odot$ and $\oplus$ denote matrix concatenation, multiplication and addition respectively. Encoder fusion of current and previous utterances' contexts is indicated in the red dotted box.*

ing conversational data in naturalistic application scenarios.

## 2. Conformer Transducer ASR Architecture

### 2.1. Neural Transducer

In this paper, the neural transducer [9] model is adopted for speech recognition. The neural transducer consists of three components, which are audio "Encoder", text "Predictor" and "Joint Network" modules respectively, as shown in Figure 1(a). Here we denote $\mathbf{x}^i_{1:T_i} = [\mathbf{x}^i_1, \mathbf{x}^i_2 ..., \mathbf{x}^i_{T_i}]$ as the $i$-th utterance of an audio clip or conversation session with $T_i$-frames and $\mathbf{y}^i_{1:U_i} = [\mathbf{y}^i_1, \mathbf{y}^i_2 ..., \mathbf{y}^i_{U_i}]$ as the corresponding label of length $U_i$. The acoustic feature sequence $\mathbf{x}^i_{1:t}$ is fed into Encoder to produce the acoustic representation $\mathbf{h}^i_t$. The history output labels $\mathbf{y}^i_{1:u-1}$ are fed into the Predictor module to generate the text representation $\mathbf{f}^i_{u-1}$. The outputs of Encoder and Predictor are then combined in the Joint Network via a non-linear function such as ReLU to obtain the hidden state $\mathbf{g}^i_{t,u-1}$ at time step $t$ with output history $\mathbf{y}^i_{1:u-1}$. These operations are as follows,

$$\begin{aligned}
\mathbf{h}^i_t &= \text{Encoder}(\mathbf{x}^i_{1:t}) \\
\mathbf{f}^i_{u-1} &= \text{Predictor}(\mathbf{y}^i_{1:u-1}) \\
\mathbf{g}^i_{t,u-1} &= \text{relu}(\mathbf{h}^i_t + \mathbf{f}^i_{u-1}) \\
P(\mathbf{y}^i_u|\mathbf{y}^i_{1:u-1}, \mathbf{x}^i_{1:t}) &= \text{softmax}(\mathbf{W}_o * \mathbf{g}^i_{t,u-1})
\end{aligned} \tag{1}$$

where $\mathbf{W}_o$ is a linear transformation applied prior to the final Softmax output layer. Among existing neural transducer systems, RNN or LSTM [9, 22] and Transformer [19, 38, 42] architectures have been used for the Encoder, while the Predictor module is commonly based on LSTM. In this paper, Conformer-Transducers (C-T) designed using Conformer based Encoder and LSTM Predictor modules are used throughout this paper. An example Conformer-Transducer model using utterance internal context only is shown in Fig. 1(a).

### 2.2. Conformer Transducer

More specifically, the Conformer based Encoder is based on a multi-block stacked architecture. Each block contains the following components in turn: a position wise feed-forward (FFN) module, a multi-head self-attention (MHSA) module, a convolution (CONV) module and a final FFN module at the end. Among these, the CONV module consists of several modules: a 1-D pointwise convolution layer, a gated linear units (GLU) activation [43], a second 1-D point- wise convolution layer followed by a 1-D depth-wise convolution layer, a Swish activation and a final 1-D pointwise convolution layer. Layer normalization (LN) and residual connections are applied to stabilize the training and allow more stacked layers. For a given input feature sub-sequence $\mathbf{x}^i_{1:t}$ at time step $t$ fed into a Conformer Encoder, the vector output $\mathbf{h}^{l,i}_t$ of Conformer Encoder $l$-th block is:

$$\begin{aligned}
\hat{\mathbf{x}}^{l,i,(0)}_t &= \mathbf{x}^{l-1,i}_{1:t} + \frac{1}{2}\text{FFN}(\mathbf{x}^{l-1,i}_{1:t}) \\
\mathbf{q}^{l,i}_t, \mathbf{k}^{l,i}_t, \mathbf{v}^{l,i}_t &= \hat{\mathbf{x}}^{l,i,(0)}_t \mathbf{W}_q, \hat{\mathbf{x}}^{l,i,(0)}_t \mathbf{W}_k, \hat{\mathbf{x}}^{l,i,(0)}_t \mathbf{W}_v \\
\hat{\mathbf{x}}^{l,i,(1)}_t &= \hat{\mathbf{x}}^{l,i,(0)}_t + \text{MHSA}(\mathbf{q}^{l,i}_t, \mathbf{k}^{l,i}_t, \mathbf{v}^{l,i}_t) \\
\hat{\mathbf{x}}^{l,i,(2)}_t &= \hat{\mathbf{x}}^{l,i,(1)}_t + \text{Conv}(\hat{\mathbf{x}}^{l,i,(1)}_t) \\
\mathbf{h}^{l,i}_t &= \text{Layernorm}(\hat{\mathbf{x}}^{l,i,(2)}_t + \frac{1}{2}\text{FFN}(\hat{\mathbf{x}}^{l,i,(2)}_t))
\end{aligned} \tag{2}$$

where $\mathbf{W}_q$, $\mathbf{W}_k$ and $\mathbf{W}_v$ are the linear transformations to generate the query, key and value, respectively.

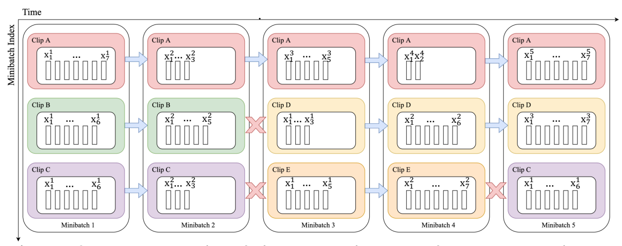


Figure 2: *An example of data serialization for contextual C-T system training. The batch size is 3 and the utterances are from five audio clips A to E. Blue arrows indicate cross utterance contextual information between clip internal utterances is used, while red cross marks indicate it is not used when start processing the 1st utterance of a new clip.*

## 3. Compact Contextual Representation

In this section, we propose cross utterance context conditioned Conformer-Transducer models with specially designed low-dimensional attention pooling layers to extract compact preceding utterances' Encoder contextual representations.

### 3.1. Encoder Contextual Representation

Although the convolutional and attention mechanisms adopted in the Conformer architectures are used to capture both local and global feature patterns within a single utterance, there has no well-established solution when using them to model cross utterance contexts. A common practice [27,36,37] is to utilize the outputs of Transformer or Conformer obtained at each frame of the preceding utterance(s). These are then being concatenated and serving as the long span context representation to augment the current utterance's input features before applying the linear transformations to produce the query, key and value vectors. In order to incorporate such frame level concatenated cross utterance Encoder contextual representations, the utterance internal context based C-T model of Eqn. (2) are now modified as

$$\begin{aligned}\hat{\mathbf{x}}^{l,i}_{1:T_i} &= \mathbf{x}^{l-1,i}_{1:T_i} + \frac{1}{2}\mathrm{FFN}(\mathbf{x}^{l-1,i}_{1:T_i})\\ \hat{\mathbf{h}}^{l,i}_{1:T_i} &= \hat{\mathbf{x}}^{l,i}_{1:T_i} \ominus \mathrm{SG}(\mathbf{h}^{l,i-1}_{1:T_i})\\ \mathbf{q}^{l,i}, \mathbf{k}^{l,i}, \mathbf{v}^{l,i} &= \hat{\mathbf{x}}^{l,i}_{1:T_i}\mathbf{W}_q, \hat{\mathbf{h}}^{l,i}_{1:T_i}\mathbf{W}_k, \hat{\mathbf{h}}^{l,i}_{1:T_i}\mathbf{W}_v\end{aligned} \tag{3}$$

where $\mathrm{SG}(\cdot)$ stands for the "stop gradient" operator, $\ominus$ denotes matrix concatenation. An example of C-T models using such frame level concatenated preceding utterances' Encoder contextual features are shown in Fig. 1(b).

### 3.2. Compact Encoder Contextual Representation

The frame-level cross utterance Encoder contextual representations of Eqn.(3) above do not provide a mechanistic approach to locate the most relevant portion of preceding utterance contexts over time. Instead, the Encoder contextual representations obtained at all time steps of preceding utterances are retained non-discriminatively. The variable length nature of the resulting concatenated contextual vectors further leads to computational efficiency and scalability issues.

In order to address the above issues, specially designed attention pooling layers are used in this paper and applied over preceding utterances' Encoder contextual vectors to auto-configure the history context weighting at different time steps. The low-rank nature of these attention pooling layers allow variable length cross utterance Encoder contextual vectors used in Sec. 3.1 to be compressed to compact, low-dimensional features to condition the prediction of current utterance outputs for Conformer-Transducer systems. This design is inspired by the attentive pooling [39] and attentive speaker embedding [40] used in speaker recognition [41] tasks. To further improve efficiency, akin to Transformer-XL [27] language models, the preceding utterances' Encoder hidden context vectors are efficiently cached prior to the attention based pooling operations. Let the C-T Encoder's outputs at the $l$-th Encoder layer be $\mathbf{h}^{l,i-1}_{1:T_{i-1}} \in \mathbb{R}^{T_{i-1}\times D}$ for the preceding $(i-1)^{\mathrm{th}}$ utterance of $T_{i-1}$ frames, where $D$ stands for the Encoder output vector dimensionality. The cross utterance Encoder contextual states are attention pooled and projected to low-dimensional representations $\tilde{\mathbf{h}}^{l,i-1}_{1:L} \in \mathbb{R}^{L\times D}$ where and $L \ll T_{i-1}$, as

$$\begin{aligned}\hat{\mathbf{h}}^{l,i-1}_{1:T_{i-1}} &= \mathrm{SG}(\mathbf{h}^{l,i-1}_{1:T_{i-1}})\\ \tilde{\mathbf{h}}^{l,i-1}_{1:L} &= \mathrm{softmax}(\mathrm{bn}(\mathrm{relu}(\mathbf{E}(\hat{\mathbf{h}}^{l,i-1}_{1:T_{i-1}})^T))) \cdot \mathbf{h}^{l,i-1}_{1:T_{i-1}}\end{aligned} \tag{4}$$

$\mathbf{E} \in \mathbb{R}^{L\times D}$ denotes the projection matrix to be learned, and $\mathrm{bn}(\cdot)$ stands for Batch Normalization. The resulting compact, fixed length $L \times D$ cross utterance Encoder contextual features are used together with the current utterance in C-T model training and evaluation. An example of C-T models using such compressed cross utterance Encoder contexts are shown in Fig. 1(c).

### 3.3. Predictor Contextual Representation

In the LSTM based Predictor module, cross utterance contextual information can be further incorporated. A standard approach also considered in this paper is to cache the Predictor hidden vector state obtained at the end of the preceding utterance and concatenate it with the current Predictor input. Given the input label sequence $\mathbf{y}^i_{1:u-1}$. The last Predictor hidden state computed from the previous utterance is cached as $\mathbf{f}^{i-1}_{U_{i-1}}$, before being concatenated with the current input $\mathbf{y}^i_{1:u-1}$ and fed into the Predictor module. The resulting cross utterance context conditioned Predictor vector outputs are

$$\mathbf{f}^i_{u-1} = \mathrm{Predictor}(\mathbf{y}^i_{1:u-1} \ominus \mathbf{f}^{i-1}_{U_{i-1}}) \tag{5}$$

An example of using such preceding utterance's context in a C-T system's Predictor are in Fig. 1(b) and (c) (via blue lines).

### 3.4. Data Serialization

In traditional utterance-level based ASR training, we shuffle the data and construct multiple utterances in one minibatch based on their duration time. However, to capture cross utterance contexts, we serialized the training data of the same audio clip or conversation session based on utterances' start times. An example of data serialization for contextual C-T system training is shown in Fig. 2. The batch size is 3 and the utterances are from five audio clips A to E. Clip A contains five utterances. These 5 utterances in Clip A (in pink) are lined up based on their start times, while the cross utterance contexts are used as indicated by blue arrows. The red cross marks indicate cross utterance contexts are not used when starting processing the 1st utterance of a new clip, for example, E or D at minibatch 3. Since the number of utterances varies from the clips or sessions, short clips may not have enough utterances to fill the minibatches. In this case, utterances of other clips will be used to fill the minibatches and minimise synchronisation overhead.

## 4. Experiments

### 4.1. Experiment Setup

The Gigaspeech M size corpus [44] with 1000-hr speech collected from Audiobook, Podcast and YouTube is used for training. The dev and test sets randomly selected from Podcast and YouTube data containing 12 and 40 hours of speech were used. In addition to the Conformer-Transducer system description of Section 2.1, raw speech were used as input features. The Conformer-Transducer model followed the ESPnet recipe [8] configuration. For C-T Encoder, we stacked 12 Encoder blocks where each Encoder block is configured with 8-head attention of 512-dim, and 2048 feed forward hidden nodes. For C-T Predictor, 1 uni-directional LSTM layer with 300 hidden size was adopted. 5000 byte-pair-encoding (BPE) tokens were served as the joint network outputs. The convolution subsampling module contains 2-D convolutional layers with kernel size 31. SpecAugment [45] and dropout (rate set as 0.1) were used in training, together with model averaging performed over the last five epochs. Besides, we investigated streaming C-T model by applying a 1-frame self-attention look-ahead at each Encoder layer. Real time factors (RTFs) are measured on the test data for the computation incurred within the Encoder sublayers where the previous and current utterance representations are fused (Fig. 1 (b) and (c), red dotted boxes). All C-T systems with or without using cross utterance context are trained from scratch[1]. The average utterance lengths are 3.96, 7.16 and 6.39 seconds for the training, Dev and Test sets respectively.

### 4.2. Evaluation Results

Table 1: *WER% & RTFs of streaming Conformer-Transducers (C-T) without/with cross utterance context on the Gigaspeech M size corpus. † denotes statistically significant WER differences (MAPSSWE, $\alpha = 0.05$ [46, 47]) over the baseline (sys.1).*

| ID | System | Data Repre. | Encoder #Prev. Utt. | Encoder Context Vec. Len. | Predictor Prev. Utt. Context | WER% Dev | WER% Test | RTF (Encoder Fusion) |
|---|---|---|---|---|---|---|---|---|
| 1 | Streaming C-T | Clip | - | | | 16.4 | 16.2 | 0.054 |
| 2 | | | 1 | UttLen ×512 | ✓ | 15.9$^\dagger$ | 15.8$^\dagger$ | 0.072 |
| 3 | | | 1 | **16** × 512 | | 16.2 | 16.0 | 0.058 |
| 4 | | | 2 | UttLen ×512 | ✓ | **15.7$^\dagger$** | **15.7$^\dagger$** | 0.076 |
| 5 | | | 2 | **32** × 512 | | 15.9$^\dagger$ | 15.8$^\dagger$ | **0.058** |

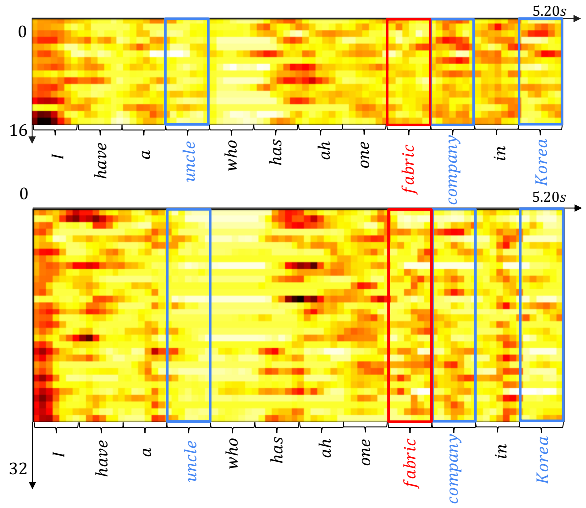


Figure 3: *Examples of attention heat maps applied to the contextual representations (L=16, top; L=32, bottom) of a single previous utterance containing a sentence "I have a uncle who has ah one **fabric** company in Korea", which assign larger weights to words "**fabric**" in the contextual C-T model of Eqn. (4)-(5). The current utterance's recognition outputs are "He always exports **fabric** to united states" and "He always exports **family** to united states" with or without using cross utterance context respectively.*

The word error rates (WERs) and real-time factor (RTFs) of streaming C-T model without or with cross utterance contexts evaluated on Gigaspeech M size corpus are shown in Table 1. Several trends can be found. **1)** utilizing frame level concatenated cross utterance context states both in Encoder and Predictor modules (sys.2) outperforms the baseline C-T system without cross utterance context (sys.1) by statistically significant WER reductions of 0.5% and 0.4% absolute (3.0% and 2.5% relative) on the dev and test sets. **2)** By using the attention pooled, 16×512 or 32×512-dimensional preceding utterance's contextual features of Eqn. (4), similar performance improvements over the baseline C-T model using utterance internal context were obtained, in particular when the number of preceding utterances is increased to 2 (Sys. 5, 4 vs. 1). As expected, the compact nature of the compressed cross utterance contextual features incur a smaller increase in computation (measured in RTFs) during Encoder context fusion by 7.5% over the baseline C-T, than that brought by frame level concatenation (Sys. 3, 5 vs. 2, 4). **3)** Increasing the number of preceding utterances produced consistent performances over only modelling the most recent one (Sys. 4, 5 vs. 2, 3), irrespective of whether frame level concatenated contextual features, or those that are attention pooled and compressed, are used. Finally, by visualizing the attention pooling heat maps of Sys. 3 and Sys. 5, interpretable weights assigned to an example preceding utterance's word contents that intuitively leads to recognition accuracy improvements for a current utterance are shown in Fig. 3.

[1]Initializing cross utterance contextual C-T models with parameters of those without using context led to poor performance.

Table 2: *WER% & RTFs of non-streaming C-T models w/o cross utterance context on the Gigaspeech M size corpus. † denotes statistically significant WER differences (MAPSSWE, $\alpha = 0.05$ [46, 47]) over the baseline (sys. 1).*

| ID | System | Data Repre. | Encoder #Prev. Utt. | Encoder Context Vec. Len. | Predictor Prev. Utt. Context | WER% Dev | WER% Test | RTF (Encoder Fusion) |
|---|---|---|---|---|---|---|---|---|
| 6 | Non-Streaming C-T | Utt. | - | | | 14.3 | 14.2 | - |
| 7 | | Clip | | | | 14.2 | 14.0 | 0.058 |
| 8 | | | 1 | UttLen × 512 | ✗ | 14.0 | 13.9 | 0.080 |
| 9 | | | 0 | - | ✓ | 14.0 | 14.0 | 0.080 |
| 10 | | | 1 | UttLen × 512 | | 14.0$^\dagger$ | 13.8$^\dagger$ | 0.080 |
| 11 | | | 1 | **8** × 512 | ✓ | 14.1 | 13.9 | 0.062 |
| 12 | | | 1 | **16** × 512 | | 14.0$^\dagger$ | 13.8$^\dagger$ | 0.064 |
| 13 | | | 1 | **32** × 512 | | 14.1 | 13.9 | 0.067 |
| 14 | | | 2 | **8** × 512 | ✓ | 14.0$^\dagger$ | 13.8$^\dagger$ | 0.062 |
| 15 | | | 2 | **16** × 512 | | 14.1 | 13.9 | 0.064 |
| 16 | | | 2 | **32** × 512 | | **14.0$^\dagger$** | **13.7$^\dagger$** | 0.067 |

Similar trends are observed on the experiments conducted using non-streaming C-T systems in Table 2. **1)** Utilizing cross utterance context the in Encoder (Sys. 8), Predictor (Sys. 9), or both (Sys. 10) consistently outperformed the baseline C-T without using such information (Sys. 6, 7). **2)** Compressing the cross utterance contextual features via attention pooling produced comparable performance (Sys. 12 vs. 10), and again a smaller increase in RTF. **3)** Further increasing the number of history utterances being considered from one to two (Sys.14 - 16), the largest performance improvements over the baseline C-T systems (Sys. 16 vs. Sys. 6, 7) by statistically significant WER reductions of 0.3% and 0.5% absolute (2.0% and 3.5% relative) were obtained on the dev and test sets, respectively.

## 5. Conclusions

In this paper, compact cross utterance contextual representations were incorporated into Conformer-Transducer (C-T) ASR systems using contextual attention pooling layers integrated with the C-T Encoder. Cross utterance contexts are also incorporated into the Predictor. Experiments on the 1000-hr Gigaspeech M corpus demonstrate that the proposed cross utterance context conditioned streaming Conformer-Transducer system outperform the baseline using utterance internal context only with statistically significant WER reductions of 0.7% to 0.5% absolute (4.3% to 3.1% relative) on the dev and test data, while incurring moderate increase of latency by 7.5% in cross utterance context fusion. Future work will improve cross utterance contextual C-T models' generalisation and efficiency.

## 6. Acknowledgements


This research was supported by Hong Kong RGC GRF grant No. 14200021, 14200218, 14200220, Innovation & Technology Fund grant No. ITS/218/21, the National Natural Science Foundation of China (No. 62206171), the International Cooperation Project of PCL, and Alibaba Group through Alibaba Innovative Research Program.

# 7. References


[1] A. Graves, "Connectionist temporal classification," in *Supervised sequence labelling with recurrent neural networks*, 2012.

[2] S. Watanabe, T. Hori *et al.*, "Hybrid ctc/attention architecture for end-to-end speech recognition," *JSTSP*, 2017.

[3] W. Chan, N. Jaitly *et al.*, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016.

[4] A. Vaswani, N. Shazeer *et al.*, "Attention is all you need," *Advances in neural information processing systems*, 2017.

[5] L. Dong, S. Xu *et al.*, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *ICASSP*, 2018.

[6] S. Karita, N. Chen *et al.*, "A comparative study on transformer vs rnn in speech applications," in *ASRU Workshop*, 2019.

[7] A. Gulati, J. Qin *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[8] P. Guo, "Recent developments on espnet toolkit boosted by conformer," in *ICASSP*, 2021.

[9] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[10] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, 2001.

[11] H. Sak, A. Senior *et al.*, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," 2014.

[12] W. Zhou, W. Michel *et al.*, "Efficient training of neural transducer for speech recognition," *INTERSPEECH*, 2022.

[13] Z. Tüske, G. Saon *et al.*, "On the limit of english conversational speech recognition," *INTERSPEECH*, 2021.

[14] M. Zeineldeen, J. Xu *et al.*, "Improving the training recipe for a robust conformer-based hybrid model," *INTERSPEECH*, 2022.

[15] J. Deng, X. Xie *et al.*, "Confidence score based conformer speaker adaptation for speech recognition," *INTERSPEECH*, 2022.

[16] G. Saon, A. Gupta *et al.*, "Diagonal state space augmented transformers for speech recognition," *arXiv preprint arXiv:2302.14120*, 2023.

[17] K. Wei, Y. Zhang *et al.*, "Leveraging acoustic contextual representation by audio-textual cross-modal learning for conversational asr," *INTERSPEECH*, 2022.

[18] S. Kim and F. Metze, "Dialog-context aware end-to-end speech recognition," in *SLT Workshop*, 2018.

[19] X. Chen, Y. Wu *et al.*, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," in *ICASSP*, 2021.

[20] F.-J. Chang, J. Liu *et al.*, "Context-aware transformer transducer for speech recognition," in *ASRU Workshop*, 2021.

[21] G. Sun, C. Zhang *et al.*, "Tree-constrained Pointer Generator with Graph Neural Network Encodings for Contextual Speech Recognition," in *INTERSPEECH*, 2022.

[22] J. Hou, J. Chen *et al.*, "Bring dialogue-context into rnn-t for streaming asr," *INTERSPEECH*, 2022.

[23] S. Kim, S. Dalmia *et al.*, "Gated embeddings in end-to-end speech recognition for conversational-context fusion," *arXiv preprint arXiv:1906.11604*, 2019.

[24] K. Irie, A. Zeyer *et al.*, "Training language models for long-span cross-sentence evaluation," in *ASRU Workshop*, 2019.

[25] X. Chen, S. Parthasarathy *et al.*, "Lstm-lm with long-term history for first-pass decoding in conversational speech recognition," *arXiv preprint arXiv:2010.11349*, 2020.

[26] W. Xiong, L. Wu *et al.*, "Session-level language modeling for conversational speech," in *EMNLP*, 2018.

[27] Z. Dai, Z. Yang *et al.*, "Transformer-xl: Attentive language models beyond a fixed-length context," *ACL*, 2019.

[28] S. Kim, "End-to-end speech recognition on conversations," Ph.D. dissertation, Carnegie Mellon University, 2019.

[29] D.-R. Liu, C. Liu *et al.*, "Contextualizing asr lattice rescoring with hybrid pointer network language model," *INTERSPEECH*, 2020.

[30] X. Liu, M. J. F. Gales *et al.*, "Use of contexts in language model interpolation and adaptation," *CSL*, 2013.

[31] I. Beltagy, M. E. Peters *et al.*, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[32] G. Sun, C. Zhang *et al.*, "Transformer language models with lstm-based cross-utterance information representation," in *ICASSP*. IEEE.

[33] J. W. Rae, A. Potapenko *et al.*, "Compressive transformers for long-range sequence modelling," *arXiv preprint arXiv:1911.05507*, 2019.

[34] Z. Fan, J. Li *et al.*, "Speaker-aware speech-transformer," in *ASRU Workshop*, 2019.

[35] E. Tsunoo, Y. Kashiwagi *et al.*, "Transformer asr with contextual block processing," in *ASRU Workshop*, 2019.

[36] T. Hori, N. Moritz *et al.*, "Transformer-based long-context end-to-end speech recognition." in *INTERSPEECH*, 2020.

[37] T. Hori, N. Moritz *et al.*, "Advanced long-context end-to-end speech recognition using context-expanded transformers," *INTERSPEECH*, 2021.

[38] Q. Zhang, H. Lu *et al.*, "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *ICASSP*, 2020.

[39] C. d. Santos, M. Tan *et al.*, "Attentive pooling networks," *arXiv preprint arXiv:1602.03609*, 2016.

[40] K. Okabe, T. Koshinaka *et al.*, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.

[41] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, 1997.

[42] C.-F. Yeh, J. Mahadeokar *et al.*, "Transformer-transducer: End-to-end speech recognition with self-attention," *arXiv preprint arXiv:1910.12977*, 2019.

[43] Y. N. Dauphin, A. Fan *et al.*, "Language modeling with gated convolutional networks," in *ICML*, 2017.

[44] G. Chen, "Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio," *arXiv preprint arXiv:2106.06909*, 2021.

[45] D. S. Park, W. Chan *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[46] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *International Conference on Acoustics, Speech, and Signal Processing,*, 1989.

[47] D. S. Pallet, W. M. Fisher *et al.*, "Tools for the analysis of benchmark speech recognition tests," in *ICASSP*, 1990.