INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Raw Waveform Encoder with Multi-Scale Globally Attentive Locally Recurrent Networks for End-to-End Speech Recognition

Max W. Y. Lam[1], *Jun Wang*[1], *Chao Weng*[1], *Dan Su*[1], *Dong Yu*[2]

[1]Tencent AI Lab, Shenzhen, China
[2]Tencent AI Lab, Bellevue WA, USA

{maxwylam, joinerwang, cweng, dansu, dyu}@tencent.com

Abstract

End-to-end speech recognition generally uses hand-engineered acoustic features as input and excludes the feature extraction module from its joint optimization. To extract learnable and adaptive features and mitigate information loss, we propose a new encoder that adopts globally attentive locally recurrent (GALR) networks and directly takes raw waveform as input. We observe improved ASR performance and robustness by applying GALR on different window lengths to aggregate fine-grain temporal information into multi-scale acoustic features. Experiments are conducted on a benchmark dataset *AISHELL-2* and two large-scale Mandarin speech corpus of $5,000$ hours and $21,000$ hours. With faster speed and comparable model size, our proposed multi-scale GALR waveform encoder achieved consistent character error rate reductions (CERRs) from 7.9% to 28.1% relative over strong baselines, including Conformer and TDNN-Conformer. In particular, our approach demonstrated notable robustness than the traditional handcrafted features and outperformed the baseline MFCC-based TDNN-Conformer model by a 15.2% CERR on a music-mixed real-world speech test set.

Index Terms: end-to-end ASR, RNN-T, feature extraction, Conformer, TDNN-Conformer

1. Introduction

End-to-end automatic speech recognition (ASR) [1–5] allows a joint optimization of components for better performances. Amongst a variety of end-to-end methods, RNN-Transducer (RNN-T) [1] has recently achieved state-of-the-art (SOTA) performances [5–10]. An RNN-T comprises an encoder-decoder architecture, where the encoder is responsible for transforming a sequence of acoustic inputs into hidden vectors and the decoder utilizes previously emitted non-blank symbols to compute prediction vectors. A recent study [11] suggests that for an RNN-T optimizing encoder and learning proper representations for acoustic inputs are remarkably more vital than the decoder.

Standard ASR systems, including the SOTA RNN-T, traditionally use handcrafted acoustic features computed over segments of fixed duration, such as band-pass filtering of signals including log Mel-filterbank values (FBANK) [12] and Mel frequency cepstral coefficients (MFCC) [13]. A potential defect of such systems is being susceptible to information loss due to an intrinsic tradeoff between temporal and frequency resolution. For example, FBANK can hardly preserve phonetic information at the frame boundaries at a fixed time resolution, especially for fast speech, nor describe or separate subtle harmonics and formants at a fixed frequency resolution; Mel-filterbank operations are prone to discard local correlations within the signal [14].

An increasing number of studies [14–22] report empirical evidence that learning directly from the waveform domain can potentially outperform the analytically handcrafted features. In the majority of the previously proposed architectures, there are typically shallow or deep convolutional layers with either 1D convolutions [16, 17, 19, 21] or 2D convolutions [14], designed to emulate band-pass filtering or spectrogram features. While convolutional networks are good at modeling local correlations as well as invariance to local translations, they can not capture long context with shallow layers. Even with very deep layers [14] at increased computational and memory cost, they are still limited in capturing dynamic long-range global context as CNN simply applies a global averaging over the entire sequence. Existing models using convolutional, recurrent neural networks (RNNs, e.g., LSTMs [23] and GRUs [24]), or self-attention networks (SAN) each has its shortcomings: SAN, on the other hand, are well suited to model longer global context but less capable to extract fine-scale local features.

Grasping the fact that RNN/CNN and SAN are complementary in modeling local detail dependencies and global long-range context, we propose to adopt an architecture called Globally Attentive Locally Recurrent (GALR) network. Its advantage has been demonstrated in the Cocktail-Party problem (i.e., speech separation tasks) in our previous studies [25–27]. By a large margin, it outperforms other advanced audio separation networks using a fully convolutional network [28] and using a dual-path RNN network [29]. Built on the previous success, the unique contributions of this paper are the followings:

a) We reveal the effectiveness of the GALR networks as building modules for a waveform encoder in end-to-end ASR. One challenge in directly taking raw waveform as input is that the sequence is densely sampled in time, and thus presents the dilemma between reducing the number of timesteps and throwing away relevant information for ASR. Our work elegantly solves the critical bottleneck of self-attention networks for modeling very long sequences and makes it memorably and computationally tractable. GALR is well suited to capture not only long-range global context but also local correlations and fine-grained patterns.

b) We propose a novel architecture that jointly learns GALRs at multiple scales on raw waveforms simultaneously. The combined multi-scale features remedy the tradeoff of temporal and frequency resolution, thus allowing the model to automatically select and combine the resolutions that could most efficiently represent the needed frequencies. Meanwhile, unlike other multi-scale approaches, which generally require very deep structure at extra cost [16], our method is computationally and memorably efficient.

c) We demonstrate that our waveform encoder can outperform by a large margin over statically hand-engineered features on large vocabulary end-to-end ASR tasks. In particular, we build our baseline models using the RNN-T architecture

http://dx.doi.org/10.21437/Interspeech.2021-2084

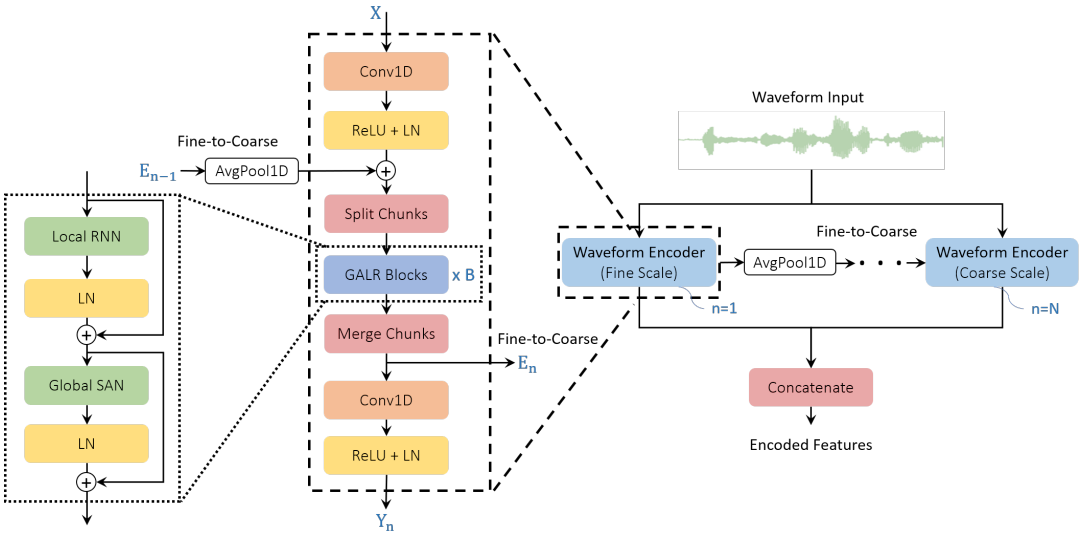

Figure 1: *Overall architecture of the multi-scale GALR waveform encoder*

with SOTA encoders, including Conformer [10] and TDNN-Conformer [30]. We observe a significant gain measured with a benchmark dataset can be consistently carried over to larger setups and generalize well to unseen complex scenarios with highly non-stationary interference. Our evaluations provide evidence for the effectiveness of directly learning a waveform encoder based on GALR and the robustness of withstanding signal corruption, as opposed to building end-to-end ASR on top of traditional features.

2. Multi-Scale GALR Waveform Encoder

We propose a raw waveform encoder with a multi-scale globally attentive locally recurrent (GALR) architecture. As shown in Figure 1, the waveform input goes through a series of N modules, each composed of B GALR blocks, to encode features at different time scales. The resultant features then concatenate and can substitute the conventional FBANK or MFCC features, and instead of being static, they are adaptive and jointly learned during the end-to-end ASR model training.

2.1. Waveform Encoder

Given a raw waveform input $\mathbf{x} \in \mathbb{R}^{\Gamma}$, we first split the sequence into L_n half-overlapping windows, denoted by $\mathbf{x}_1, ..., \mathbf{x}_{L_n}$, where $\mathbf{x}_i \in \mathbb{R}^{M_n}, i = 1, ..., L_n$, and M_n is the window length changing from small to large while the scale index n goes from 1 to N. These windows are 50% overlapped, and the last one is padded with a zero vector to make $L_n = \lceil 2\Gamma/M_n \rceil$. Then, analogous to traditional short-time Fourier transform (STFT), L_n frames are generated by non-linearly projecting $\mathbf{x}_i$ onto a D-dimension feature space, forming a matrix $\mathbf{F}_n \in \mathbb{R}^{D \times L_n}$:

$$\mathbf{F}_n = \text{LN}(\text{ReLU}(\text{Conv1D}([\mathbf{x}_1, ..., \mathbf{x}_{L_n}]))), \quad (1)$$

where $\text{Conv1D}(\cdot) = \mathbf{U}_n * [\mathbf{x}_1, ..., \mathbf{x}_{L_n}]$, $*$ denotes the 1D convolution operation, $\mathbf{U}_n \in \mathbb{R}^{D \times M_n}$ is a learnable matrix of D basis vectors, $\text{ReLU}(\cdot)$ is the rectified linear unit for ensuring the non-negativity [25], and $\text{LN}(\cdot)$ is the layer normalization method [31] for stabilizing and accelerating the model training.

To inherit fine-to-coarse information, we then connect the output of the $(n-1)$-th network to $\mathbf{F}_n$:

$$\tilde{\mathbf{F}}_n = \mathbf{F}_n + \text{AvgPool1D}\left(\mathbf{E}_{n-1}\right), \quad (2)$$

where $\mathbf{E}_{n-1} \in \mathbb{R}^{D \times L_{n-1}}$ is the output (detailed in Section 2.3) of the preceding GALR block at a finer scale (i.e., $L_{n-1} > L_n$), and for initialization, we define $\mathbf{E}_0 = \mathbf{0}$ for the first block, and $\text{AvgPool1D}(\cdot)$ is an averaging pooling operation that interpolates the length L_{n-1} of $\mathbf{E}_{n-1}$ to match the length L_n of $\mathbf{F}_n$.

Before conducting GALR operations, we further split $\tilde{\mathbf{F}}_n$ into S_n half-overlapping chunks each of length K_n. The first and last chunks are padded with zeros to generate $S_n = \lceil 2L_n/K_n \rceil$ chunks. These chunks can be represented as a 3D tensor $\mathcal{Q}_n \in \mathbb{R}^{D \times S_n \times K_n}$.

2.2. GALR Blocks

As shown in the middle of Figure 1, for the building blocks in each waveform encoder, we adopt the same GALR structure originally proposed for our speech separation tasks [25]. In this work, we find that it can generalize surprisingly well for ASR. For integrity, we briefly describe the GALR architecture here. The left diagram in Figure 1 illustrates the connections within a GALR block. For each block, we denote the input as $\mathcal{Q}_{n,b} \in \mathbb{R}^{D \times S_n \times K_n}$, where $n = 1, ..., N$ is the scale index, $b = 1, ..., B$ is the block index, and $\mathcal{Q}_{n,1} = \mathcal{Q}_n$. Each GALR block contains two primary layers – a local RNN layer and a global SAN [32] layer, corresponding to the intra- and the inter-chunk processing, respectively.

2.2.1. Local RNN Layer

The local RNN layer with D hidden nodes captures short-term dependencies within each chunk:

$$\mathcal{L}_{n,b} = \left[\text{Linear}\left(\text{RNN}\left(\mathcal{Q}_{n,b}[:, s, :]\right)\right), s = 1, ..., S_n\right], \quad (3)$$

where $\text{Linear}(\cdot)$ is a trainable linear mapping with a bias term enabled, $\text{RNN}(\cdot)$ is an RNN model, e.g. GRU or LSTM, and $\mathcal{Q}_{n,b}[:, s, :] \in \mathbb{R}^{D \times K_n}$ refers to the local sequence within the s-th chunk, each of length K_n. The output of the local RNN then goes through $\text{LN}(\cdot)$ followed by a residual connection:

$$\tilde{\mathcal{L}}_{n,b} = \text{LN}(\mathcal{L}_{n,b}) + \mathcal{Q}_{n,b}. \quad (4)$$

2.2.2. Global SAN Layer

Then, we transpose the 3D tensor $\tilde{\mathcal{L}}_{n,b} \in \mathbb{R}^{D \times S_n \times K_n}$ to process inter-chunk sequences, each of length S_n. A multi-head

SAN layer [32] is used to capture the global dependencies:

$$\mathcal{G}_{n,b} = \text{UpSmpl}\left(\text{SAN}\left(\text{DownSmpl}\left(\tilde{\mathcal{L}}_{n,b}\right)\right)\right), \quad (5)$$

where SAN$(\cdot)$, UpSmpl$(\cdot)$, and DownSmpl$(\cdot)$ are the self-attentive, upsampling, and downsampling operations, respectively. Readers are referred to [26] for more details of these operations, which we omit here for simplicity.

Afterward, the output of the b-th GALR block is passed forward with a residual connection as the input for the $(b+1)$-th GALR block as follows:

$$\mathcal{Q}_{n,b+1} = \text{LN}(\mathcal{G}_{n,b}) + \tilde{\mathcal{L}}_{n,b}, \quad (6)$$

2.3. Merging Chunks

As shown in the middle diagram of Figure 1, after processing through the B GALR blocks, we merge the chunks to transform the 3D tensor back to a sequence of feature vectors for the downstream ASR task. Inspired by the mask estimation in [26], the merging operations entail a non-linear mapping, without which we observed a loss degradation in our ablation study. In particular, we use a Swish-gated 2D convolutional layer for the non-linear mapping:

$$\mathcal{F}_n = \text{Conv2D}\left(\text{Swish}\left(\mathcal{Q}_{n,B}\right)\right), \quad (7)$$

where Conv2D$(\cdot)$ is a point-wise 2D convolutional layer with a 1×1 kernel, and Swish$(\cdot)$ is the element-wise Swish activation function [33]. Afterward, we merge the S_n chunks using OverlapAdd [29] to transform the 3D tensor $\mathcal{F}_n \in \mathbb{R}^{D \times S_n \times K_n}$ back to a feature matrix:

$$\mathbf{E}_n = \text{OverlapAdd}\left(\mathcal{F}_n\right), \quad (8)$$

which is also passed to the next GALR encoder to establish the fine-to-coarse information flow, as indicated in Eq. 2.

2.4. Multi-scale Features

Current SOTA ASR systems (e.g., RNN-T [10]) typically use analytical acoustic features computed over a fixed window size of 25ms and a stride length of 10ms. Consequently, the time and frequency resolution have been fixed a priori rather than being adaptive or learnable using the training data and the end-to-end model learning. These irreversible operations inevitably cause information loss at a landing stage of scaffold for ASR.

In the light of the successful fine-grained modeling of GALR networks, we are able to flexibly and effectively handle frames over fine-to-coarse windows (i.e. using 0.2ms to 2ms windows for the Cocktail-Party problem [25]). The fine-to-coarse range could be larger in ASR. For example, striking a balance between computation and performance in our practical setup for large industrial datasets, we empirically set three window sizes: 6.25ms, 12.5ms, and 25ms. During model training we let the network decide which elements under the varying resolutions contain useful information adaptively to the training data based on the back-propagated RNN-T loss.

Optimizing the RNN-T loss per sample [1] entails computing a tensor in shape $D' \times T \times U$, where T is the length of encoded vectors, U is the length of prediction vectors, and D' is the feature dimension. The RNN-T encoder (e.g. in [10]) has down-sampling processes to keep T small enough for not exceeding GPU memory and make the back-propagation feasible. Similarly, we add a non-linear down-sampling layer after the chunks-merging operation:

$$\mathbf{Y}_n = \text{LN}\left(\text{ReLU}\left(\text{Conv1D}_{C_n}\left(\mathbf{E}_n\right)\right)\right), \quad (9)$$

where $\text{Conv1D}_{C_n}(\cdot)$ denotes the 1D-convolution for downsampling with a kernel size of $2C_n$, a stride length of C_n, and a padding length of C_n on both sides, such that the resultant length of $\mathbf{Y}_n$ is about $1/C_n$ of the length of $\mathbf{E}_n$, i.e., $\mathbf{Y}_n \in \mathbb{R}^{D \times \lceil L_n/C_n \rceil}$.

After computing the downsampled features $\mathbf{Y}_n$ at $n = 1, ..., N$ different scales, a straightforward approach to generate a multi-scale feature is to directly concatenate $\mathbf{Y}_n$ along the D dimension. However, there could be a sequence length mismatch problem. Mathematically, we want to make

$$\left\lceil \frac{L_n}{C_n} \right\rceil = \left\lceil \frac{L_{n+1}}{C_{n+1}} \right\rceil \Leftrightarrow \left\lceil \left\lceil \frac{2\Gamma}{M_n} \right\rceil \frac{1}{C_n} \right\rceil = \left\lceil \left\lceil \frac{2\Gamma}{M_{n+1}} \right\rceil \frac{1}{C_{n+1}} \right\rceil \quad n = 1, ..., N-1, \quad (10)$$

therefore, we set the hyperparameters M_n and C_n to satisfy a relation $M_n C_n = M_{n+1} C_{n+1}$.

Due to the nested ceiling operations in Eq.10, there could still be cases where some of the N sequences are one element longer or shorter. We define $T = \min_n \lceil L_n/C_n \rceil$ and discard the edge elements indexed at $L > L_{\text{min}}$. This yields the final multi-scale features:

$$\mathbf{Y} = \text{Concat}\left(\mathbf{Y}_1, ..., \mathbf{Y}_n\right). \quad (11)$$

The encoded waveform features $\mathbf{Y} \in \mathbb{R}^{ND \times T}$ are then fed to the later RNN-T encoder model, e.g., Conformer [10].

3. Evaluation and Analysis

3.1. Experimental Setup

3.1.1. Data Preparation

Our models were trained on three datasets at 16kHz sampling rate: the 1,000-hour *AISHELL-2* benchmark [34], a ∼5,000-hour (*5khrs*) and a ∼21,000-hour (*21khrs*) Mandarin speech datasets [35]. The models were evaluated on a variety of test sets, covering 1) *Mic*: the benchmark AISHELL-2 test set, 2) *Read*: 1.5-hour read speech (Read), 3) *Spon*: 2-hour spontaneous speech, and 4) *Music*: 2.2-hour speech with background music collected from a real-world video-sharing platform.

3.1.2. Model Setup and Training Details

Two cutting-edge encoder architectures that gave SOTA performances [30], Conformer and TDNN-Conformer, were taken as our reference systems. They were built on two types of input representations: 1) Traditional static features. To select the strongest static features, we empirically examined the performances on the largest training set (*21khrs*) with regard to MFCC and FBANK features both following the recipe in Kaldi [36] (PyKaldi [37] was used to read the features extracted in Kaldi). On the *Spon* test set, 40-dimension high-resolution MFCC outperformed 80-dimension FBANK by about 0.1% absolute reduction on average character error rates (CERs). A similar observation was also made in [35]. Consequently, we selected MFCC for our baseline systems using handcrafted features. 2) End-to-end learnable features, for which we also trained baselines using the single-scale and multi-scale convolution-based approach [16, 18] to directly learn filterbanks from the raw waveform. All systems were trained following the RNN-T, which was implemented in the PIKA[1] library [35, 38], and conducted on 16 NVIDIA Tesla V100 GPU devices. For simplicity, we refer our readers to [38] for the detailed RNN-T setups.

[1] https://github.com/tencent-ailab/pika

For the Conformer models, we reproduced the small, medium, and large Conformer encoders, with setups as described in [10], and denoted them as Conf-S, Conf-M, and Conf-L. Moreover, the TDNN-Conformer model (denoted as TDNN-Conf) has shown better performances in [30] than TDNN-Transformer [8] and Conformer [10]. Our evaluation also showed a consistent conclusion.

For the multi-scale GALR networks, we set $N = 3, B = 1, \{M_1, M_2, M_3\} = \{100, 200, 400\}, \{K_1, K_2, K_3\} = \{48, 24, 12\}, \{C_1, C_2, C_3\} = \{8, 4, 2\}$. The GALR feature dimension (D) was set to 128 for *AISHELL-2* and to 256 for *5khrs* and *21khrs* datasets, respectively. For a fair comparison, all models had a comparable number of parameters to its counterpart. We removed the MFCC feature extraction step and the first encoder layers, which were the first 8/4/3 conformer layers for the S/M/L Conformer models and 3 TDNN layers for the TDNN-Conformer models, and replaced them with the multi-scale GALR waveform encoder, while the overall sizes matched the respective baseline models.

3.2. Results and Discussion

3.2.1. Performances on Benchmark AISHELL-2

Table 1: *Performances of MFCC and the multi-scale GALR based SOTA encoders trained on the AISHELL-2 training set and evaluated on the AISHELL-2 test set (Mic)*

Encoder	**#Params. (M)**		**CERs (%)**	
	MFCC	**GALR**	**MFCC**	**GALR**
Conf-S [10]	8.7	8.3	17.1	**12.3**
Conf-M [10]	27.2	26.9	15.2	**11.9**
Conf-L [10]	114	114	13.3	**11.7**
TDNN-Conf [30]	102	99.0	12.7	**11.7**

First and foremost, we trained various types of RNN-T encoders on the benchmark *AISHELL-2* [34]. As shown in Table 1, across different configurations of the SOTA Conformer encoders, applying the multi-scale GALR waveform encoder in place of the handcrafted features consistently obtained CER reductions, from 7.9% to 28.1% relative. The results also indicated that the traditional TDNN-Conformer outperformed the traditional Conformers, echoing the report by [30], therefore, we selected the TDNN-Conformer encoder as the reference models for the following evaluations in larger scales.

It was also worth mentioning that, to focus on the comparisons concerning different encoder architectures, all the ASR performances through out this paper were obtained from a pure RNN-T system, without using any additional resource or technique (e.g., NNLM, MBR [8] or LAS rescoring [9]), which could be used together with our approach to further boost the performances.

3.2.2. Performances on 5khrs Using Varying Window Scales

This section describes an ablation study to verify that the proposed GALR waveform encoder benefits from the multi-scale structure. To this end, we trained the GALR based TDNN-Conformer in different sets of window scales, as listed with M_n in milliseconds in Table 2. Here we added another strong baseline that also learned features directly from the raw waveform, i.e. the 1D-convolution-based filterbank learning approach (Conv1D) [17]. As shown in Table 2, the performances of multi-scale GALR based RNN-T systems remarkably reduced the CERs over the non-adaptive handcrafted features from 16.0% to 21.3% relatively. In contrast, although the Conv1D based RNN-T systems also improved over the MFCC baseline, the improvements were marginal compared to GALR.

Table 2: *Performances of MFCC, Conv1D, and GALR based TDNN-Conformer trained on the 5khrs speech and evaluated on the Read test set.*

Features	**Window Scales** (M_n in ms)	**CERs (%)**
MFCC	25ms	9.4
Conv1D [16, 17]	25ms	9.2
	12.5ms	9.1
	{6.25,12.5,25}ms	8.9
GALR	12.5ms	9.0
	{6.25,12.5}ms	7.9
	{6.25,12.5,25}ms	7.6
	{6.25,12.5,25,50}ms	**7.4**

3.2.3. Performances on 21khrs Large Industrial Dataset

Table 3: *Performances of the TDNN-Conformer and the one using multi-scale GALR encoder, both trained on the 21khrs speech and evaluated on Read, Spon, and Music test sets*

Encoder	**CERs (%)**			**Speed**
	Read	*Spon*	*Music*	
TDNN-Conf [30]	4.4	13.4	29.7	239 char/s
(w) GALR encoder	**4.1**	**13.2**	**25.2**	**382 char/s**

We further investigated the performances on the *21khrs* training set, which was a larger and more realistic industrial dataset. We also used a variety of test sets with a broader diversity such as different speaker speaking styles (i.e., read and spontaneous) and scenarios (with music interference in a video-sharing platform). As shown in Table 3, the model using the proposed multi-scale GALR encoder obtained consistently lower CERs than the traditional TDNN-Conformer. The result showed that the advantage of our approach could sustain in large-scale data and generalize well to complex scenarios. Especially, the CER for *Music* was reduced by a large margin of 15.2% relatively. Please note that our baseline model was fairly strong using TDNN, which is successful in modeling context information. This demonstrated the powerful capability of the proposed GALR to capture global dynamic context, such as in the unseen complex scenarios with non-stationary music interference as illustrated here. This evaluation empirically proves the robustness of withstanding interference by directly learning a waveform encoder based on GALR. Meanwhile, our approach sped up the processing notably from 239 to 382 characters per second.

4. Conclusions

We propose a novel architecture that jointly learns GALR at multiple different scales on raw wave-forms simultaneously, which intrinsically remedy the tradeoff of temporal and frequency resolution, thus mitigating the information loss for end-to-end ASR. Our evaluations demonstrate the effectiveness of directly learning waveform encoder based on GALR, as opposed to building end-to-end ASR on top of traditional features. Moreover, our method is computationally and memory efficient. Future work includes exploring this novel architecture in a cocktail party scenario to solve the speech separation and ASR tasks simultaneously in an end-to-end solution.

5. References

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

[2] H. Soltau, H. Liao, and H. Sak, "Neural speech recognizer: Acoustic-to-word lstm model for large vocabulary speech recognition," *Proc. Interspeech 2017*, pp. 3707–3711, 2017.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[4] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 193–199.

[5] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[6] S. Wang, P. Zhou, W. Chen, J. Jia, and L. Xie, "Exploring rnn-transducer for chinese speech recognition," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019, pp. 1364–1369.

[7] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving rnn transducer modeling for end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 114–121.

[8] C. Weng, C. Yu, J. Cui, C. Zhang, and D. Yu, "Minimum Bayes Risk Training of RNN-Transducer for End-to-End Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 966–970. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1221

[9] T. N. Sainath, R. Pang, D. Rybach, Y. He, R. Prabhavalkar, W. Li, M. Visontai, Q. Liang, T. Strohman, Y. Wu *et al.*, "Two-pass end-to-end speech recognition," *Proc. Interspeech 2019*, pp. 2773–2777, 2019.

[10] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Proc. Interspeech 2020*, pp. 5036–5040, 2020.

[11] H. Shrivastava, A. Garg, Y. Cao, Y. Zhang, and T. Sainath, "Echo state speech recognition," *arXiv preprint arXiv:2102.09114*, 2021.

[12] *A pattern-matching procedure for automatic talker recognition*, vol. 35, 1963.

[13] *Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*, vol. 28, 1980.

[14] D. Oglic, Z. Cvetkovic, P. Bell, and S. Renals, "A deep 2d convolutional network for waveform-based speech recognition," *Proc. Interspeech 2020*, pp. 1654–1658, 2020.

[15] T. N. Sainath, B. Kingsbury, A.-r. Mohamed, and B. Ramabhadran, "Learning filter banks within a deep neural network framework," in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 297–302.

[16] Z. Zhu, J. H. Engel, and A. Hannun, "Learning multiscale features directly from waveforms," *Interspeech*, 2016.

[17] N. Zeghidour, N. Usunier, I. Kokkinos, T. Schaiz, G. Synnaeve, and E. Dupoux, "Learning filterbanks from raw speech for phone recognition," in *2018 IEEE international conference on acoustics, speech and signal Processing (ICASSP)*. IEEE, 2018, pp. 5509–5513.

[18] N. Zeghidour, N. Usunier, G. Synnaeve, R. Collobert, and E. Dupoux, "End-to-end speech recognition from the raw waveform," *arXiv preprint arXiv:1806.07098*, 2018.

[19] *Acoustic modeling of speech waveform based on multi-resolution, neural network signal processing*, 2018.

[20] M. Ravanelli and Y. Bengio, "Speech and speaker recognition from raw waveform with sincnet," *arXiv preprint arXiv:1812.05920*, 2018.

[21] H. Khan and B. Yener, "Learning filter widths of spectral decompositions with wavelets," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 4606–4617.

[22] E. Loweimi, P. Bell, and S. Renals, "On learning interpretable cnns with parametric modulated kernel-based filters." in *INTERSPEECH*, 2019, pp. 3480–3484.

[23] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[24] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[25] M. W. Lam, J. Wang, D. Su, and D. Yu, "Effective low-cost time-domain audio separation using globally attentive locally recurrent networks," *SLT*, 2021.

[26] ——, "Sandglasset: A light multi-granularity self-attentive network for time-domain speech separation," *ICASSP*, 2021.

[27] J. Wang, M. W. Lam, D. Su, and D. Yu, "Tune-in: Training under negative environments with interference for attention networks simulating cocktail party effect," *AAAI*, 2021.

[28] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[29] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 46–50.

[30] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi *et al.*, "Recent developments on espnet toolkit boosted by conformer," *arXiv preprint arXiv:2010.13956*, 2020.

[31] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[33] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[34] J. Du, X. Na, X. Liu, and H. Bu, "Aishell-2: Transforming mandarin asr research into industrial scale," *arXiv preprint arXiv:1808.10583*, 2018.

[35] C. Weng, J. Cui, G. Wang, J. Wang, C. Yu, D. Su, and D. Yu, "Improving attention based sequence-to-sequence models for end-to-end english conversational speech recognition," in *Proc. Interspeech 2018*, 2018, pp. 761–765. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1030

[36] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[37] D. Can, V. R. Martinez, P. Papadopoulos, and S. S. Narayanan, "Pykaldi: A python wrapper for kaldi," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5889–5893.

[38] C. Weng, C. Yu, J. Cui, C. Zhang, and D. Yu, "Minimum Bayes Risk Training of RNN-Transducer for End-to-End Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 966–970. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1221