Chunk-Level Speech Emotion Recognition: A General Framework of Sequence-to-One Dynamic Temporal Modeling

Wei-Cheng Lin, *Student Member, IEEE* and Carlos Busso, *Senior Member, IEEE*

Abstract—A critical issue of current speech-based sequence-to-one learning tasks, such as *speech emotion recognition* (SER), is the dynamic temporal modeling for speech sentences with different durations. The goal is to extract an informative representation vector of the sentence from acoustic feature sequences with varied length. Traditional methods rely on static descriptions such as statistical functions or a *universal background model* (UBM), which are not capable of characterizing dynamic temporal changes. Recent advances in deep learning architectures provide promising results, directly extracting sentence-level representations from frame-level features. However, conventional cropping and padding techniques that deal with varied length sequences are not optimal, since they truncate or artificially add sentence-level information. Therefore, we propose a novel dynamic chunking approach, which maps the original sequences of different lengths into a fixed number of chunks that have the same duration by adjusting their overlap. This simple chunking procedure creates a flexible framework that can incorporate different feature extractions and sentence-level temporal aggregation approaches to cope, in a principled way, with different sequence-to-one tasks. Our experimental results based on three databases demonstrate that the proposed framework provides: 1) improvement in recognition accuracy, 2) robustness toward different temporal length predictions, and 3) high model computational efficiency advantages.

Index Terms—Sequence-to-one modeling, speech emotion recognition, attention model, chunk-level modeling

1 Introduction

Summarization from perceived information is an essential ability during human decision making processes. Our brain can effectively extract and summarize information obtained from different sources, including visual and acoustic modalities, to make decisions. The concept of sequence-to-one learning tasks in machine learning aim to imitate and learn the human's summarization mechanism [1], [2]. A critical challenge of sequence-to-one learning is efficiently extracting insightful information from data with different durations. More specifically, the model is required to *learn how to summarize* relevant temporal information from a varied length input, mapping the sequence into a single label.

Speech emotion recognition (SER) is a task that is often formulated as a sequence-to-one problem following the labels provided by existing databases. Some of the emotional corpora are annotated with time-continuous emotional traces [3], [4], [5], providing labels for sequence-to-sequence formulation. However, those databases are hard to collect, as the annotation process is time consuming and the inter-evaluation agreement is often low [6], [7]. There are also additional challenges with emotional traces, including compensating for the reaction lag of the evaluators [8], [9]. Therefore, most existing speech emotional corpora are labeled at the sentence-level (i.e., one global label is assigned per sentence [10], [11], [12]). As a result, most studies in SER rely on standard sequence-to-one learning tasks.

Traditionally, studies rely on estimation of *high level descriptors* (HLDs) (i.e., a fixed set of statistical functions) from *low level descriptors* (LLDs) extracted from speech [13]. For instance, we can compute the mean of the fundamental frequency and the variance of the *Mel frequency cepstral coefficients* (MFCCs) to obtain a single fixed dimensional feature vector that represents the sentence, regardless of its duration. This vector is then used to train a machine learning model such as a *support vector machine* (SVM) [14], [15] or a *fully connected neural network* (FCNN) [16]. Another approach to obtain the sentence-level representation vector is to train a *universal background model* (UBM) such as a *Gaussian mixture models* (GMM), and then utilize a *bag-of-words* (BOW) model or Fisher vector algorithm to extract the encoding output [17], [18]. However, these methods provide static descriptions by either using fixed statistical functions or a fixed pre-trained background model, which cannot reflect the dynamic temporal information in the expression of emotion, leading to limited performance for SER systems.

Deep learning approaches for SER systems have recently led to state-of-the-art performance [19], [20], [21]. Different architectures exploring temporal information, such as *recurrent neural networks* (RNNs), *convolution neural networks* (CNNs) or hybrid neural networks (CNN-LSTM), have shown state-of-the-art performance by deriving features directly from LLDs or raw waveforms [22], [23], [24], [25], [26]. A conventional approach to dealing with speech

• *The authors are with the Erik Jonsson School of Engineering and Computer Science, University of Texas at Dallas, Richardson, TX 75080 USA. E-mail: {wei-cheng.lin, busso}@utdallas.edu.*

Manuscript received 7 Jan. 2021; revised 20 Apr. 2021; accepted 18 May 2021. Date of publication 26 May 2021; date of current version 31 May 2023.
(Corresponding author: Carlos Busso.)
Recommended for acceptance by F. Metze.
Digital Object Identifier no. 10.1109/TAFFC.2021.3083821

sequences with varied lengths is to force them to have the same length by either cropping the signal or zero-padding the sequence [25], [26]. However, cropping a sentence into a specific duration truncates temporal information that can be valuable. For example, acoustic features at the end of a sentence provide discriminative information to predict happiness [27]. The zero-padding method fixes the length of the input sequences, which is convenient for batch training. However, it does not solve the essential problem of temporal modeling, achieving robust performance toward any duration of the inputs, especially for long sequences. We need a dynamic temporal framework that is able to capture the entire information in a sentence regardless of its duration, allowing end-to-end training.

This study presents a general framework to solve current sequence-to-one temporal modeling issues, where our focus is on SER tasks. The framework consists of four components: The first block is the *feature extraction* step, where we obtain representative frame-level acoustic features. These features can be either LLDs, Mel-filter bank, or raw spectrogram. The second block is the *dynamic chunk segmentation* step, where the frame sequence of arbitrary duration is split into a fixed number of small chunks with the same duration by adjusting the overlap between chunks. This approach does not rely on the zero padding technique. The third block is the *chunk-level feature representation* step, which extracts a feature representation for each chunk. The implementation of this step is flexible, as we can use several deep learning approaches, including CNN, RNN or FCNN. The fourth block is the *sentence-level temporal aggregation* step, which combines the chunk-level features into a sentence-level representation. This component is also flexible, since it can be implemented with different approaches, such as an attention model, a gate mechanism or a temporal pooling. The core part of the framework is our proposed novel chunk segmentation process, which enables flexible combinations of different state-of-the-art methods used in deep-learning by transforming a speech signal of varied length into a fixed number of chunks with the same duration. This end-to-end framework not only preserves complete temporal information, but also effectively captures emotionally rich regions within a sentence by jointly training the chunk-level aggregation models. One major advantage of our proposed flexible framework is that it can accommodate different deep-learning implementations. We explore multiple combinations of the framework components, implementing the chunk-level feature representation with LSTM, CNN, or functional models, and the sentence-level temporal aggregation with a mean pooling layer (*NonAtten*), a gated network (*GatedVec*), an attention mechanism (*RNN-AttenVec*), or a scaled dot-product self-attention model (*Self-AttenVec*). The proposed formulation is also computationally efficient, since the size and number of chunks are fixed, facilitating parallel computing.

Our experimental results based on the MSP-Podcast database [10] demonstrate that our proposed framework (under any combination) outperforms other sentence-level and chunk-level SER baseline models. We find that the key factor leading to the performance improvement is the sentence-level aggregation module, indicating the importance of modeling the complete temporal information of the sentence. Further analysis shows that our proposed framework not only increases the accuracy of our predictions, but also introduces additional advantages including robust predictions toward different duration inputs, model efficiency, and task generality. The two major contributions of this study are:

- A novel dynamic chunk segmentation approach, which can map varied length data into a fixed number of data chunks with fixed lengths.
- A flexible sequence-to-one modeling framework, which can model complete temporal information with a flexible combination of different modules to cope with general sequence-to-one tasks.

The rest of the paper is organized as follows. Section 2 discusses the research background and related work. Section 3 presents the proposed framework, providing detailed explanations of its components. Section 4 presents the experimental setup used to train and test our approach, including the database, acoustic features, and implementation details. Section 5 describes the experimental results on the MSP-Podcast corpus, comparing our proposed method with baseline models. Section 6 evaluates the generalization of the proposed framework, presenting results with two different emotional corpora. Finally, Section 7 presents the concluding remarks and future directions of this study.

2 Background

2.1 Segmentation Approaches in SER

Various studies in SER have adopted the concept of modeling a sentence at the chunk or segment level [28], [29]. Chunk-level SER forces the model to learn short-term subsequences by dividing original sequences of arbitrary length into short segments with a predefined fixed step size (i.e., the overlapping area between segments). Typically, it combines these subsequences (i.e., chunk-level) prediction outputs into a sentence-level representation according to a mean pooling layer or a majority vote rule.

Han *et al.* [30] formed their segment-level features by stacking neighboring LLD frames to train a *deep neural network* (DNN). The outputs of the trained segment-level classifier were used to estimate the probability of each emotion for that segment. This approach created probability curves for the emotions in a sentence. Finally, they estimated statistics, such as the mean over these curves, which were used as the sentence-level feature representations of a static classifier implemented with an *extreme learning machine* (ELM). Mao *et al.* [31] proposed a similar approach, consisting of a segment-level classifier with a CNN model. They demonstrated the improved performance of segment-level models by comparing to results with the ones obtained by modeling the entire sentence. Tzinis and Potamianos [32] employed HLDs to represent segment-wise global features from LLDs, which obtained better performance compared to LLDs under a LSTM model. Tarantino *et al.* [33] and Sahoo *et al.* [34] found that a smaller step size (i.e., more overlap between chunks) can increase the discrimination of the feature representation in the network. These results showed that chunk-based SER can lead to better performance than sentence-based approaches.

These studies set the step size of the chunks as a fixed parameter, resulting in a varied number of segments in a sentence depending on its duration. We argue this segmentation approach is not optimal, since it restricts the aggregation of chunk-level predictions into limited static methods such as concatenating with a mean pooling layer. Our proposed segmentation approach aims to generate data chunks by varying the step size of the chunk as a function of the duration of the utterances, producing a fixed number of segments for different sentence durations.

2.2 Attention Models in SER

One essential component of the proposed framework is the fusion of information in the chunks through attention models, which have been widely used in SER [35], [36], [37], [38]. The most common way to apply this method is by building an attention model using frame-level features. Attention models are often implemented with RNN-based networks. The inputs of the attention models are activations of intermediate hidden layers in the network. This approach produces attention weights per frame, facilitating models to capture emotionally salient instances. This approach obtains an attention vector to recognize emotions [36], [37], [38]. Recent self-attention SER models based on scaled dot-product attention layers have become popular, since they can be implemented even with only fully connected layers, reducing the computational complexity of the models. This model improves prediction performance, while taking into account computational efficiency [33], [39], [40]. In contrast to previous studies constructing attention models at the frame-level, our formulation constructs attention models at the chunk-level. Therefore, it reduces the computational cost since the number of chunks is fixed and significantly lower than the number of frames.

3 Proposed Framework

This study proposes a flexible framework, which can cope with general speech-based sequence-to-one learning tasks such as SER, speech gender detection, or speaker recognition. The framework solves the dynamic temporal modeling of varied length sequences, which is a critical problem for sequence-to-one learning. We model the complete temporal information of any input sequence, without relying on cropping or zero padding techniques, by mapping a varied duration (or length) sequence into a fixed number of small chunks with a fixed size. Fig. 1 shows the generic formulation that we propose. The core component of the framework is the proposed chunk-based segmentation process, which enables the use of various effective chunk-level temporal aggregating models, achieving competitive sentence-level recognition performances. This section presents the detailed descriptions of each block of our system.

3.1 Feature Extraction

The first block in our framework consists of extracting frame-based acoustic features. Our formulation is flexible, where different acoustic features can be used, including common feature sets, spectrogram, or LLDs. Alternatively, we can also use raw waveforms. Typically, these raw features are utilized to build end-to-end deep learning systems.

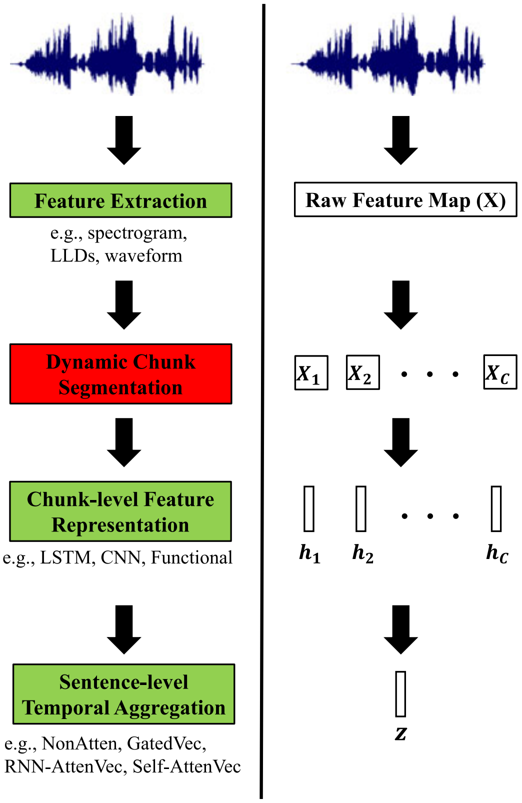

Feature Extraction
e.g., spectrogram, LLDs, waveform
Dynamic Chunk Segmentation
Chunk-level Feature Representation
e.g., LSTM, CNN, Functional
Sentence-level Temporal Aggregation
e.g., NonAtten, GatedVec, RNN-AttenVec, Self-AttenVec
Raw Feature Map (X)
X_1 X_2 · · · X_C
h_1 h_2 · · · h_C
z

Fig. 1. Diagram of the proposed framework. The left side of the figure shows the four system components of our approach. Blocks in green represent flexible components, which can be implemented with different alternative methods. The core part of the framework is the dynamic chunk segmentation (tagged in red), which enables arbitrary combinations across blocks. The right side of the figure shows the corresponding hierarchical aggregation of the model (speech, frames, chunks, and sentence) and the notation.

The LLDs consist of frequency, amplitude, and spectral-related features (e.g., fundamental frequency, energy, and MFCCs). These acoustic features are extracted within a small window (i.e., frame) from the original audio signal. The size of the small window w_{len} and window hop size Δw_{len} are fixed parameters during the feature extraction procedure. We denote the frame-based feature map as $X \in \mathbb{R}^{M\times d}$, where M is the arbitrary number of frames, depending on the duration of the speech signal, and d is the dimension of the acoustic feature.

3.2 Chunk-Based Segmentation Process

The second block is the chunk-based segmentation process, which aims to split the feature map X of varied length into a fixed number of data chunks that have the same fixed duration. This is the key step in our formulation, providing the flexibility to use different chunk-level representation methods (Section 3.3), and different sentence-level temporal aggregation approaches (Section 3.4).

There are two parameters that need to be defined in this process. The first variable is the desired length for the chunk window w_c. This variable should be big enough to preserve reliable emotional information, but small enough to process short sentences. We discuss how to set this variable in Section 5.6. The second parameter is the maximum sentence duration of the corpus $T_{\max} = \max\{T_1, T_2, \ldots, T_i, \ldots, T_N\}$,

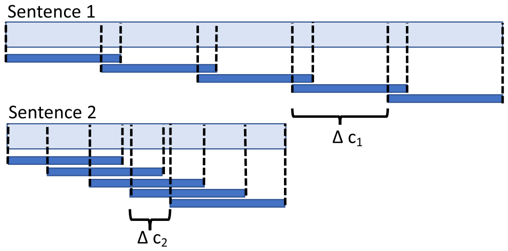


Fig. 2. Proposed chunk-based segmentation to split sentences of different durations into C chunks with fixed duration (w_c). We achieve this goal by adjusting the chunk step size (Δc_i).

where T_i denotes the duration of sentence i. Notice that we can intentionally set a bigger value for $T_{\max}$ to avoid problems of sentences with durations longer than expected. However, this study makes the assumption that we know the maximum duration of the train and the test samples, which are the same. This assumption is reasonable since a system would typically set a maximum response time to reduce latency (i.e., return a result within $T_{\max}$ secs), which naturally restricts the maximum length of the input sentences. $T_{\max}$ cannot be too long for SER tasks, since emotions may change within the segment. In this case, a single label or prediction would not properly characterize the emotion in the long segment.

We can use $T_{\max}$ and w_c to estimate a fixed number of chunks C, according to Eq. (1)

$$C = \left\lceil \frac{T_{\max}}{w_c} \right\rceil. \tag{1}$$

We obtain a fixed number of chunks for each sentence by dynamically changing the overlap between chunks. The step size of the chunks Δc_i for sentence i is defined by Eq. (2). As we increase C, Δc_i decreases, resulting in more overlap between chunks

$$\Delta c_i = \frac{T_i - w_c}{C - 1}. \tag{2}$$

Fig. 2 visualizes the proposed approach for two sentences with different durations. The key difference between them is the chunk step size Δc_i (i.e., the overlap between chunks). This approach is able to split sentences of different durations into a fixed number of chunks C that have the same duration w_c by adjusting the chunk step size. We denote these data chunks as $\{\boldsymbol{X_1}, \boldsymbol{X_2}, \ldots, \boldsymbol{X_C}\}$ where $\boldsymbol{X_j} \in \mathbb{R}^{m \times d}$. The dimension m is a fixed number, which indicates the number of frames within a chunk window w_c. Notice that the unit of variables w_c, T_i and Δc_i are in seconds. We provide the values for all the parameters used in our evaluation in Section 4.3.

3.3 Chunk-Level Feature Representation

The third block in our framework is to extract the chunk-level feature representation for each of these data chunks $\{\boldsymbol{X_1}, \boldsymbol{X_2}, \ldots, \boldsymbol{X_C}\}$. There are two clear advantages of extracting feature representations from chunks. First, the size of the chunk is fixed which simplifies deep learning architectures. Second, this step can be parallelized, estimating the feature representation for the C chunks at the same time.

TABLE 1
Architectures of Different Chunk-Level Feature Representation Models

Functional model			LSTM model		
Layer	Dimension	Activation	Layer	Dimension	Activation
Input	$1 \times 15d$	N/A	Input	$m \times d$	N/A
Linear	$1 \times b$	ReLU	LSTM	$m \times b$	Tanh
			LSTM	$1 \times b$	Tanh

CNN model					
Layer	Channels	Kernel	Stride	Dimension	Activation
Input	N/A	N/A	N/A	$m \times d$	N/A
Permute	N/A	N/A	N/A	$d \times m$	N/A
CNN-block	128	(1,3)	1	depends	ReLU
CNN-block	128	(1,3)	1	depends	ReLU
CNN-block	64	(1,3)	1	depends	ReLU
CNN-block	64	(1,3)	1	depends	ReLU
CNN-block	32	(1,3)	2	depends	ReLU
Flatten	N/A	N/A	N/A	depends	N/A
Linear	N/A	N/A	N/A	$1 \times b$	ReLU

While various methods can be applied, we focus on the three most common approaches used in previous SER studies: 1) statistical functions, 2) LSTM, and 3) CNN. The three approaches use LLDs as inputs, which we describe in Section 4.2. We present the model architectures in Table 1.

For the statistical functions, we adopt HLDs extracted from LLDs to obtain chunk-level vector representations, without relying on deep learning structures. These vectors are the statistical descriptions over acoustic features obtained for each data chunk (e.g., mean of fundamental frequency, kurtosis of energy). We apply a total of 15 HLDs for the functional representation in this study: mean, max, min, std, median, argmax, argmin, skew, kurtosis, 99th percentile, 1st percentile, range (99th,1st), 75th percentile, 25th percentile and interquartile range. Then, we pass these functional vectors through a linear layer with ReLU activation, mapping the HLDs into a b-dimensional chunk-level feature representation.

For the LSTM model, we feed the LLDs obtained for each chunk into two consecutive LSTM layers with b nodes and dropout regularization. The dropout nodes are imposed on the linear transformation of the inputs with a rate $p = 0.5$. Note that we do not drop nodes for the linear transformation of the recurrent state. Then, we exploit the final time step output of the second LSTM layer as the chunk-level feature representation.

For the CNN model, we use 1D convolution over the LLDs. We use 1D convolution instead of 2D convolution since the feature map created by the LLDs does not necessarily have spatial relationships. Therefore, the model implemented with 1D CNN can focus on temporal feature information. As Table 1 shows, the CNN model has an encoder-like architecture. The CNN block consists of a 1D-CNN, a BatchNorm, and a ReLU layer. The output dimensions of each CNN block depends on the length of the chunk window w_c and the CNN padding mode. We do not use dilated kernels or padding in the CNN layers. Table 1 shows other selected parameters (e.g., number of channels, kernel size, stride). After flattening the output of the final CNN layer, we add a linear layer with ReLU activation to map the dimension of the vector into a fixed size b-dimensional chunk-level feature representation.

The chunk-level feature representation from the input data chunks $\{\boldsymbol{X_1}, \boldsymbol{X_2}, \ldots, \boldsymbol{X_C}\}$ produced by the three methods is denoted by $\{\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_C}\}$, where $\boldsymbol{h_t} \in \mathbb{R}^{1\times b}$.

3.4 Sentence-Level Temporal Aggregation

The forth block in our framework is aggregating the temporal information across chunks. Having a fixed number of chunks per sentence, regardless of its duration, simplifies the aggregation of temporal information to form the final sentence-level feature representation. A key advantage of aggregating temporal information with this approach is the computational efficiency, since the number of chunks is significantly lower than the number of frames in a sentence. Therefore, our chunk-based approach is more efficient than frame-based aggregation models.

The goal of this block is to combine the C chunk-level feature vectors $\{\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_C}\}$ into a single sentence-level feature representation $\boldsymbol{z}$, where $\boldsymbol{z} \in \mathbb{R}^{1\times b}$. Several approaches can be used. This study explores the following four alternative methods:

NonAtten. We directly average the C chunk-level feature vectors to obtain the sentence-level representation $\boldsymbol{z}$ (Eq. (3)). This approach corresponds to a mean pooling layer

$$\boldsymbol{z} = \frac{1}{C}\sum_{t=1}^{C} \boldsymbol{h_t}. \tag{3}$$

GatedVec. An alternative approach is the gated mechanism [41]. This approach controls the information flow from different channels, in our case, chunks. Eq. (4) shows this operation, which consists of a trainable sigmoid *neural network* (NN) layer (W, b) and a point wise multiplication operation. By concatenating the gate model after the chunk-level vectors, we can produce the gating weights g_t (scalar), which ranges from 0 to 1. Eq. (5) computes the weighted average vector for the sentence-level representation $\boldsymbol{z}$

$$g_t = \sigma(W \cdot \boldsymbol{h_t} + b) \tag{4}$$

$$\boldsymbol{z} = \sum_{t=1}^{C} g_t \boldsymbol{h_t}. \tag{5}$$

RNN-AttenVec. This approach relies on attention models. We first stack $\{\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_C}\}$ into a chunk-level hidden feature map $\boldsymbol{H} \in \mathbb{R}^{C\times b}$, and feed $\boldsymbol{H}$ into an attention model formed with a vanilla RNN layer. The attention model is trained to produce the attention weights α_t by using the *general* score function presented in Luong *et al.* [42]. These attention weights are then utilized to multiply the corresponding time step's hidden states $\{\overline{\boldsymbol{h}}_1, \overline{\boldsymbol{h}}_2, \ldots, \overline{\boldsymbol{h}}_C\}$, where $\overline{\boldsymbol{h}}_t \in \mathbb{R}^{1\times q}$, resulting in the weighted summation vector $\boldsymbol{v}$ (i.e., context vector) in Eq. (6). The dimension q is the number of nodes in the RNN attention model. Finally, we concatenate the vector $\boldsymbol{v}$ with the last hidden state $\overline{\boldsymbol{h}}_C$, passing it through a NN layer (W) with the $tanh$ activation function to obtain a sentence-level feature representation $\boldsymbol{z}$ (Eq. (7)). Since the time steps in the RNN layer are fixed to C (i.e., attention to chunks rather than attention to all the input frames), the attention model is very computationally efficient

$$\boldsymbol{v} = \sum_{t=1}^{C} \alpha_t \overline{\boldsymbol{h}}_t \tag{6}$$

$$\boldsymbol{z} = tanh(W[\boldsymbol{v}; \overline{\boldsymbol{h}}_C]). \tag{7}$$

Self-AttenVec. The last alternative method relies on self-attention using the *multi-head* (MH) attention structure [43]. The MH attention model consists of several scaled dot-product attention layers running in parallel. The scaled dot-product attention (Eq. (8)) is formulated by the variables query (Q), key (K), value (V), and a scaling factor d_k. The first softmax term produces attention weights applied on the value V (Eq. (8)). Eq. (10) shows the MH model concatenated with multiple single heads, where the inputs V, Q and K are projected by different trainable parameter matrices (i.e., matrices W_j^Q, W_j^K and W_j^V in Eq. (9)). Lastly, the concatenated output of the heads is mapped into a matrix with the same dimension as the input V by the output parameter matrix W^O, so we can perform residual connections with the input. An advantage of the MH attention over the RNN-based attention model is the improved computational efficiency, since this model does not require any recurrent layer

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{8}$$

$$Head_i = Attention(QW_j^Q, KW_j^K, VW_j^V) \tag{9}$$

$$MH(Q, K, V) = Concat(Head_1, \ldots, Head_h)W^O. \tag{10}$$

We use the stacked hidden feature map $\boldsymbol{H}$ as the self-attention input (i.e., the V, Q and K variables are equal to the same $\boldsymbol{H}$ matrix). Note that we do not apply positional encoding since we are focusing on a sequence-to-one problem. Finally, we average the output attention matrix $\widetilde{\boldsymbol{H}} \in \mathbb{R}^{C\times b}$ along the C axis to obtain the sentence-level representation $\boldsymbol{z}$

$$\boldsymbol{z} = \frac{1}{C}\sum_{t=1}^{C} \widetilde{\boldsymbol{H_t}}, \tag{11}$$

where $\widetilde{\boldsymbol{H_t}}$ is the t-th row of matrix $\widetilde{\boldsymbol{H}}$. An additional advantage of chunk-level self-attention is that we map the original arbitrary length sequence into a fixed length equal to C. This approach significantly reduces the sequence length, which is typically the part that contributes the most to the complexity of the scaled-dot product attention model [44].

After obtaining the sentence-level representation vector $\boldsymbol{z}$, we feed it into the output layer, which we implement with two fully connected layers. The final output of the network is a prediction score for either arousal, dominance, or valence.

4 Experimental Settings

4.1 MSP-Podcast Corpus

This study relies on the MSP-Podcast corpus [10] to build and evaluate our proposed approach. The dataset consists of spontaneous speech turns that are rich in emotional content from various online audio-sharing podcast websites under Creative Commons licenses. The podcasts include spontaneous discussions on a variety of topics including politics, sports, entertainment, art, technology and economics, providing broad, rich, and natural emotional displays. The collected podcasts are processed to identify clean audio without music, noise, overlapping speech, or multiple speakers in the background [45]. The dataset provides both categorical and attribute-based emotional annotations, which are labeled by at least five annotators for each speech segment using a crowdsourcing approach [46]. We evaluate our models using regression tasks to predict the emotional attributes of arousal (calm versus active), valence (negative versus positive), and dominance (weak versus strong). The ground truth labels are the average of the scores across different annotators. We use version 1.6 of the corpus which has 50,362 speaking turns (83h29m). We have identified the speaker identity for 42,567 sentences belonging to 1,078 speakers. We use the speaker information to define the partitions for the train (34,280 speech turns), development (5,958 speech turns) and test (10,124 speech turns) sets. The partitions aim to define speaker independent sets. The readers are referred to Lotfian and Busso [10] for more details on this corpus.

We also consider the IEMOCAP [11] and MSP-IMPROV [47] databases to validate the results with other emotional corpora in Section 6. The IEMOCAP and MSP-IMPROV databases are multimodal emotional corpora that have been widely used in SER studies. Both databases are designed to elicit spontaneous emotional expression with dyadic interactions between actors. These recordings are collected under controlled laboratory environments. Therefore, there are inevitable domain mismatches between these corpora and the MSP-Podcast corpus, which makes them ideal candidates to evaluate the generalization of our proposed framework. Both of these corpora provide attribute-based annotations for arousal, valence, and dominance.

4.2 Acoustic Features

For the acoustic features, we extract the LLD feature set proposed for the Interspeech 2013 computational paralinguistics challenge [13] using the OpenSmile toolkit [48]. The window length w_{len} is set to 32ms, and the window step size Δw_{len} is set to 16 ms (50 percent overlap). In total, the set includes 130 frame-based acoustic features ($d = 130$), which are normalized by subtracting the mean and dividing by the standard deviation. The parameters of this normalization are estimated over the training set. Therefore, the output feature map X for each sentence is a $M \times 130$ normalized LLD matrix, where the number of frames M depends on the duration of the sentence.

4.3 Implementation

Each emotion attribute is regarded as an independent sequence-to-one task building separate models for arousal, valence and dominance. We implement our approach using chunks of 1 sec (w_c= 1). Studies have shown that segments as short as 0.5 secs can be used in SER tasks [49], so 1 sec is a good compromise. Section 5.6 presents the performance of the system by using chunks of different durations. Therefore, the fixed number of frames m within each chunk window is 62 ($16ms \times 62 \approx 1sec$). Since the duration of the sentences is between 2.75 and 11 secs for the MSP-Podcast corpus [10], $T_{\max}$ is 11 secs. The number of chunks C is 11, according to Eq. (1). The value of the step size for the chunks Δc_i depends on the duration of the sentence T_i (Eq. (2)). For example, if the duration of the input sentence is $T_i = 6$ secs, then Δc_i is 0.5 secs. As a result, we split every sentence into fixed 11 chunks with 1 sec length for each sentence regardless of its duration. The dimension of each data chunk $\boldsymbol{X_t}$ is fixed to 62×130 (i.e., $m \times d$).

For the network settings, we fixed the number of nodes for all layers, matching the dimensions of the LLDs (i.e., $d = b = q = 130$). Our multi-head attention model consists of three heads, where the output dimension of each head is 50 (i.e., d_k=50 in Eq. (8)). We use the Adam optimizer with a batch size of 128, 256 and 512 for the LSTM, CNN and functional models, respectively. The number of training epochs is fixed to 100, which is sufficient for convergence of all models. We save the best models with an early stopping criterion based on the development loss. The cost function optimizes the *concordance correlation coefficient* (CCC). We also report the accuracy of our prediction in the testing set in terms of CCC. We randomly split the original test set into 15 small subsets with similar size, reporting the average results. We implement this strategy to conduct a statistical analysis using a two-tailed t-test over the 15 subsets in the test partition. We define statistical significant at p-value = 0.05. All models are implemented in Keras.

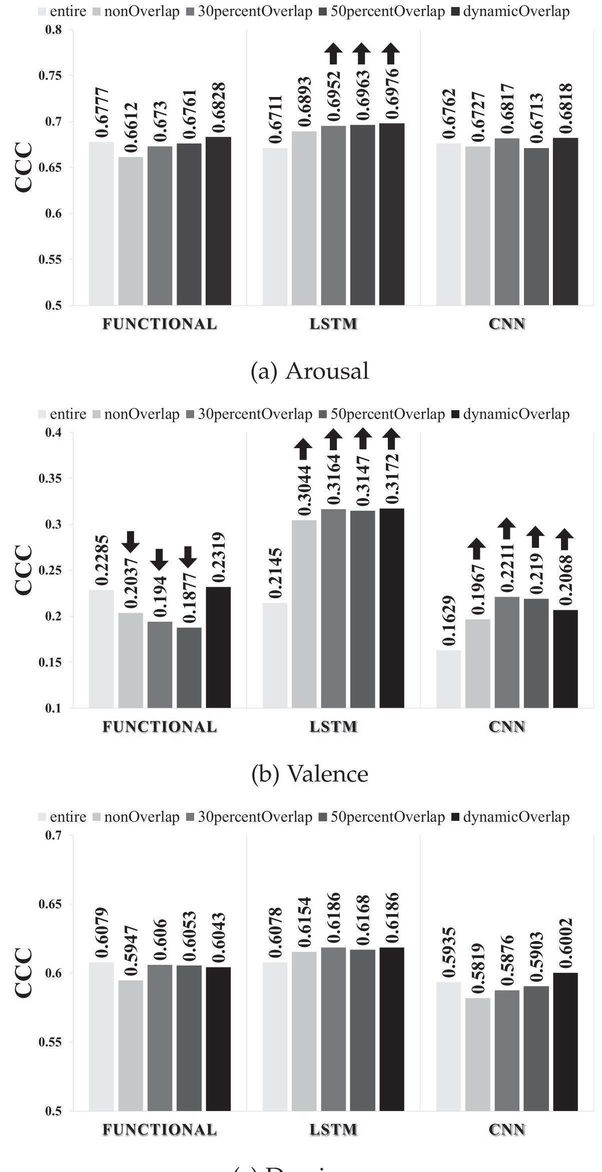
5 Experimental Results and Analysis

5.1 Preliminary Study of Chunk-Level SER

We first evaluate the role of splitting a sentence into small chunks using alternative methods. We compare these approaches with our proposed solution. The ultimate goal of a sequence-to-one SER task is to encode the emotional information of the entire sentence into a single vector. We expect that this vector is able to capture emotionally-relevant content in the input speech sentence. To achieve this goal, studies usually pad zeros to the sequences to match the maximum length for the batch training. An obvious issue is the poor capacity toward long sequences since a single fixed dimension vector may not be sufficient to represent such complex, long, and temporal dynamic information [50].

An alternative method to avoid the model directly learning long sequences is to split the original sequence into small segments (chunks) through cropping and padding techniques [33], [34]. Similar to the proposed approach, we need to set a desired chunk window length w_c. For the baseline, we use a *fixed* chunk step size Δc, which is the conventional approach. Note that the number of chunks per sentence still varies as a function of the duration of the sentence. During the training stage, each data chunk is treated as an independent training sample sharing the same sentence-level label. In the inference stage, the final sentence score is the average of these chunk-level predictions.

Exploring the aforementioned methods, we compare our dynamic chunk-based segmentation process with alternative baselines. We present the baseline results in Fig. 3 for LSTM, CNN, and functional models. The model architectures are the same as the ones presented in Table 1, which



(a) Arousal

(b) Valence

(c) Dominance

Fig. 3. Preliminary results of feature extraction models with different chunking methodologies. The symbols ↑ and ↓ indicates that the results are statistically significantly better or worse that the *entire* baseline, which processes the sentences without chunk segmentation.

are directly combined with two fully connected output layers without attention models. The settings to train and evaluate the network are described in Section 4.3 (e.g., batch size and cost function). The approaches considered in this study are:

(1) *Sentence-level SER*
   - entire: extracting the sentence-level representation vector directly from the entire sentence. These sequences are zero padded to reach the maximum length of the sentences in the corpus (i.e., 11 secs).

(2) *Chunk-level SER*
   - nonOverlap: splitting the sentence into chunks without overlap. We set the length of the chunk to w_c = 1 sec, using the same value used by our method (Section 4.3). The chunk step size Δc is fixed to 1 sec (i.e., 0 percent overlap between chunks). The last data chunk, if shorter than 1 sec, is padded with zeros to reach a duration of 1 sec.
   - 30percentOverlap: same as *nonOverlap* but the chunk step size Δc is fixed to 0.7 sec (i.e., 30 percent overlap between chunks).
   - 50percentOverlap: same as *nonOverlap* but the chunk step size Δc is fixed to 0.5 sec (i.e., 50 percent overlap between chunks).
   - dynamicOverlap: split sentence into small chunks with a dynamic chunk step size Δc_i depending on its duration (i.e., our proposed method in Section 3.2). Note that we do not apply any sentence-level temporal aggregation model. Each chunk is independently treated, sharing the same sentence-level label during training.

Fig. 3 shows the results for arousal, valence and dominance. We denote with the symbols ↑ and ↓ cases where the chunk-level segmentation methods lead to significantly better or worse performance than the result of the sentence-level approach, respectively. The figure shows that the chunk-level SER methods often perform better than the sentence-level SER (i.e., *entire*) for LSTM architectures. The results are particularly clear for arousal and valence. This result shows that reducing the input sequence length can benefit recurrent-based models. However, we do not observe the same behavior when we rely on a CNN or a functional model. The chunk-level SER methods for the CNN model only improve results for valence. We do not observe clear benefits of using chunk-based models using statistical functions, where we even observe worse performance for valence. We notice that static encoding models such as CNN or statistical functions are not able to effectively capture emotionally-relevant features, since each small data chunk only contains local emotional information. Moreover, forcing data chunks to share the same sentence-level label is implausible, since emotions are not uniformly distributed in a sentence [51]. Another interesting finding is that the step size between chunks Δc does not play a key role in the prediction performance. Different overlaps between chunks do not drastically affect the performance. Our proposed dynamic chunk step size Δc_i can even achieve the best performance in many cases (e.g., valence results with LSTM and functional models).

We conclude two main issues of current chunk-level SER: (1) each data chunk only contains partial sentence-level information, and (2) each data chunk shares the same sentence-level emotional label. Our proposed framework can solve both issues, hierarchically extracting emotion-relevant information from the frame-level, chunk-level, and sentence-level, via a simple data segmentation process.

5.2 Proposed Chunk-Level SER Results

As we stated in previous sections, the key advantage of our proposed dynamic chunk segmentation process is the fixed number of output chunks, allowing us to aggregate complete sentence-level temporal information by different techniques in an efficient way. This section compares implementations of our proposed framework with two models presented in Section 5.1: *entire* and *dynamicOverlap*. We use

TABLE 2
Framework Performance Using Different Combination of Chunk-Level Feature Representation and Sentence-Level Temporal Aggregation Methods

	Aro [CCC]	Val [CCC]	Dom [CCC]
	Functional Model		
entire	0.6777	0.2285	0.6079
dynamicOverlap	0.6828	0.2319	0.6043
NonAtten	0.6992^{*}	$0.2585^{*\dagger}$	0.6224
GatedVec	$\mathbf{0.7038}^{*\dagger}$	$\mathbf{0.2942}^{*\dagger}$	0.6236
RNN-AttenVec	0.6666	0.1835	0.5955
Self-AttenVec	0.6987^{*}	$0.2679^{*\dagger}$	**0.6253**
	LSTM Model		
entire	0.6711	0.2145	0.6078
dynamicOverlap	**0.6976**	0.3172	**0.6186**
NonAtten	0.6807	0.3275^{*}	0.6085
GatedVec	0.6771	0.3141^{*}	0.6011
RNN-AttenVec	0.6955^{*}	0.3006^{*}	0.6175
Self-AttenVec	0.6837	$\mathbf{0.3337}^{*}$	0.6004
	CNN Model		
entire	0.6762	0.1629	0.5935
dynamicOverlap	0.6818	0.2068	0.6002
NonAtten	$\mathbf{0.7035}^{*\dagger}$	$0.2683^{*\dagger}$	$\mathbf{0.6268}^{*\dagger}$
GatedVec	$0.7027^{*\dagger}$	$\mathbf{0.2856}^{*\dagger}$	$0.6201^{*\dagger}$
RNN-AttenVec	0.6845	0.1582	0.5885
Self-AttenVec	0.7012^{*}	$0.2310^{*\dagger}$	$0.6207^{*\dagger}$

The symbols $*$ and † indicate that the improvements of our methods over the entire *and* dynamicOverlap *baselines are statistically significant, respectively (two-tailed t-test, p-value < 0.05).*

dynamicOverlap as the representative chunk-level baseline model, since the performances of chunk methods with different overlaps are similar. In addition, the key difference between *dynamicOverlap* and our proposed framework is the addition of the sentence-level temporal aggregation. Therefore, it is straightforward to compare the effectiveness of modeling complete temporal information with the methods described in Section 3.4.

Table 2 shows the results for different combinations of chunk-level feature representation models and sentence-level temporal aggregation methods. For the LSTM models, Table 2 shows that the results for chunk-level methods are significantly better than directly learning the entire sequence for all emotional attributes (especially for valence). Although arousal and dominance have similar accuracy with *dynamicOverlap* (i.e., the differences are not statistically significant), our proposed *Self-AttenVec* method achieved the best valence CCC result (CCC=0.3337). Valence is an attribute that is particularly challenging to predict with acoustic features [52], [53], indicating that complete sentence-level information can bring complemental benefits for more complex tasks. The advantage of applying attention models is amplified in the CNN and functional models. Table 2 shows that the models often obtain significantly better performance than the *entire* and *dynamicOverlap* baselines for all emotional attributes (e.g., see results for *GatedVec* model). These results verify that aggregation of sentence-level temporal knowledge is necessary and particularly important for the static CNN or functional-based model.

The results show that we cannot find a general combination of our frameworks which is able to reach the best performance for all different tasks and models (chunk-level feature representation and sentence-level temporal aggregation models). For instance, the *RNN-AttenVec* model gives sub-optimal results when the features are extracted with either CNN or functional models. This result demonstrates that the combination of framework modules should be task or model dependent. Based on the results in Table 2, we suggest the sentence-level temporal aggregation models of *RNN-AttenVec* or *Self-AttenVec* for the LSTM model, and *NonAtten* or *GatedVec* for the CNN and functional models. We keep only these combinations for the rest of the evaluations in this study. Notice that the CCC improvements reported in this section are only possible after using the proposed chunk-based segmentation.

5.3 Analysis of Chunk-Level Attention Weights

We analyze the chunk-level weights of the sentence-level temporal aggregation methods. We select the *RNN-AttenVec* model to analyze the LSTM model. The attention weights α_t in the *RNN-AttenVec* model (Eq. (6)) are the softmax output of the *general* score function [42], so they are constrained to sum to 1. For the CNN and functional models, we use the *GatedVec* weights for the analysis (g_t in Eq. (5)). Each weight is the sigmoid output value of the gated model corresponding to each input chunk. Its value ranges from 0 to 1. Other temporal aggregation methods are not considered. For the *NonAtten* framework, the weights are constant across chunks so this analysis is not relevant. Although the *Self-AttenVec* method has the same softmax attention weights as *RNN-AttenVec*, we only present the analysis for the *RNN-AttenVec* method as the representation of the analysis for this type of attention weights in the LSTM model since its performances are generally better than the performances of the *Self-AttenVec* model (see Table 2 for arousal and dominance). Our models produce attention weights based on chunk-level representations, which are used to combine the vectors $\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_C$. The number of weights per sentence is fixed to $C = 11$ under our experimental settings, which facilitate the visualization of the weights. The purpose of this analysis is to evaluate if the models are assigning different weights to different chunks within the sentences.

Fig. 4 presents the results. The dots inside the solid line represent the mean weights. The shaded area around the lines indicates the weight's standard deviation across the entire testing set for each specific chunk. The first interesting point we can notice in Fig. 4 is that the attention weights of LSTM-based *RNN-AttenVec* present a decreasing trend along with time for all emotional attributes, indicating that the first chunks are, in general, weighted more. The results show the importance of the first impression in the decision making process [54]. The second observation is the high value of the standard deviations, which shows that the values of the weights are different from sentence to sentence. For both CNN and functional-based *GatedVec* models, we observe different patterns for emotional attributes. For valence, the gated weights have high deviations and the same decreasing trend as the LSTM-based *RNN-AttenVec* model. For arousal and dominance, we observe a flat-shape

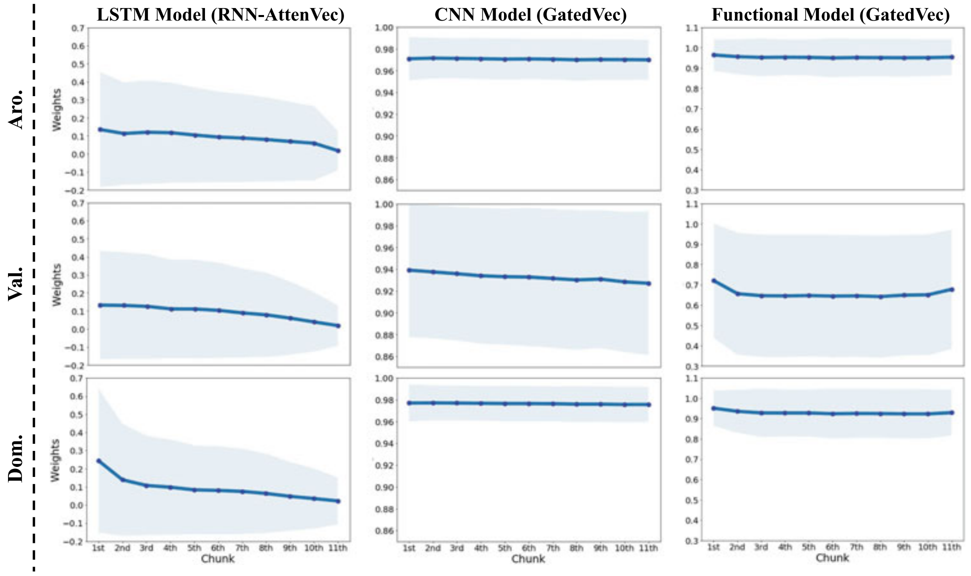


Fig. 4. Analysis of the chunk-level attention weighs for *RNN-AttenVec* implemented with LSTM, and *GatedVec* implemented with CNN and functional models. The solid lines represent the average score for each of the $C = 11$ chunks and the shaded area represents the standard deviation.

with small deviations suggesting that the weights of chunks are similar without much variation across sentences. This result suggests that keeping the weights constant for arousal and dominance (i.e., *NonAtten* model) may be sufficient to obtain similar performance to the *GatedVec* model. This result is consistent with the CCC values observed in Table 2. The high variability in the weights for valence illustrates the benefits of adding temporal modeling in its prediction [55].

5.4 Results as a Function of Sentence Duration

This section analyzes additional advantages of the proposed chunk-level SER framework by analyzing the CCC performance as a function of the duration of the sentences. We compare the selected models to a sentence-level baseline implemented with zero padding to fix the duration of the sentence (*entire* model presented in Section 5.1). To evaluate the model performance for sentences of different lengths, we arbitrarily split the test set into short ($\leq$ 5sec), medium (5-8sec) and long ($\geq$ 8sec) subsets based on the duration of the sentences. The test set has 4,280 short, 3,684 medium, and 2,160 long sentences. Following the approach presented in Section 4.3, we further split each of these test sets (i.e., short, medium and long sets) into 15 subsets, reporting the average scores, and estimating statistical significance with a two-tailed t-test.

Table 3 shows the results of the selected LSTM, CNN, and functional models. The first observation is that the performance of the *entire* model degrades as we increase the duration of the sentences for all three emotional attributes. The result shows unstable performances for sentences of different durations, since zero padding introduces artifacts in the feature vector affecting the temporal model. The same trend can be observed when this baseline is implemented with either LSTM, CNN, or functional models. The results verify the challenges in modeling long sequences using the zero padding technique while building the sentence-level SER (see the results for medium and long subsets, especially for the valence attribute). However, our proposed models systematically improve the performance for different duration of the data, especially for medium and long sequences. These results demonstrate that temporal modeling based on smaller chunks can be useful to aggregate long-term temporal information, leading to robust prediction accuracy, regardless of the duration of the sentences.

5.5 Analysis of Computational Benefits

As we set the desired chunk window length w_c in Section 3.2, we generate C chunks per sentence. Each chunk has m frames. The variable m depends on the chunk window length (w_c) and step size (Δw_{len}) during the feature extraction stage (Section 3.1). Eq. (12) gives this relationship

$$w_c = m \times \Delta w_{len}. \tag{12}$$

Since we feed these data chunks into a LSTM model, the GPU is able to process the chunks in parallel. We can observe that the time step size in the *backpropagation through time* (BPTT) and RNN-based forward algorithm reduces from an arbitrary number M (i.e., total number of frames in a sentence) to a fixed variable m in Eq. (13), where E is the prediction loss, $\hat{y}$ is the network prediction outputs, s is the consecutive time-step hidden outputs and W is the network trainable parameters

$$\frac{\partial E_m}{\partial W} = \sum_{k=0}^{m} \frac{\partial E_m}{\partial \hat{y}_m} \frac{\partial \hat{y}_m}{\partial s_m} \left(\prod_{j=k+1}^{m} \frac{\partial s_j}{\partial s_{j-1}} \right) \frac{\partial s_k}{\partial W}. \tag{13}$$

Typically m is significantly less than the original total number of frames, which can effectively improve the computational efficiency of the RNN-based models. For instance, if the duration of the input sentence is 10 seconds, with a LLDs extraction step size equal to $\Delta w_{len} = 0.016$ seconds (16ms), we will obtain 625 frames for the LLDs feature map. Now, if we set w_c to 1 second, m will approximately equal to 62, reducing the time step size from 625 to 62. The

TABLE 3
Analysis of Performance as a Function of the Sentence Duration

LSTM Model			
	Short($\leq 5sec$)		
	Aro [CCC]	**Val [CCC]**	**Dom [CCC]**
entire	0.6800	0.2497	0.6188
RNN-AttenVec	0.7010*	0.3246*	0.6235
Self-AttenVec	0.6809	0.3494*	0.6031
	Medium($5 \sim 8sec$)		
	Aro [CCC]	**Val [CCC]**	**Dom [CCC]**
entire	0.6622	0.1838	0.5984
RNN-AttenVec	0.6870	0.2876*	0.6149
Self-AttenVec	0.6794	0.3337*	0.5955
	Long($\geq 8sec$)		
	Aro [CCC]	**Val [CCC]**	**Dom [CCC]**
entire	0.6565	0.1836	0.5828
RNN-AttenVec	0.6862	0.2452*	0.5951
Self-AttenVec	0.6892*	0.2775*	0.5905
CNN Model			
	Short($\leq 5sec$)		
	Aro [CCC]	**Val [CCC]**	**Dom [CCC]**
entire	0.6947	0.1994	0.6160
NonAtten	0.7134	0.3159*	0.6332
GatedVec	0.7059	0.3326*	0.6262
	Medium($5 \sim 8sec$)		
	Aro [CCC]	**Val [CCC]**	**Dom [CCC]**
entire	0.6622	0.1440	0.5825
NonAtten	0.6960*	0.2259*	0.6281*
GatedVec	0.6995*	0.2473*	0.6172*
	Long($\geq 8sec$)		
	Aro [CCC]	**Val [CCC]**	**Dom [CCC]**
entire	0.6545	0.1244	0.5440
NonAtten	0.6814	0.2061*	0.6040*
GatedVec	0.6899	0.2122*	0.6036*
Functional Model			
	Short($\leq 5sec$)		
	Aro [CCC]	**Val [CCC]**	**Dom [CCC]**
entire	0.6939	0.2826	0.6241
NonAtten	0.7006	0.2992	0.6288
GatedVec	0.7093	0.3374*	0.6297
	Medium($5 \sim 8sec$)		
	Aro [CCC]	**Val [CCC]**	**Dom [CCC]**
entire	0.6644	0.2096	0.5956
NonAtten	0.6963*	0.2322	0.6175
GatedVec	0.7011*	0.2593*	0.6226
	Long($\geq 8sec$)		
	Aro [CCC]	**Val [CCC]**	**Dom [CCC]**
entire	0.6559	0.1406	0.5839
NonAtten	0.6922*	0.1960*	0.6053
GatedVec	0.6886	0.2364*	0.5989

The sentences are grouped into short, medium and long sentences. The symbol $*$ indicates that the improvements of our methods over the entire *baseline are statistically significant (two-tailed t-test, p-value < 0.05).*

TABLE 4
Analysis of the Computational Efficiency of the Proposed Framework for Different Implementations

Model	# of Par. [10^6]	Train [sec/epoch]	Online [ms/uttr]
LSTM entire	0.289	436.6	519.2
LSTM RNN-AttenVec	0.374	140.1	40.9
LSTM Self-AttenVec	0.368	87.5	41.3
CNN entire	1.587	76.9	1.9
CNN NonAtten	0.269	53.0	1.8
CNN GatedVec	0.269	83.9	3.0

The table lists the number of parameters, time for training, and time for online processing.

same model efficiency improvement is also observed with the CNN model. We split the original big feature map into multiple small and fixed size sub-maps, which can be processed in parallel by the GPU.

Table 4 compares the model efficiency in terms of the number of parameters, time for training (i.e., average in seconds per training epoch), and time for online processing (i.e., average in millisecond per utterance during inference). All models are trained and tested under the same single NVIDIA GeForce RTX 2080 Ti GPU environment. As we expect, even though LSTM-based chunk-level SER models are equipped with an additional attention model (i.e., *RNN-AttenVec* or *Self-AttenVec*), they considerably improve the model efficiency. Both training and online testing times have been significantly reduced, achieving models that are 10 times faster during online processing than the baseline (i.e., the *entire* model). Even though we observe an increase in the number of parameters for the LSTM models, the proposed chunk-level attention models have low complexity. For the CNN model, we observe the same efficiency improvement trend, but the relative improvements are smaller, since the convolution operation is already well parallelized in the GPU. For the CNN-based model, the number of parameters is significantly lower than the baseline model over the entire sentence. The *entire* model requires the complete sentence-level feature map, resulting in a high dimensional flatten output at the final CNN layer. This high dimension vector representation increases the complexity of the model when we add the fully connected output layers. However, chunk-level SER does not inherit this problem, since it receives small and fixed size sub-maps as the input.

5.6 Effect of the Window Size for the Chunks

While the maximum sentence duration in a corpus is fixed (T_{max}), adjusting the chunk window length w_c changes the number of chunks C. There is a tradeoff in setting the value for w_c. On the one hand, increasing w_c means more information is contained within a chunk. However, it reduces the role of the temporal aggregation model since C decreases. Furthermore, long sequences might include changes of emotions within the chunks if w_c is too large, making it more difficult to learn its ambiguous information. On the other hand, decreasing w_c can enhance the influence of the temporal aggregation model since C increases. However, short-term information may not be sufficient to reliably recognize emotion if w_c is too small. Increasing C also increases the computational complexity of the sentence-level temporal aggregation models. This section evaluates the performance of the system as a function of w_c.

Fig. 5 shows the valence results for different values of w_c (i.e., 0.5 secs, 1 sec, 1.5 secs and 2 secs). The results for arousal and dominance present similar trends so we do not present them in this paper. All settings are the same as the setting described in Section 4, where we only change the variable w_c. We only present specific combinations of the framework as representative examples for each chunk-level feature representation approach (e.g., *RNN-AttenVec* for LSTM). The figure shows that the recognition performance decreases as we increase the length of w_c to 1.5 secs or 2 secs. However,

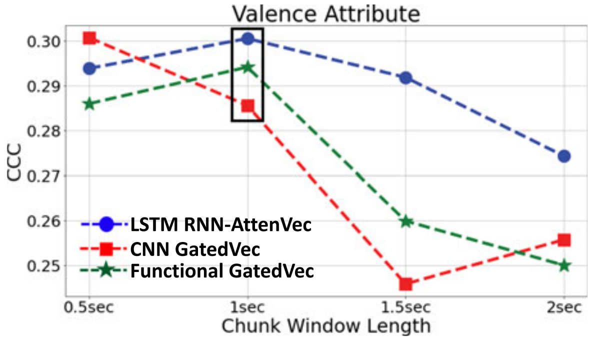
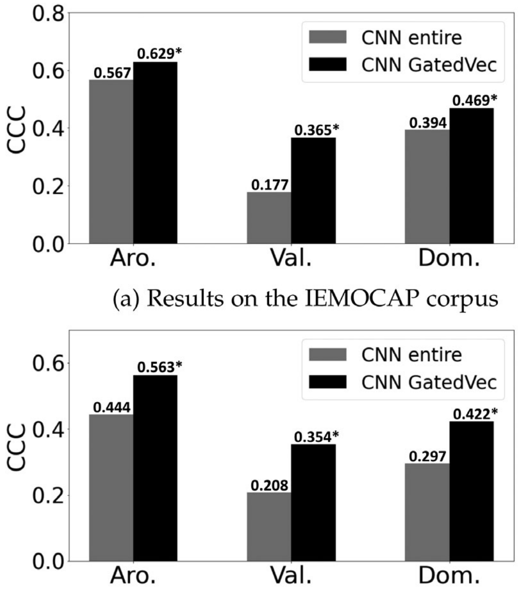


Fig. 5. Analysis of the performance of the models for valence as a function of the window length w_c. We evaluate the *RNN-AttenVec* model implemented with LSTM, and the *GatedVec* model implemented with CNN and functional models.



(a) Results on the IEMOCAP corpus

(b) Results on the MSP-IMPROV corpus

Fig. 6. Results on the IEMOCAP and MSP-IMPROV corpora. The figure compares the performance of the proposed *GatedVec* framework implemented with CNN with the *entire* baseline implemented with CNN. The CCC performances are calculated using a leave-one-session-out cross validation approach, where we randomly split the data into 15 subsets to perform the statistical test. Results tagged with $*$ indicate statistically significant improvements over the baseline.

the degradation of performance does not significantly change when we decrease w_c to 0.5 secs. We conclude that the sentence-level aggregation methods can effectively capture useful temporal information even when the chunks only contain limited emotional information. Similar results have been reported by Arias *et al.* [56], where they successfully recognized emotions using 0.5 secs windows. Based on the results, we believe that setting the chunk window length w_c to 1 sec is a good balance to obtain reliable short-term chunk information and long-term sentence-level aggregation.

We highlight that the value of w_c is task dependent. For some tasks, the target traits are conveyed across all frames. An example is basic gender acoustic traits, which do not change too much across time. Therefore, we can use a longer chunk duration to capture full gender patterns. For other tasks, such as SER, the target information varies from frame to frame. For these tasks, the duration of the chunk window should be short enough to capture temporal variations to be leveraged by the sentence-level temporal aggregation models. The value of w_c also depends on the speech style. If the recordings are spontaneous speech from a group of people, we can expect many interruptions and overlapped speech. Since these patterns might challenge the model during training, we can choose a window size with smaller duration to capture a single speaker for most of the chunks. Using a reduced window size increases the number of chunks (Eq. (1)), giving more freedom to the sentence-level temporal aggregation model to ignore trivial or confusing acoustic patterns.

6 Generalization of Proposed Approach

This section evaluates the generalization of our framework for SER systems on other emotional corpora. We consider two benchmark databases widely used in the community: the IEMOCAP [11] and MSP-IMPROV [47] databases. This section considers within-corpus evaluations, defining separate train, development, and test partitions for each database. We implement the approach for arousal, valence, and dominance. To simplify the evaluation, we only consider the CNN model (same architecture described in Table 1) with the *GatedVec* algorithm to aggregate the temporal information across chunks. We compare our approach with the *entire* sentence-level SER baseline. All model settings are the same as the ones described in Section 4.3, including the acoustic features (i.e., the 130 normalized LLDs). The collection of the IEMOCAP and MSP-IMPROV datasets did not impose strict duration range for sentences, so we artificially define $T_{max} = 17$ secs, discarding sentences with duration outside this range. We also discarded sentences that were shorter than 1 sec. Sentences with duration between 1 sec and 17 secs cover over 97 percent of the sentences for both datasets. We set $w_c = 1$ sec, resulting in $C = 17$. We implement the evaluation using a speaker-independent *cross-validation* (CV) setting, using 5 folds for the IEMOCAP corpus (5 dyadic sessions), and 6 folds for the MSP-IMPROV corpus (6 dyadic sessions). For every CV, one dyadic session (i.e., 2 speakers) is used for the test set, one dyadic session is used for the development set, and the rest of the corpus is used for the train set. Consistent with Section 4.3, we report the average CCC values obtained across 15 randomly split subsets of the CV prediction results, evaluating statistical significance with a two-tailed t-test.

The experiment results are shown in Fig. 6. Our proposed chunk-level SER method systematically outperforms the baseline model, which directly learns discriminative information from the entire sentence. This result is particularly clear for valence. The same improved performance trend can be observed in the MSP-Podcast corpus (see the CNN model in Table 2). The results from the MSP-IMPROV and IEMOCAP corpora validate the general effectiveness of the proposed framework.

7 Conclusion and Future Work

This study presented a general chunk-level sequence-to-one framework to cope with important dynamic temporal information in SER tasks. The proposed approach is able to split sentences with varied durations into a fixed number of chunks, which have the same length by dynamically adapting the overlap between chunks. The approach offers the flexibility to efficiently combine different chunk-level feature representation frameworks (e.g., functional statistics, LSTM or CNN) with alternative sentence-level temporal aggregation models (e.g., *GatedVec*, *RNN-AttenVec* or *Self-AttenVec*). The proposed framework hierarchically extracts task-relevant features at the frame, chunk, and sentence levels, providing an appealing end-to-end framework. The experimental results based on multiple databases showed the benefits in model efficiency and accuracy by using the proposed chunk-level temporal modeling methodology. The results show higher accuracies for sentences with medium and long durations, which are challenging for conventional approaches relying on zero padding. Our solution solves a critical issue for directly building sentence-level models.

The framework offers multiple potential research directions to extend this study. First, we expect that this framework can be extended to sequence-to-sequence learning tasks, providing an appealing solution when emotional information is available within a sentence (e.g., emotional traces). Estimating continuous emotion trends from a speech recording is a challenging task. Robust frame-level predictions require time-continuous traces (i.e., emotional labels) that are well synchronized with the speech signal. This synchronization is hard to achieve due to the reaction lag of the evaluators [9]. Since this framework decreases the resolution from frame-level to chunk-level analysis, the strict synchronization requirement is relaxed. Second, the framework can also be beneficial for multi-modal processing. Multimodal approaches also require synchronization between different modalities, which is usually infeasible. For example, the sampling rate for videos typically does not match the sampling rate for speech. We can increase the tolerance for timing mismatches across signals by turning a frame-level analysis into a chunk-level analysis. Third, our current framework splits a sentence into chunks based exclusively on duration. We can alternatively use other criteria to include more meaningful linguistic information. An interesting extension of this approach is to develop an advanced framework that leverages the output from an *automatic speech recognition* (ASR) system to define favorable chunk durations, considering both acoustic and linguistic information. Finally, we can apply this approach in other sequence-to-one problems.

Acknowledgments

This work was supported by the National Science Foundation (NSF) under Grants CNS-2016719 and CAREER IIS-1453781. Github:https://github.com/winston-lin-wei-cheng/Chunk-Level-Attention-SER.

References

[1] Z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," *ACM SIGKDD Explorations Newslett.*, vol. 12, no. 1, pp. 40–48, Nov. 2010.

[2] L. Deng and J. Chen, "Sequence classification using the high-level features extracted from deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2014, pp. 6844–6848.

[3] L. Martinez-Lucas, M. Abdelwahab, and C. Busso, "The MSP-conversation corpus," in *Proc. Interspeech*, Oct. 2020, pp. 1823–1827.

[4] F. Ringeval, A. Sonderegger, J. Sauer, and D. Lalanne, "Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions," in *Proc. Int. Workshop Emot. Representation, Anal. Synth. Continuous Time Space*, Apr. 2013, pp. 1–8.

[5] G. McKeown, M. Valstar, R. Cowie, M. Pantic, and M. Schröder, "The SEMAINE database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent," *IEEE Trans. Affect. Comput.*, vol. 3, no. 1, pp. 5–17, Jan.–Mar. 2012.

[6] S. Parthasarathy, R. Cowie, and C. Busso, "Using agreement on direction of change to build rank-based emotion classifiers," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 11, pp. 2108–2121, Nov. 2016.

[7] R. Cowie, G. McKeown, and E. Douglas-Cowie , "Tracing emotion: An overview," *Int. J. Synthetic Emot.*, vol. 3, no. 1, pp. 1–17, Jan.–Jun. 2012.

[8] S. Mariooryad and C. Busso, "Analysis and compensation of the reaction lag of evaluators in continuous emotional annotations," in *Proc. Affecti. Comput. Intell. Interact.*, Sep. 2013, pp. 85–90.

[9] S. Mariooryad and C. Busso, "Correcting time-continuous emotional labels by modeling the reaction lag of evaluators," *IEEE Trans. Affect. Comput.*, vol. 6, no. 2, pp. 97–108, Apr.–Jun. 2015.

[10] R. Lotfian and C. Busso, "Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings," *IEEE Trans. Affect. Comput.*, vol. 10, no. 4, pp. 471–483, Oct.–Dec. 2019.

[11] C. Busso *et al.*, "IEMOCAP: Interactive emotional dyadic motion capture database," *J. Lang. Resour. Eval.*, vol. 42, no. 4, pp. 335–359, Dec. 2008.

[12] M. Grimm, K. Kroschel, and S. Narayanan, "The Vera am Mittag German audio-visual emotional speech database," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun. 2008, pp. 865–868.

[13] B. Schuller *et al.*, "The INTERSPEECH 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism," in *Proc. Interspeech*, Aug. 2013, pp. 148–152.

[14] P. Shen, Z. Changjun, and X. Chen, "Automatic speech emotion recognition using support vector machine," in *Proc. Int. Conf. Electron. Mech. Eng. Inf. Technol.*, Aug. 2011, pp. 621–625.

[15] T. Seehapoch and S. Wongthanavasu, "Speech emotion recognition using support vector machines," in *Int. Conf. Knowl. Smart Technol.*, Jan.–Feb. 2013, pp. 86–91.

[16] M. Abdelwahab and C. Busso, "Study of dense network approaches for speech emotion recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2018, pp. 5084–5088.

[17] Q. Jin, C. Li, S. Chen, and H. Wu, "Speech emotion recognition with acoustic and lexical features," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2015, pp. 4749–4753.

[18] W. Lin and C. Lee, "A thin-slice perception of emotion? an information theoretic-based framework to identify locally emotion-rich behavior segments for global affect recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.* , Mar. 2016, pp. 5790–5794.

[19] H. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for speech emotion recognition," *Neural Netw.*, vol. 92, pp. 60–68, Aug. 2017.

[20] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *Proc. Affect. Comput. Intell. Interaction*, Sep. 2013, pp. 511–516.

[21] S. Parthasarathy and C. Busso, "Predicting emotionally salient regions using qualitative agreement of deep neural network regressors," *IEEE Trans. Affect. Comput.*, vol. 12, no. 2, pp. 402–416, Apr.–Jun. 2021.

[22] V. Chernykh, G. Sterling, and P. Prihodko, "Emotion recognition from speech with recurrent neural networks," Jan. 2017, *arXiv: 1701.08071*.

[23] Z. Zhao, Z. Bao, Z. Zhang, N. Cummins, H. Wang, and B. Schuller, "Attention-enhanced connectionist temporal classification for discrete speech emotion recognition," in *Proc. Interspeech*, Sep. 2019, pp. 206–210.

[24] Z. Aldeneh and E. Mower Provost , "Using regional saliency for speech emotion recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2017, pp. 2741–2745.

[25] G. Trigeorgis *et al.*, "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2016, pp. 5200–5204.

[26] W. Q. Zheng, J. S. Yu, and Y. X. Zou, "An experimental study of speech emotion recognition based on deep convolutional neural networks," in *Proc. Int. Conf. Affect. Comput. Intell. Interaction*, Sep. 2015, pp. 827–831.

[27] H. Wang, A. Li, and Q. Fang, "F0 contour of prosodic word in happy speech of Mandarin," in *Proc. Affect. Comput. Intell. Interaction, ser. Lecture Notes in Comput. Sci.*, Oct. 2005, pp. 433–440.

[28] Y. Kim and E. Mower Provost, "Emotion spotting: Discovering regions of evidence in audio-visual emotion expressions," in *Proc. ACM Int. Conf. Multimodal Interaction*, Oct. 2016, pp. 92–99.

[29] S. Zhang, A. Chen, W. Guo, Y. Cui, X. Zhao, and L. Liu, "Learning deep binaural representations with deep convolutional neural networks for spontaneous speech emotion recognition," *IEEE Access*, vol. 8, pp. 23 496–23 505, Feb. 2020.

[30] K. Han, D. Yu, and I. Tashev, "Speech emotion recognition using deep neural network and extreme learning machine," in *Proc. Interspeech*, Sep. 2014, pp. 223–227.

[31] S. Mao, P. Ching, and T. Lee, "Deep learning of segment-level feature representation with multiple instance learning for utterance-level speech emotion recognition," in *Proc. Interspeech*, Sep. 2019, pp. 1686–1690.

[32] E. Tzinis and A. Potamianos, "Segment-based speech emotion recognition using recurrent neural networks," in *Proc. Int. Conf. Affect. Comput. Intell. Interaction*, Oct. 2017, pp. 190–195.

[33] L. Tarantino, P. Garner, and A. Lazaridis, "Self-attention for speech emotion recognition," in *Proc. Interspeech*, Sep. 2019, pp. 2578–2582.

[34] S. Sahoo, P. Kumar, B. Raman, and P. PratimRoy , "A segment level approach to speech emotion recognition using transfer learning," *Proc. Asian Conf. Pattern Recognit., Ser. Lecture Notes Comput. Sci.*, vol. 12047, pp. 435–448, Nov. 2019.

[35] M. Chen, X. He, J. Yang, and H. Zhang, "3-D convolutional recurrent neural networks with attention model for speech emotion recognition," *IEEE Signal Process. Lett.*, vol. 25, no. 10, pp. 1440–1444, Oct. 2018.

[36] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2017, pp. 2227–2231.

[37] C.-W. Huang and S. Narayanan, "Attention assisted discovery of sub-utterance structure in speech emotion recognition," in *Proc. Interspeech*, Sep. 2016, pp. 1387–1391.

[38] P.-W. Hsiao and C.-P. Chen, "Effective attention mechanism in dynamic models for speech emotion recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2018, pp. 2526–2530.

[39] Y. Li, T. Zhao, and T. Kawahara, "Improved end-to-end speech emotion recognition using self attention mechanism and multitask learning," in *Proc. Interspeech*, Sep. 2019, pp. 2803–2807.

[40] R. Li, Z. Wu, J. Jia, S. Zhao, and H. Meng, "Dilated residual network with multi-head self-attention for speech emotion recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2019, pp. 6675–6679.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[42] T. Luong, H. Pham, and C. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Nat. Lang. Process.*, Sep. 2015, pp. 1412–1421.

[43] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 5998–6008.

[44] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *Proc. Int. Conf. Learn. Representations*, Apr.–May 2020, pp. 1–12.

[45] S. Mariooryad, R. Lotfian, and C. Busso, "Building a naturalistic emotional speech corpus by retrieving expressive behaviors from existing speech corpora," in *Proc. Interspeech*, Sep. 2014, pp. 238–242.

[46] A. Burmania, S. Parthasarathy, and C. Busso, "Increasing the reliability of crowdsourcing evaluations using online quality assessment," *IEEE Trans. Affect. Comput.*, vol. 7, no. 4, pp. 374–388, Oct.–Dec. 2016.

[47] C. Busso, S. Parthasarathy, A. Burmania, M. AbdelWahab, N. Sadoughi, and E. MowerProvost , "MSP-IMPROV: An acted corpus of dyadic interactions to study emotion perception," *IEEE Trans. Affect. Comput.*, vol. 8, no. 1, pp. 67–80, Jan.–Mar. 2017.

[48] F. Eyben, M. Wöllmer, and B. Schuller, "OpenSMILE: The Munich versatile and fast open-source audio feature extractor," in *Proc. ACM Int. Conf. Multimedia*, Oct. 2010, pp. 1459–1462.

[49] J. Arias, C. Busso, and N. Yoma, "Shape-based modeling of the fundamental frequency contour for emotion detection in speech," *Comput. Speech Lang.*, vol. 28, no. 1, pp. 278–294, Jan. 2014.

[50] T. Trinh, A. Dai, T. Luong, and Q. Le, "Learning longer-term dependencies in RNNs with auxiliary losses," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2018, pp. 4965–4974.

[51] L. Van Boven , K. White, and M. Huber, "Immediacy bias in emotion perception: Current emotions seem more intense than previous emotions," *J. Exp. Psychol.: General*, vol. 138, no. 3, pp. 368–382, Aug. 2009.

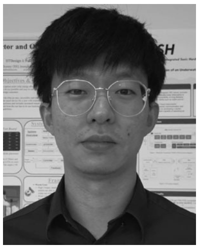
[52] K. Sridhar, S. Parthasarathy, and C. Busso, "Role of regularization in the prediction of valence from speech," in *Proc. Interspeech*, Sep. 2018, pp. 941–945.

[53] C. Busso and T. Rahman, "Unveiling the acoustic properties that describe the valence dimension," in *Proc. Interspeech*, Sep. 2012, pp. 1179–1182.

[54] D. Carney, C. RandallColvin, and J. A. Hall, "A thin slice perspective on the accuracy of first impressions," *J. Res. Pers.*, vol. 41, no. 5, pp. 1054–1072, Oct. 2007.

[55] C.-C. Lee, C. Busso, S. Lee, and S. Narayanan, "Modeling mutual influence of interlocutor emotion states in dyadic spoken interactions," in *Proc. Interspeech*, Sep. 2009, pp. 1983–1986.

[56] J. Arias, C. Busso, and N. Yoma, "Energy and F0 contour modeling with functional data analysis for emotional speech detection," in *Proc. Interspeech*, Aug. 2013, pp. 2871–2875.



Wei-Cheng Lin (Student Member, IEEE) received the BS degree in communication engineering from National Taiwan Ocean University, Taiwan in 2014 and the MS degree in electrical engineering from National Tsing Hua University, Taiwan in 2016. He is currently working toward the PhD degree with the Electrical and Computer Engineering Department, The University of Texas at Dallas. His research interests include human-centered behavioral signal processing, deep learning, and multimodal or speech signal processing. He is also a student member of the IEEE Signal Processing Society and International Speech Communication Association.

Carlos Busso Senior Member, IEEE) received the BS and MS degrees with high honors in electrical engineering from the University of Chile, Santiago, Chile, in 2000 and 2003, respectively, and the PhD degree in electrical engineering from the University of Southern California (USC), Los Angeles, in 2008. He is currently a professor with the Electrical Engineering Department, The University of Texas at Dallas (UTD). He was selected by the School of Engineering of Chile as the best electrical engineer graduated in 2003 across Chilean universities. His work has direct implication in many practical domains, including national security, health care, entertainment, transportation systems, and education. His research interest is in human-centered multimodal machine intelligence and applications. His current research includes the broad areas of affective computing, multimodal human-machine interfaces, nonverbal behaviors for conversational agents, in-vehicle active safety system, and machine learning methods for multimodal processing. At USC, he received a provost doctoral fellowship from 2003 to 2005 and a fellowship in Digital Scholarship from 2007 to 2008. At UTD, he leads the Multimodal Signal Processing (MSP) laboratory. He was the recipient of an NSF CAREER Award, the ICMI Ten-Year Technical Impact Award in 2014, the Hewlett Packard Best Paper Award at the IEEE ICME 2011 (with J. Jain), and the Best Paper Award at the AAAC ACII 2017 (with Yannakakis and Cowie). In 2015, his student was the recipient of the third prize IEEE ITSS Best Dissertation Award (N. Li). He was the general chair of ACII 2017 and ICMI 2021. He is a member of ISCA, and AAAC, and a senior member of the IEEE and ACM.