Audio-Based Piano Performance Evaluation for Beginners With Convolutional Neural Network and Attention Mechanism

Weiqing Wang, Jin Pan, Hua Yi, Zhanmei Song, and Ming Li, *Senior Member, IEEE*

***Abstract*—In this paper, we propose two different audio-based piano performance evaluation systems for beginners. The first is a sequential and modularized system, including three steps: Convolutional Neural Network (CNN)-based acoustic feature extraction, matching via dynamic time warping (DTW), and performance score regression. The second system is an end-to-end system with CNNs and the attention mechanism. It takes two acoustic feature sequences as input and directly predicts a performance score. We evaluate two proposed methods with our new open-access Yingcai Piano Performance Evaluation Phase III Dataset (YCU-PPE-III) that contains more than 2000 piano audio pieces recorded in multiple real test sessions. Experimental results show that the modularized system achieves a mean absolute error (MAE) of 3.79 in a 0-100-point range. Another end-to-end system also achieves an MAE of 4.40, which shows that it is possible to train a robust end-to-end piano performance evaluation system with only two thousand audio pieces.**

***Index Terms*—Attention, computer assisted piano learning, convolutional neural network, dynamic time warping, piano performance evaluation.**

I. Introduction

LEARNING to play piano is essential for college students in the preschool education department. They need to gain a certain level of piano playing skills from courses before graduating and becoming teachers in kindergartens. However, it is challenging for these students who are also piano beginners to perform well in such studies. Beginners need frequent practices, immediate and personalized feedback when learning a melody in the classroom. Moreover, it is both time-consuming and labor-intensive for the instructors to manually grade each student's performance in the mid-term and final exams which are all live playing examinations. It is also nearly impossible to manually check each student's performance at the beginning of each class for a big course roster. Hence, we propose a system to automatically assess piano performances for beginners.

There are several works related to music performance assessment and music tutoring. Russell [1] hypothesizes that there are some commonalities between the musical performance assessments of different instruments. They indicate that technique, musical expression, and overall perception are the most important factors in musical performance assessments. Since all of these factors are relatively subjective, Vidwans *et al.* [2] provide several objective measurements that can be used in automatic music tutoring systems. They extract both musical score dependent and independent features, then assess the performance using regression with these features. Huanhuan *et al.* [3] introduce a violin tutoring framework, which compares the transcription results with the reference and provide the evaluation of the performance. Deep neural network (DNN) is also a powerful tool for music performance assessment. Pati *et al.* [4] propose two DNNs for this task. One is a 1D convolutional neural network (CNN) with pitch contours as the input features, while the other is a 2D CNN with the input of Mel spectrograms. Finally, these two CNNs are combined and jointly trained to obtain better performance. For DNN based methods, a well-collected large scale dataset is important for network training. Bozkurt *et al.* [5] present a dataset for singing performance assessment. Along with the dataset, they also propose a logistic regression model that directly compares two fundamental frequency sequences to assess the singing performance.

For the task of piano performance evaluation, the common approach is also to compare the input sequence with a standard template and then measure their similarity with some alignment methods. Therefore, a good representation of the template and input is very important.

Some of the previous research focused on the comparison between Musical Instrument Digital Interface (MIDI) sequences. MIDI is a digital interface that carries event messages like

Manuscript received June 5, 2020; revised September 21, 2020, December 13, 2020, and February 14, 2021; accepted February 17, 2021. Date of publication February 23, 2021; date of current version March 17, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61773413, in part by National Social Science Fund of China under Grant BHA160085, in part by the Key Research and Development Program of Jiangsu Province under Grant BE2019054, in part by the Six talent peaks project in Jiangsu Province under Grant JY-074, in part by Guangzhou Key Area Research and Development Program under Grant 202007030011, and in part by Guangzhou Municipal People's Livelihood Science and Technology Plan under Grant 201903010040. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Federico Fontana. *(Corresponding author: Ming Li.)*

Weiqing Wang was with the School of Data and Computer Science at Sun Yat-sen University, Guangzhou 510006, Guangdong, China (e-mail: weiq.wong@icloud.com).

Jin Pan was with the School of SYSU-CMU Joint Institute of Engineering at Sun Yat-sen University, Guangzhou 510006, Guangdong, China (e-mail: jinpoon@hotmail.com).

Hua Yi and Zhanmei Song were with the School of Preschool Education, Shandong Yingcai University, Jinan 250104, China (e-mail: 175206548@qq.com; songzhanmei@126.com).

Ming Li is with the Data Science Research Center at Duke Kunshan University, Kunshan 215306, China and the School of Computer Science at Wuhan University, Wuhan 430072, China (e-mail: ming.li369@dukekunshan.edu.cn).

Digital Object Identifier 10.1109/TASLP.2021.3061267

notation, pitch, and velocity. Morita *et al.* [6] introduce a spline curve to a piano performance evaluation system and achieved 0.65 average correlation coefficients between system-estimated scores and expert-evaluated scores. Akinaga *et al.* [7] employed a set of three regression curves for onset time, MIDI velocity, and duration. They also apply Karhunen-Loeve expansion (KL expansion) and a k-Nearest Neighbor (KNN) algorithm using the MIDI sequence as input to predict the performance score.

In general, the MIDI stream automatically generated from the musical score is a good template for comparison since the musical score contains a more precise onset and duration than the audio signal does. However, the audio piece performed by an instructor may have additional information, such as timbre, dynamics, emotion, or instructor's personal style. Also, many pianos can generate only audio signals rather than MIDI sequences, and MIDI data requires special capture equipment.

Since MIDI data requires some specific hardware, the audio signal is a better choice as the template for the purpose of being widely used, but an additional step is needed to transform the audio signal into a MIDI-like sequence, called Automatic Music Transcription (AMT) [8]–[10]. It can be considered that the AMT system replaces the MIDI capture equipment. Many related works focus on how to transcribe an audio signal to the musical score like piano roll notations [11], which is a simplified representation of the piano MIDI sequence.

There are several statistical methods for AMT tasks. Raphael presented a hidden Markov model approach and a likelihood model for piano music transcription [12]. Marolt presented a connectionist approach to automatic transcription of polyphonic piano music and proposed a partial tracking method based on a combination of an auditory model and adaptive oscillator networks [13]. Other than piano-music transcription, a probabilistic model for multiple-instrument automatic music transcription is proposed in [14]. Non-negative matrix factorization (NMF) has also been applied to AMT [15]. Others have proposed an unsupervised AMT system for non-negative, sparse, linear decomposition of power spectra [16], [17]. However, this unsupervised decomposition usually reduces the transcription's correspondence among the pitches, resulting in difficulties when interpreting the transcription results. The incorporation of harmonic constraints in the training algorithm addresses these problems [18], [19].

Deep learning methods have also been investigated for AMT tasks. Bock *et al.* [20] presented a piano transcription system using a recurrent neural network (RNN). Boulanger *et al.* [21] applied a combination of an RNN and a restricted Boltzmann machine (RBM) to generate a piano-roll. Convolutional neural networks (CNNs) have also been used for AMT tasks [22]–[24]. Kelz *et al.* [22] proposed a convolutional neural network acoustic model with an F1-score of 70.60% on the MAPS dataset [25]. Sigtia *et al.* [23] and Hawthorne *et al.* [24] proposed an end-to-end hybrid neural network with a combined CNN and RNN framework. Moreover, by integrating a separate onset detection module with the acoustic model, the performance can be further enhanced [24].

After extracting the piano transcription using AMT, we need a robust audio alignment approach to measure the similarity between the template and the input and estimate the performance score. Ewert *et al.* [26] introduced several different audio features and three types of cost matrix to improve the accuracy of alignment and synchronization. Li *et al.* [27] proposed an onset based method to align the audio and musical score. Since the sustained note can cause the audio-score mismatch, they reduce the sustained spectral components after detecting an onset, improving the alignment accuracy. Both of these alignment methods only compare one recording with another. In [28], a technique that aligns multiple audio segments is proposed, called joint alignment.

Recently, many sequence-to-sequence models with attention mechanisms are proposed for machine translation [29], speech recognition [30], and image captioning [31]. Since it can address the problems of long sequences, attention mechanism has become an essential part of most sequence-to-sequence models and transduction models [32]. Therefore, attention mechanism is a good alignment method for many end-to-end models.

In this paper, we first propose a modularized piano performance evaluation system, which can be considered as a sequential pipeline. This system contains three parts: high-level acoustic feature extraction, alignment, and regression. Since audio may suffer from various kinds of noise brought by the recording process, we adopt the piano key posterior probabilities (PKPP) [33] as the acoustic features generated by the CNN-based acoustic model. PKPP is a frame-level acoustic feature that can represent the probability of each piano key being pressed in a frame. PKPP is also a relatively high-level feature that describes a better approximation of what the performer plays. After extracting the PKPP feature sequences, we align these sequence pairs by the dynamic time warping (DTW) [34] algorithm to measure the similarity between two audio pieces. Then, we extract the global matching features from the DTW results, which can represent the quality of a student's performance. Finally, a regression model is adopted to predict an overall performance score using these global matching features. Figure 1 shows the architecture of our proposed modularized piano performance evaluation systems.

We also proposed an end-to-end neural network, which employs the attention mechanism to align the feature sequences. The network contains two convolutional layers, followed by a gated recurrent unit (GRU) [35] layer. Then we automatically extract the attention map and obtain a weighted feature sequence. Finally, two fully-connected layers take the mean of the feature sequences as input to predict a performance score. Figure 5 shows the architecture of our proposed end-to-end deep neural network-based approach. In addition to the network structure, we also use a data augmentation method with pitch shifting and time stretching similar to [36]. A correlation coefficient related loss function and the transfer learning strategy are employed to further enhance the performance.

This paper is an extended work of our conference paper [33]. The major new novelty of this journal paper is the end-to-end neural network-based approach. Furthermore, the experimental data is extended from 200 to over 2000 pieces. For the modularized system, we refine both the CNN network structure and global feature extractor to further enhance the performance. We

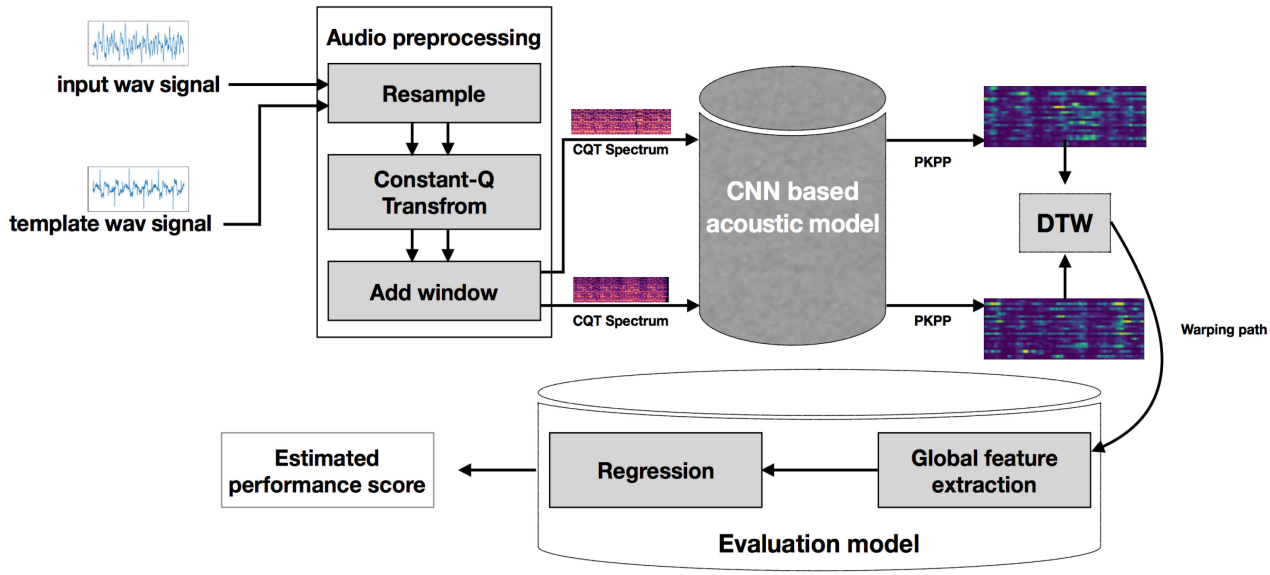


Fig. 1. The architecture of the proposed modularized system.

TABLE I
THE DETAILS OF ALL SONGS. THE SCORES OF THESE SONGS CAN BE ACCESSED AT https://github.com/spwb/PianoScore.git

Song ID	Song Title	0-60	60-70	70-80	80-90	90-100	Total
1	The people in their hometown	2	17	36	112	38	205
2	Ping-Pong variations (part 4 and 5)	0	8	120	87	0	215
3	Ping Pong variations (part 1, 2 and 3)	0	15	119	80	0	215
4	Serenade	3	14	43	126	14	200
5	Long drum dance of Yao nationality	1	24	91	89	0	205
6	In May	5	11	31	130	28	205
7	Song of spring	11	12	40	73	54	190
8	Herdsmen sing for the chairman	0	25	87	13	0	125
9	Nasty woodpecker	12	24	52	83	24	195
10	Turkish march	0	8	95	27	0	130
11	William Tell Overture	6	14	55	60	10	145
12	Champagne	9	9	35	105	42	200

also add more experiments for the modularized system under different evaluation setups.

The rest of our paper is organized as follows. Section 2 introduces the YCU-PPE-III dataset. Section 3 describes the modularized system. Section 4 introduces the end-to-end framework. Experimental results are provided in Section 5. Section 6 presents the conclusions.

II. DATASET DESCRIPTION

A. *Dataset for Piano Performance Evaluation*

To measure the performance of piano audio from actual environments, we collected over 2000 audio files recorded in different examination sessions at Shandong Yingcai University in 2017 and 2018. From this collection, we constructed a dataset called the Yingcai Piano Performance Evaluation phase III dataset (YCU-PPE-III).[1] We divided these recordings into 12 different categories according to the song title, and each category contains hundreds of recordings. Because these song titles are not in English, we have labeled each song with a number from 1 to 12 in this work, as shown in Table I.

The piano audio was recorded by connecting the line-out or earphone interface on an electronic piano to the mic-in or line-in interface on a smartphone. We use voice activity detection (VAD) [37] to detect and cut non-audio segments at the beginning and the end of each recording. In addition to this, we did not remove other silent audio segments since these silent segments provide information about rhythm and speed. For each song, there are around 200 audio recordings played by students and an audio template played by instructors. We manually convert the musical scores to MIDI. All of the musical scores in Table I can be found at Github (https://github.com/spwb/PianoScore.git). These musical scores are relatively simple since all of these musical scores come from a textbook for beginners. The instructors played an audio template for each song.

Additionally, three different instructors independently graded each audio piece, making sure that the grade (range from 0 to 100) labels are relatively objective. We take the average of the performance scores from these three annotators as the

[1]Please contact the corresponding author for accessing this dataset, available for non-commercial academic research purposes.

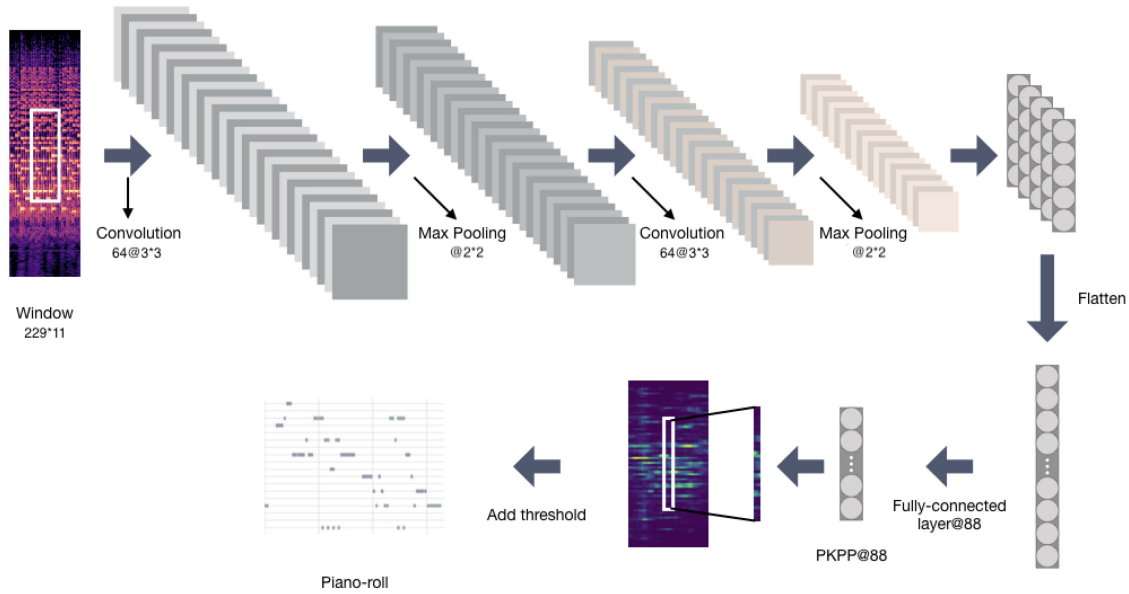


Fig. 2. The architecture of our acoustic model. This figure shows one of our three acoustic models, and the dimension of input and output is 229 and 88. The setups of the other two models are 229-d input to 12-d output and 88-d input to 12-d output, respectively. These three acoustic models have similar architectures.

label of each recording. Since these audio files were recorded in examination sessions and most of the performance scores are passing (greater than 60), which means that most of the students pass the examination according to the teacher's criteria, as Table I shows.

B. Dataset for Acoustic Model Training

We trained our proposed acoustic model using the MAPS dataset [25], which consists of 270 pieces of piano sound and corresponding MIDI annotations. In our experiment, we train the acoustic model on the 210 synthesized recordings and test it with 60 real recordings. In the training step, 180 synthesized recordings are used for training, and the remaining recordings are used for validation, which is the same as configuration 2 in [23].

III. The Modularized System

In this section, we discuss the modularized system with three submodules. This system can be considered as a sequential pipeline, including acoustic model, alignment, and regression, as shown in Figure 1.

A. Acoustic Model Design

1) Audio Preprocessing: We train the CNN-based acoustic model using the MAPS dataset, as mentioned in Section 2. We choose constant-Q transform as the input of our acoustic model.

A constant-Q transform (CQT) is a kind of time-frequency representation proven to have better performance for musical signal analysis than Fourier Transform by holding the ratio of successive pitches at the constant value $2^{\frac{1}{12}}$ [38]. Furthermore, the CQT frequency axis is a linear note distribution that corresponds to most instruments, including the piano. Although similar to the short-time Fourier transform, CQT requires fewer frequency bins at high frequencies because of its exponential frequency resolution. Given a minimum frequency f_0 and the number of bins per octave b, the constant-Q transform $C[k]$ is defined according to the following equations.

$$K = \left\lceil b * \log_2 \left(\frac{f_{max}}{f_0} \right) \right\rceil \tag{1}$$

$$Q = (2^{\frac{1}{b}} - 1)^{-1} \tag{2}$$

where b determines the number of bins per octave, then:

$$N_k = \left\lceil Q \frac{f_s}{f_k} \right\rceil \tag{3}$$

where $k < K$, $f_k = (2^{1/b})^k f_0$ and $f_s = \frac{1}{T}$, then $C[k]$ can be calculated as:

$$C[k] = \frac{1}{N_k} \sum_{n<N_k} x[n] w_{N_k}[n] e^{-2\pi Q/N_k} \tag{4}$$

Preprocessing is done using librosa [39]. After downsampling the audio to 16 kHz, we apply CQT to each clip with 128 ms frame size and 32 ms frameshift and produce a 229-d and 88-d CQT spectrum. For 229-d spectrums, the bins per octave is 36, and for 88-d spectrums, the bins per octave is 12. Figure 2 shows one of our three acoustic models, which takes 229-d CQT spectrum as input. These three acoustic models have the same basic architecture and are only different in the dimension of inputs and outputs. We also apply the z-score normalization to each dimension of the CQT spectrum at a per audio piece level. For CNN training, we set the window size to 11, with the outputs corresponding to the targets of the central frame.

Since there are 88 keys on a piano, we convert the MIDI annotation to an 88-d binary piano key vector as the target of the acoustic model. We also generate a 12-d binary piano key vector

TABLE II
ACOUSTIC MODEL PERFORMANCE ON THE MAPS DATASET COMPARED WITH OTHER CNN MODELS

Network setup	$Precision$	$Recall$	F_1
Pan's [33]	-	-	62.03
Hybrid ConvNet [23]	-	-	64.14
ConvNet [22]	74.50	67.10	70.60
88-d to 12-d	87.53	72.13	79.08
229-d to 12-d	91.41	81.32	86.07
229-d to 88-d	83.62	66.88	74.32

according to the pitch chroma. The pitch chroma means that each octave contains 12 semitones, and the ratio of frequencies between successive semitones is constant. The binary piano key vector contains only 0 and 1, representing the status of piano keys, whereas the output of the CNN-based acoustic model is a posterior probability between 0 and 1. Note that the binary vector contains more than one nonzero value as multiple keys can be pressed at the same time.

2) Architecture: Convolutional neural networks have been widely employed on signal processing tasks, and researchers have demonstrated the effectiveness of CNNs with AMT tasks [23]. The spectral context contains patterns that describe chord, rhythm, and harmonic features. Our acoustic model is based on CNN.

The acoustic model consists of two convolutional layers and one fully-connected layer as depicted in Figure 2. The model takes a context window of 11 frames as input and produces a PKPP vector of the central frame as output. There are 64 kernels in each convolutional layer, with a kernel size of 3×3. The two convolutional layers are both activated by a hyperbolic tangent function. They are followed by a max-pooling layer of size 2×2. Then we use a fully connected layer to predict the PKPP, and the output size is 12 or 88. A sigmoid function is set on the last layer to scale the output in the range of 0 to 1. We set dropout with a rate of 0.5 to each convolutional layer to avoid overfitting. We have mentioned that there are three acoustic models which are only different at the dimensions of inputs and outputs. The setups of these acoustic models are 229-d input to 88-d output, 229-d input to 12-d output, 88-d input to 12-d output, respectively, as shown in Table II.

3) Training and Results: We trained the network with 1.5 million frames and tested with 50 000 frames. The output is a posterior probability vector with values between 0 and 1, which is directly used for the subsequent modeling. In order to evaluate the accuracy of this acoustic model, we separately employ a threshold of 0.5 to obtain a standard output containing only 0 and 1 as values just for calculating the F1-score.

Since the target is a sparse matrix and contains numerous zeros, we measure the performance of the acoustic model with F1-score. The F1-score is calculated as follows:

$$\text{precision} = \frac{1}{T}\sum_{t=1}^{T}\frac{TP[t]}{TP[t]+FP[t]}, \tag{5}$$

$$\text{recall} = \frac{1}{T}\sum_{t=1}^{T}\frac{TP[t]}{TP[t]+FN[t]}, \tag{6}$$

$$\text{f1 score} = \frac{2\times P\times R}{P+R}, \tag{7}$$

where T is the number of frames, P is the precision, R denotes the recall, and TP, FP, and FN denote true positive, false positive, and false negative, respectively.

Table II shows the results of three different acoustic model configurations. For example, "88-d to 12-d" means that we use the 88-d CQT spectrum as the input and set the network output size to 12 according to pitch chroma. In addition to the potential of extracting robust features, the acoustic model also diminishes the dimension of the acoustic feature. This reduction in output data accelerates the DTW algorithm considerably, making our system more efficient.

Our network achieves 74% F1-score with 229-d input and 88-d output, as shown in Table II. Compared with the ConvNet in [22] and the hybrid ConvNet in [23], our acoustic model achieves comparable performance.

B. DTW Based Alignment

Since we assume that the piano performance is based on the similarity between the input and template audio, we must first measure that similarity. There are various methods to solve this problem. One traditional approach uses hidden Markov models and applies the Viterbi algorithm to calculate the most likely path [40], [41].

In this paper, we employ Dynamic Time Warping (DTW) [34] as our solution for alignment. DTW can measure the similarity between two sequences that may vary in duration and speed, but it is also difficult to be used on long sequences due to its space complexity and quadratic time. To reduce both time and space complexity, Salvador *et al.* [42] proposed a DTW algorithm in linear time and space, called FastDTW.

Given two sequences S_1 and S_2, DTW defines a cost matrix $D \in R^{M\times N}$, where M and N are the dimensions of S_1 and S_2, respectively. We use cosine similarity as the cost measure to compute the cost matrix D. The accumulated cost matrix $A \in R^{M\times N}$ can be derived from matrix D with the optimal transition for node $n_{i,j}$ in A given by

$$A_{i,j} = \min\{A_{x,y} + d_{i,j}\},$$
$$(x,y) \in \{(i,j-1),(i-1,j),(i-1,j-1)\}, \tag{8}$$

where $d_{i,j}$ is the cost of $n_{i,j}$ given by the cosine similarity between frame i in S_1 and frame j in S_2.

After calculating the matrix A, we obtain a global alignment cost at the final entry in the accumulated cost matrix. We also extract an optimal warping by applying a backtracking algorithm to matrix A, as shown in Figure 3.

C. Global Feature Extraction

The global cost of the accumulated cost matrix reveals the overall similarity between the two sequences, but we are interested in more details. The optimal path further describes the similarity. As Figure 3 shows, the warping path for the better performance (i.e., receiving a higher grade according to the annotators) is smoother and closer to the ideal diagonal path

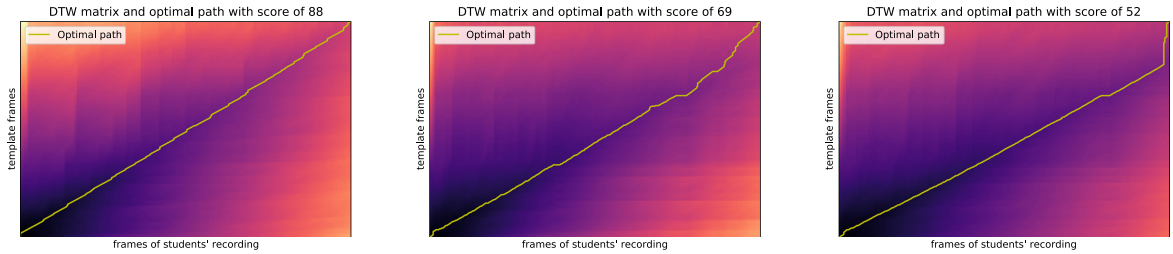


Fig. 3. Examples of DTW optimal paths. The DTW optimal path with 88 points (the grades annotated by instructor) is more towards a diagonal line.

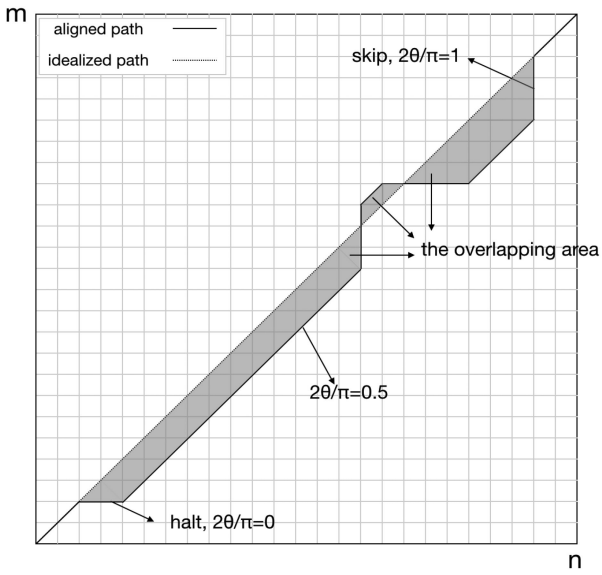


Fig. 4. Some features extracted from path. The length of template is n and the length of input is m.

than the lower-scoring one. Therefore, the optimal warping path presents us with additional features, as shown in Figure 4. In our system, motivated by [43], we extract several features:

- frame ratio. The 'frame ratio' reflects the overall difference between templates and inputs in terms of tempo and speed. For a template sequence with n frames and an input sequence with m frames, the $\text{frame ratio} = |\frac{m}{n} - 1|$.
- average cost. The 'average cost' reflects the average difference between the template and input audio. The average difference describes the average distance between each frame of input audio and the corresponding frame of template audio. It is calculated as $C_{avg} = \frac{C}{l}$, where C is the overall cost and l is the length of the warping path. Note that we do not adopt the overall cost because the average cost has more robustness against duration differences.
- ratio of skips in the path. The 'number of skips' shows how many template frames are skipped in the input audio, which occurs, for example, when the performer forgets part of the piece. It is defined as the transition from $\text{node}_{i,j-1}$ to $\text{node}_{i,j}$. The ratio of skips is calculated as $\text{ratio of skips} = \frac{\text{number of skips}}{n}$.
- ratio of halts in the path. The term 'halt' is defined as the transition from $\text{node}_{i-1,j}$ to $\text{node}_{i,j}$, as it reflects a pause in the performance at that point compared to the template. The ratio of skips is calculated as $\text{ratio of halts} = \frac{\text{number of halts}}{m}$.
- the standard deviation of the angle sequence. We first calculate the $\phi = \frac{2\theta}{\pi}$, where θ is the angle in each cell of the grid. We can then extract both the mean and standard deviation of this sequence. Only the standard deviation contributes to our experiments.
- the standard deviation of the cost sequence along the warping path. The warping path of the DTW result shows how the frame-level cost change from one point to another. We calculate the standard deviation of the frame-wise cost sequence along the optimum alignment path.
- the ratio of the overlapping area between the idealized path and the warping path. The idealized path is the diagonal of the cost matrix. When the student plays slowly or misses some notes, the slope of the alignment path can be different from the slope of the optimal path. We can calculate the ratio of the overlapping area between these two paths and the total area of the cost matrix.

We believe these seven features can represent the overall performance of each template-input audio pair, but they also may contain some redundant information. For example, the 3 rd and 4th feature may account for the general difference in the length between the performance and the reference recording. We use the frame ratio to compensate for this factor since the frame ratio can capture the general length difference. These joint multi-dimensional feature vectors can become more informative to capture the cases when the performer forgets part of the piece or makes a pause in the performance.

D. Song-Dependent and Song-Independent Modeling

Since there are 12 songs in our dataset, both song-dependent and song-independent evaluations are necessary. For the song-dependent model, we employ the leave-one-out (LOO) cross-validation for the modularized system on each song. However, these evaluation systems have limitations because each song requires a separate regression model, although the acoustic feature extraction, DTW alignment and global feature extraction steps are the same. To obtain a more robust and simple system which can be tested with unseen songs, we utilize a strategy called leave-one-song-out (LOSO) cross-validation, or song-independent method. This method requires normalized features for all songs. We globally normalize the features using min-max normalization where the min and max values are calculated by

all training data. Then, for a specific song, we test only with data from that song while training the evaluation model with data from all other songs.

For song-dependent modeling, we employ linear least-squares minimization to predict the estimated performance score. For the song-independent modeling, we employ the support vector machine (SVM) as our regression method since we have more samples by pooling data from multiple songs together.

E. Template Selection

There are two kinds of templates in our dataset: audio template and MIDI template. The audio templates are performed by some instructors who also grade students' recordings. For each audio template, we extract the CQT spectrum and PKPP features. In order to consider MIDI as templates, we first transcribe all picture-format scores to MusicXML, a universal format for describing musical information. Then MIDI files are generated from MusicXML using the MuseScore2 software. For the MIDI files, we extract the MIDI event and produce binary piano key (BPK) vectors as the template. Here the BPK is in frame level, and the frame size is 64 ms, which is the same as the window size of the acoustic model. We use these BPK features as the template in the DTW matching as a contrastive system and compare it with PKPP features based ones.

IV. The End-to-End Framework

In this section, we propose an end-to-end deep neural network framework with CNN and the attention mechanism. Similar architecture has been used for machine translation [44] and speaker verification [45], and both of these networks work well on short sequences. However, the CQT sequences of the piano audio recordings usually contain thousands of frames, and the original architecture cannot handle such long sequences. Moreover, since our dataset contains only about 2000 audio pieces, we need a robust data augmentation method to expand the size of training samples and reduce overfitting. In this section, we introduce our proposed end-to-end model and its detailed implementations.

A. Data Augmentation and Preprocessing

Due to the size limitation of our dataset, we cannot directly train the model with only about 2000 audio pieces. Considering that the audio sequences may be too long for the attention mechanism to learn the internal information, we can accelerate the audio signals from both the template and the input by several times to reduce the length. However, if the speed-up audio is too short after acceleration, it will lose much temporal-level detailed information. We find that the performance does not change a great deal if the audio pieces are accelerated by less than five times. By changing the speed of the original audio piece, we can obtain more data for training. Another approach is that we can change the pitch of the audio pieces by raising or lowering the pitch/key. Note that the two audio pieces to be compared should be in the same condition, which means that they should have the same speed-up and pitch-shift levels. Therefore, we can believe that speed-up and pitch-shift do not change the similarity between the template and the input audio pieces.

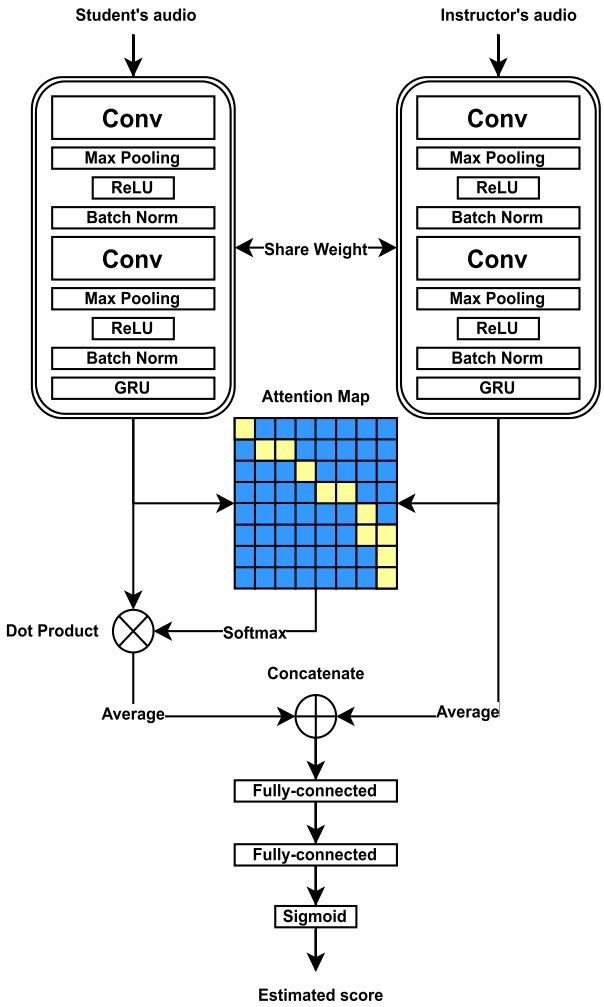


Fig. 5. Architecture of the proposed end-to-end system.

We use the 88-d CQT spectrum as the input of this neural network. Each audio feature sequence is first normalized to zero-mean along the time axis and zero-padded in the end to maintain a fixed length in a batch. Since the performance score is in the range of 0 to 100, we can directly divide the score by 100 as the normalized target.

B. Architecture

Figure 5 shows the architecture of our end-to-end system. The kernel size of each conv layer is 5×5, followed by a 2×5 max-pooling layer. The number of channels is 4 and 16 for each conv layer. The hidden size and the number of layers of GRU are 128 and 2, respectively. Then we can calculate the attention map and the mean of the weighted feature maps. The size of two fully-connected layers is 512 and 128, respectively.

Two convolutional layers and a gated recurrent unit (GRU) extract the frame-level hidden state $\mathbf{h}$, and then calculate the attention map. We multiply the attention map by the student's hidden state, producing a weighted feature. Then we average both students' weighted features and the instructor's hidden state

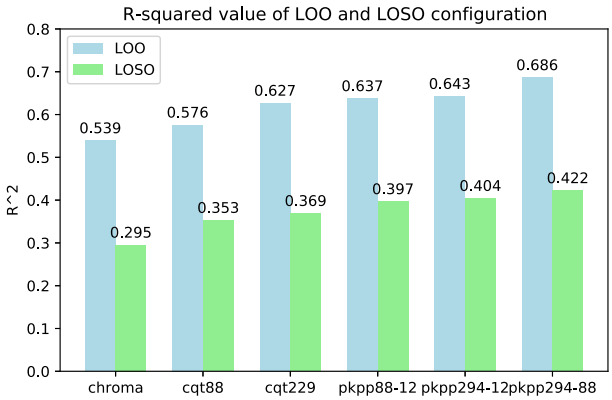


Fig. 6. The R-squared values of modularized systems with LOO and LOSO configurations.

along the time axis and concatenate these two features. Finally, a fully-connected layer with an ReLU and a fully-connected layer with a sigmoid function predict a score between 0 and 1.

Although we accelerate the audio and reduce the length, the input feature sequences are still too long for the attention calculation to process. After zero-padding, the features contains thousands of frames. Therefore, we employ two conv layers with a large pooling size to reduce the dimensions of the features. Second, the acoustic features are zero-padded since the matrix operation requires the same dimension in each batch. If the original features are long, there are only a few zeros at the end of the features. If the original features are short, many zeros will change the mean value of these features, and the neural network cannot predict an accurate score. To solve this problem, we use a mask matrix to ignore all zeros. Then we can calculate the sum of the features and divide them by their original length.

C. Attention Mechanism

We employ the dot product attention described in [29]. Assume that the student's acoustic feature sequence is $\{\mathbf{s}_1, \ldots, \mathbf{s}_m\}$ with m frames and instructor's acoustic feature sequence is $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ with n frames.

Then, GRU extracts $\mathbf{h}_i$, which is the student's hidden state at position i, and $\bar{\mathbf{h}}_t$, which is the instructor's hidden state at position t. All hidden states has the same dimension, which is the hidden size. The alignment scores can be calculated using softmax:

$$a_{t,i} = \frac{\exp(\text{score}(\mathbf{h}_i, \bar{\mathbf{h}}_t))}{\sum_{i'}^{m} \exp(\text{score}(\mathbf{h}_{i'}, \bar{\mathbf{h}}_t))} \tag{9}$$

where the 'score' function is dot product between $\mathbf{h}_i$ and $\bar{\mathbf{h}}_t$. Hence, $a_{t,i}$ is a scalar, and all $a_{t,i}$ forms an attention map $\mathbf{M}$ with size of $n \times m$, where $\mathbf{M}_{t,i} = a_{t,i}$, as shown in Figure 7.

Finally, the weighted context vector $\mathbf{c}_t$ at position t can be calculated as:

$$\mathbf{c}_t = \mathbf{M}_t \mathbf{H}^{\mathrm{T}} \tag{10}$$

where $\mathbf{M}_t = [a_{t,1}, \ldots, a_{t,m}]$, and $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_m]$. By averaging both $\mathbf{c}_t$ and $\bar{\mathbf{h}}_t$ over all time steps t, we can obtain two fixed-length context vectors which contain the information of the audio pieces. Then we can concatenate these two vectors and get a joint vector as the input of the fully-connected layer.

D. Training, Fine-Tuning, and Testing

As Table I shows, our dataset contains 12 songs. For each particular test song, we first pre-train the model on 11 songs (song independent modeling). Once the model converges, we will fine-tune and test the model on the remaining test song. Therefore, there will be 12 models if we pick different songs as the fine-tuning dataset (song dependent modeling).

1) Pre-Training and Testing: Since we have to train 12 different models, each time we select 11 songs as the pre-training dataset. For the 11 songs, we split the original data into a training set and a validation set, and then perform data augmentation on both the training set and the validation set. But the dataset is still unbalanced after augmentation. During the training process, we select balanced data in each batch to avoid overfitting.

2) Fine-Tuning and Testing: After the pre-trained model converges on the 11 songs, we can fine-tune and test this model on data from the remaining test song using 2-fold cross-validation. We equally split the original data of this test song into a training set and a testing set, and perform data augmentation for each audio recording. Therefore we can make sure that the augmented audio pieces derived from the same original pieces are in only one of the two sets. Then we fine-tune the pre-trained model using this test song's training set with a small learning rate.

E. Loss Function

We first employ the mean square error (MSE) loss during the training process, but the output always converges to the mean performance score of the entire dataset even though this dataset is balanced. To solve this problem, we use a Concordance Correlation Coefficient loss [46], [47] L_{coef} to constrain the output. The Concordance Correlation Coefficient can be calculated as:

$$\text{coef} = \frac{2\rho\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2} \tag{11}$$

where μ_x and μ_y are the means for the output and target, σ_x^2 and σ_y^2 are the corresponding variances, and ρ is the correlation coefficient between output and target. To maximize the coefficient, we can minimize the loss L_{coef}:

$$L_{coef} = 1 - \text{coef} \tag{12}$$

The total loss L should be:

$$L = L_{mse} + \lambda L_{coef} \tag{13}$$

where the L_{mse} is the MSE loss, and the λ is a constant.

V. Experimental Results

A. System Setup

1) The Modularized System: For the modularized system, the PKPP feature sequences are extracted at first. The acoustic model performed 88-d CQT to 12-d PKPP, 229-d CQT to 12-d PKPP, and 229-d CQT to 88-d PKPP, respectively. We compare

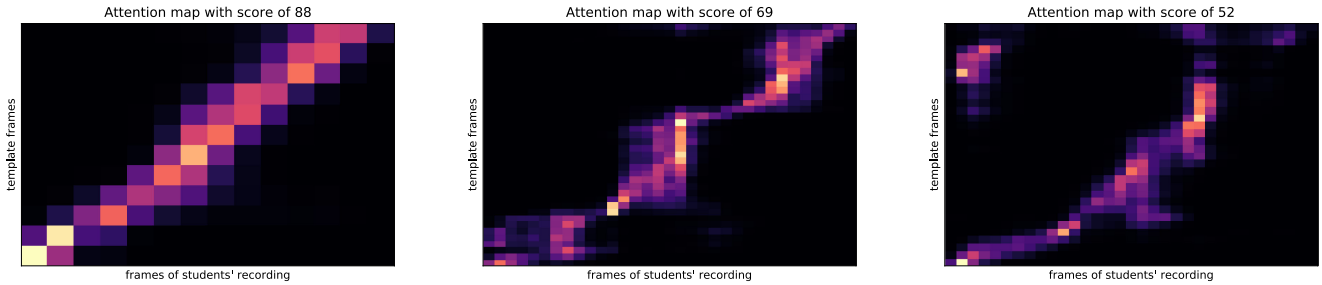


Fig. 7. Examples of attention maps. The attention map with a higher score also shows a better aligned path. The data for attention map is same as the data for DTW example in Figure 3.

the modularized methods with several baseline systems, which have the same basic architectures as the modularized system without the CNN-based acoustic model. The baseline system takes 88-d and 229-d CQT spectrum as input. After the alignment and the global feature extraction, we applied both leave-one-out (LOO) cross-validation and leave-one-song-out (LOSO) cross-validation in our experiments.

For LOO setup, we apply the linear regression, which can be easily solved by pseudo inverse $\hat{w} = (X^T X)^{-1} X^T y$, where $\hat{w}$ is the weight, $X = [x_1, x_2, \ldots, x_7, 1]$ is the feature matrix, and y is the performance score vector. For LOSO setup, we employ SVM as the regressor with rbf kernel using sklearn scikit-learn. The regularization parameter is 10, and the kernel coefficient (gamma) is set to "auto".

2) The End-to-End System: In the end-to-end framework, for each test song, we first pick the other 11 songs for pre-training and split these audio pieces into the training set and validation set. Then we can perform data augmentation on both the training set and the validation set. All of the audio pieces are accelerated by 2, 2.5, 3, 3.5, 4, and 4.5 times. Then we shift the pitch of the audio by raising or lowering 1$\sim$8 semitones using the function pitch_shift in librosa [39]. After the data augmentation, we can obtain over 200000 audio pieces, which is 96 times more than the original data. We extract the 88-d CQT spectrum and perform z-score normalization along the time axis. Each feature is zero-padded to a fixed length, and two corresponding features should have the same speed and same pitch. We train the model for 200 epochs with a batch size of 96. The learning rate of stochastic gradient descent (SGD) optimization is 0.01, with a momentum of 0.9. The learning rate is divided by 10 every 50 epochs. The λ of the loss function mentioned in Section IV is set to 0.01. The validation data is 10 percent of the original data, and it is randomly selected. In addition, we set a dropout at the end of the penultimate fully-connected layer to reduce overfitting.

After obtaining 12 pre-trained models, we can fine-tune each model with 2-fold cross-validation on the data of the remaining test song. The fine-tuning process is similar to the pre-training process. The only difference is that we only fine-tune the model for 50 epochs, and the learning rate is set to 0.0001.

During the testing process, we first predict the score for all data in the testing set of the test song. Since an original audio piece can be augmented to several pieces with different speeds and pitches, we will calculate the mean score of these audio pieces as the final score of the original audio.

B. Results and Discussions

1) The Modularized System: Table III shows the mean absolute error (MAE) and standard deviation (STD) of the baseline system compared to the modularized system. The input to the baseline system is 88-d CQT and 229-d CQT. Because of the different inputs and outputs of the acoustic model, our modularized system has three different setups. The results demonstrate that the proposed modularized systems are better than the baseline system. Table III also shows the results when given chroma as an acoustic feature as a label, which does not match the performance of the baseline and the modularized systems. Table VII shows the correlation coefficients (CCs) of all systems. Table III and Table VII show that the CNN based acoustic model can improve the performance of the baselines system via the proposed PKPP features.

In Table VII, we also include human performance (one vs. mean), human performance (one vs. another), and 229-88 PKPP vs. one human annotator, these three columns. The inter-rater agreement or disagreement is described as the correlation coefficient of human performance (one vs. another). From the results, we can observe that, the inter-rater agreement between any two annotators is relatively lower than the inter-rater agreement between one annotator and our 229-88 PKPP modularized system, which certainly is not a fair comparison since the system is being trained on the mean-ratings of these annotators. However, in this song dependent LOO setup, the output of the proposed system is able to regress to the mean of the individual human annotators which is shown in Table VII by the low MAE and the high correlation scores with the average. Figure 6 shows the R-squared values of each system with LOO and LOSO configuration, where we can find that the performance of the PKPP-based system is better than the CQT-based system.

Table V presents the MAE and STD of the song independent LOSO method. Each time we use the audio pieces from 11 songs as training data and test on the remaining song. The song ID indicates which song is selected as testing data. This setup does not require song dependent training data and also achieves promising performance.

To check if the improvement of the modularized system is statistically significant, we perform a t-test between the result of the baseline system and the modularized system by using the ttest_ind function in scipy [48]. The null hypothesis is that the MAE of the modularized system is not smaller than the

TABLE III
MAE AND STD OF THE MODULARIZED SYSTEM (AUDIO AS TEMPLATE, SONG DEPENDENT (LOO))

	chroma		88-d CQT		229-d CQT		88-12 PKPP		229-12 PKPP		229-88PKPP	
SongID	MAE	STD	MAE	STD	MAE	STD	MAE	STD	MAE	STD	MAE	STD
1	6.64	4.77	6.25	5.11	5.79	4.60	5.84	4.35	5.77	4.60	5.28	4.62
2	2.64	1.90	2.60	1.93	2.61	1.93	2.72	1.98	2.54	1.91	2.51	1.94
3	3.07	2.92	3.03	2.19	3.04	2.21	3.01	2.15	3.22	3.38	3.03	2.23
4	4.88	4.04	4.69	3.97	4.57	4.16	4.40	3.92	4.66	4.00	4.26	3.90
5	3.07	2.88	2.78	2.53	2.70	2.51	2.88	2.92	2.99	2.87	2.69	2.62
6	5.77	5.53	5.50	4.76	5.19	4.58	5.05	4.54	4.66	4.06	4.30	3.65
7	6.33	5.66	5.61	5.19	5.01	4.71	5.15	4.55	4.46	4.80	4.93	4.78
8	3.16	2.48	3.19	2.50	3.27	2.44	3.07	2.42	2.93	2.00	3.10	2.28
9	5.50	5.42	5.95	6.04	5.50	5.37	4.88	5.06	4.89	4.92	4.41	4.32
10	3.33	3.66	3.02	2.47	2.90	2.08	2.79	2.02	2.97	2.08	2.93	2.31
11	4.73	4.66	5.57	5.27	5.16	4.79	5.11	4.71	4.21	3.54	4.65	4.26
12	5.22	6.55	4.29	5.77	4.18	5.35	4.37	5.55	4.37	6.43	3.44	5.23
Average	4.57	4.68	4.37	3.97	4.15	3.72	4.10	3.68	3.97	3.71	**3.79**	**3.51**

TABLE IV
MAE AND STD OF THE MODULARIZED SYSTEM (BPK AS TEMPLATE AND PKPP AS INPUT, SONG DEPENDENT (LOO))

	88-12 PKPP		229-12 PKPP		229-88 PKPP	
SongID	MAE	STD	MAE	STD	MAE	STD
1	7.04	5.11	7.30	5.30	5.57	4.69
2	2.58	1.91	2.77	2.06	2.89	2.09
3	3.08	2.13	2.81	2.04	3.08	2.18
4	5.66	4.55	5.40	4.40	4.76	4.11
5	3.71	3.04	3.57	3.46	3.54	3.16
6	5.46	5.36	5.59	5.33	4.07	3.88
7	6.46	7.30	7.14	7.05	4.49	4.42
8	4.02	2.99	3.96	3.13	3.12	2.40
9	7.04	7.20	7.04	7.06	4.69	4.23
10	3.07	3.16	3.19	3.11	2.97	2.39
11	6.38	5.07	5.99	4.91	5.00	4.83
12	4.98	6.00	5.27	6.49	3.91	5.14
Average	4.95	4.48	5.00	4.52	4.00	3.62

MAE of the baseline system. Table VIII shows that the p-value of this t-test is smaller than 0.001, which indicates that the aforementioned null hypothesis is beaten and our modularized method achieves significant performance improvement against the baseline.

Table IV and VI present the MAE results of the systems using the musical scores as the templates. The dimension of BPK features is the same as the dimension of PKPP features. Compared with the baseline and the modularized systems taking audio recordings as templates, the musical score method does not achieve better performance for measuring the similarities. Instructors may have different playing styles, such as temporal variations or dynamics, which are not reflected in the musical score. Furthermore, PKPP sequences are the posterior probabilities of whether piano keys are pressed down, whereas the BPK sequences only contain 0 and 1. Therefore, we intend to use PKPP features from audio rather than the musical score as the template.

Table IX shows a comparison between our modularized system and the system in our previous conference paper [33]. Here we designed more comprehensive global matching features based on the DTW alignment and the CQT/PKPP features (7 dimensions in this work vs. 3 dimensions in [33]). Results in Table IX show that adding new similarity measurement features would help improve the performance.

C. Results and Discussions

1) The Modularized System: Figure 11 shows an audio piece whose annotated performance score is 88, but our systems predict that the performance score is around 46. After checking the DTW warping path and the song piece, we found that this student played the piece twice. As Figure 11 shows, the DTW algorithm can only align the second part of the student's audio with the instructor's audio. Therefore, our system cannot give the student a high score since we assume that the students only played this song once. However, the student played this song well, so the instructor gave a high score. The right part of the DTW warping path can also demonstrate his excellent performance.

2) The End-to-End Model: Table XI shows the MAE and STD of the end-to-end system. Each time we select one song as the fine-tuning and test set and the rest 11 songs as the pre-training set. We first train the model on the training part of the pre-training set and evaluate on the validation part of this set. There is not much difference between these 12 models. However, if we directly test the pre-trained model on the unseen test set, the performance is unsatisfactory compared with the LOSO configuration. The results of the model before the fine-tuning show that the end-to-end system may not generalize well on unseen testing data.

Therefore, we fine-tune the model with 2-fold cross-validation on data from the test song. Results show that all of the MAE improved after fine-tuning. Compared with the results of LOO in table III, the MAE of the end-to-end model is higher. It may be inappropriate to directly compare these two songs dependent systems, as the modularized system with LOO configuration did not use information from other songs, and the end-to-end system did not use all the information from the testing song (only half). However, the results of the modularized systems can provide a benchmark reference for us when we evaluate the end-to-end system, which shows that it is possible to train an end-to-end performance evaluation model with very limited data.

We also test the modularized system on the augmented dataset and find that no matter how we change the speed and pitch, the MAE of the modularized system is always greater than the MAE of the end-to-end model. Therefore, we believe that the

TABLE V
MAE AND STD OF THE MODULARIZED SYSTEM (AUDIO AS TEMPLATE, SONG INDEPENDENT (LOSO))

	chroma		88-d CQT		229-d CQT		88-12 PKPP		229-12 PKPP		229-88 PKPP	
SongID	MAE	STD	MAE	STD	MAE	STD	MAE	STD	MAE	STD	MAE	STD
1	6.95	5.64	6.85	6.13	6.94	6.18	6.41	5.53	6.87	6.13	6.47	6.01
2	3.26	2.33	3.37	2.43	3.42	2.59	3.21	2.36	3.91	2.69	2.95	2.47
3	4.73	3.04	4.08	2.73	4.08	2.80	4.64	3.26	4.20	3.20	5.43	3.56
4	5.11	4.52	5.05	4.54	5.27	4.54	5.07	4.45	5.34	4.76	4.91	4.57
5	4.08	3.03	3.26	2.73	3.73	2.93	4.75	3.43	4.74	3.36	3.80	2.94
6	8.22	5.57	7.46	5.05	7.04	4.95	6.43	4.82	5.64	5.58	5.21	4.91
7	8.02	9,77	7.06	8.99	7.00	8.58	6.45	8.04	6.21	7.76	6.75	8.48
8	3.27	2.75	3.47	2.76	3.34	2.54	2.95	2.39	2.89	2.18	3.02	2.36
9	6.85	7.41	6.47	7.83	6.22	7.93	6.83	6.96	6.74	7.04	6.57	7.29
10	3.65	2.79	5.04	3.10	4.88	3.06	3.89	2.87	3.41	2.36	3.77	2.60
11	6.38	5.75	6.14	5.44	5.69	5.01	5.07	4.35	4.80	4.07	4.77	4.38
12	5.37	8.00	5.00	7.64	4.85	7.69	5.93	8.30	5.25	8.04	4.72	7.47
Average	5.57	5.87	5.27	4.94	5.20	4.90	5.13	4.73	5.00	4.76	4.86	4.75

TABLE VI
MAE AND STD OF THE MODULARIZED SYSTEM (BPK AS TEMPLATE AND PKPP AS INPUT, SONG INDEPENDENT (LOSO))

	88-12 PKPP		229-12 PKPP		229-88 PKPP	
SongID	MAE	STD	MAE	STD	MAE	STD
1	8.47	4.70	9.11	4.81	6.49	5.89
2	4.83	3.17	7.10	3.73	3.65	2.72
3	3.81	2.69	3.51	2.52	3.91	2.88
4	6.16	4.92	6.00	4.78	6.04	4.34
5	4.10	3.24	3.69	3.20	4.83	3.07
6	6.46	6.72	6.37	6.46	5.38	6.23
7	9.04	10.78	8.66	10.63	6.01	9.28
8	4.63	3.69	6.51	4.11	3.28	2.50
9	8.90	9.54	8.33	9.01	7.84	7.72
10	3.57	2.58	3.43	2.47	4.71	2.98
11	6.18	5.57	6.27	5.25	5.49	5.26
12	6.29	8.07	6.56	8.36	5.70	7.27
Average	6.03	5.47	6.29	5.44	5.27	5.01

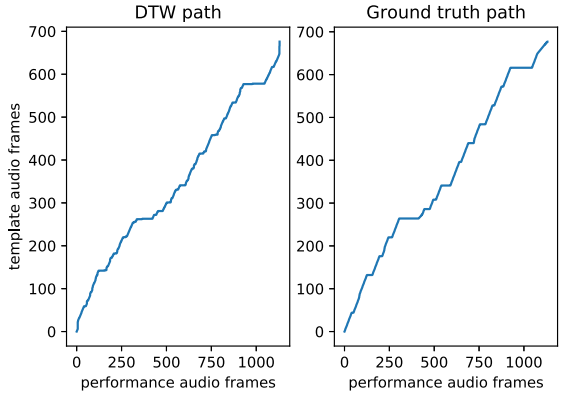


Fig. 8. An example of manually labeled alignment path.

end-to-end model has much potential to outperform the modularized system if we collect much more data in the future.

Figure 7 shows three attention maps during the fine-tuning process. Similar to the DTW alignment map, the attention map with a higher performance score also shows a better aligned path.

D. Discussion About the Role of the Alignment Path

Figure 3 and 7 show that the alignment path plays a very important role for the final score prediction. The alignment path of a high-performance recording is almost diagonal, whereas the alignment path of a recording with a lower performance score might have an irregular curve.

To investigate the influence of the alignment path on the final score prediction, we manually label the alignment paths of some audio pieces in our dataset. For convenience, we only label the 12th song, including about 200 audio pieces. We first convert these audio pieces to spectrograms and find the notes based on the amplitude. The onset can be easily labeled, but it is not easy to label the offset since the note may last for an unexpectedly long time. Hence, we can only estimate the offset intuitively. Figure 8 shows an example of the manually labeled alignment path. More alignment paths can be found at GitHub (https://github.com/spwb/PianoScore.git). Most of the manually labeled paths are very similar to our DTW estimated path in shape, which means that the DTW is quite effective for note alignment. Table X shows the average time deviation between the DTW estimated path and the corresponding manually aligned path for all 200 recordings of the 12^{th} song with different systems under the LOO configuration. From the results we can observe that, there are only moderate differences between those systems on the accuracy of alignment.

Then we extract the global matching features from these ground-truth and DTW estimated alignment paths, which is the same as mentioned in Section 3. There are two types of global matching features. One category is the path feature, including the frame ratio, skip, halt, angle, and overlapping area. The other group is the cost feature, including the average cost and standard deviation of the cost sequence.

For path features, we can directly extract them from the alignment path. For cost features, we extract them from the DTW cost matrix with the ground-truth or DTW-estimated alignment path. Then we can evaluate how the alignment path affects the final score prediction using both the path features and cost features. The results of different feature combinations with LOO configuration are shown in Figure 9. We can observe that the ground-truth paths show better performance than the estimated paths with only path features. Besides, the ground-truth path based path features and cost features together achieve the best performance. Moreover, we can find that a good alignment path

TABLE VII
THE CORRELATION COEFFICIENTS (CC) OF THE DIFFERENT SYSTEMS WITH LOO AND LOSO CONFIGURATIONS

	chroma	88-d CQT	229-d CQT	88-12 PKPP	229-12 PKPP	229-88 PKPP	human performance one vs. mean	human performance one vs. another	229-88 PKPP vs. one human annotator
CC (LOO)	0.74	0.77	0.79	0.80	0.80	0.81	0.88	0.67	0.72
CC (LOSO)	0.55	0.60	0.62	0.64	0.64	0.65	-	-	-

TABLE VIII
THE P-VALUE OF A ONE-TAILED T TEST WITH A NULL HYPOTHESIS THAT THE MAE OF THE MODULARIZED SYSTEM IS NOT SMALLER THAN THE MAE OF THE BASELINE SYSTEM

	modularized system	Baseline system	
	system	88-d CQT	229-d CQT
LOO	88-12 PKPP	4.49×10^{-5}	-
	229-12 PKPP	5.30×10^{-8}	6.14×10^{-3}
	229-88 PKPP	3.03×10^{-22}	4.59×10^{-13}
LOSO	88-12 PKPP	1.38×10^{-1}	-
	229-12 PKPP	2.99×10^{-3}	1.54×10^{-2}
	229-88 PKPP	3.19×10^{-10}	3.20×10^{-9}

TABLE IX
MAE AND STD OF THE PROPOSED MODULARIZED SYSTEM AND THE METHOD USED IN [33] UNDER THE LOO CONFIGURATIONS

System	number of global matching features	88-d CQT		229-88 PKPP	
		MAE	STD	MAE	STD
Ours	7	4.37	3.97	3.79	3.51
Pan's[33]	3	5.17	4.30	4.57	4.05

TABLE X
AVERAGE TIME DEVIATION (S) BETWEEN THE DTW ESTIMATED PATH AND THE MANUALLY ALIGNED PATH FOR THE 12^{th} SONG WITH DIFFERENT SYSTEMS UNDER THE LOO CONFIGURATION

88-d CQT	229-d CQT	88-12 PKPP	229-12 PKPP	229-88 PKPP
0.133	0.136	0.133	0.136	0.135

TABLE XI
MAE AND STD OF THE END-TO-END SYSTEM

	pre-trained model on the validation set		pre-trained model on the test set (song independent)		fine-tuned model on the test set (song dependent)	
SongID	MAE	STD	MAE	STD	MAE	STD
1	5.57	5.18	6.67	5.05	6.07	4.68
2	5.79	5.15	18.79	4.73	2.90	2.07
3	5.70	5.22	15.71	4.65	3.15	2.53
4	5.56	5.12	3.91	3.32	3.72	3.28
5	5.60	5.19	3.64	3.30	3.27	3.12
6	5.60	5.24	4.92	3.96	4.21	4.68
7	4.93	4.20	9.28	6.88	6.19	7.90
8	5.70	5.00	3.35	2.62	3.10	2.42
9	5.24	4.74	6.37	6.87	6.27	7.00
10	5.36	5.30	4.16	3.16	3.22	2.16
11	5.52	5.01	14.92	7.00	6.00	4.49
12	5.65	5.02	6.22	4.81	4.63	6.03
Average	5.52	5.03	8.41	7.18	**4.40**	**4.84**

is very important, because it not only generate path-based global matching features, but also provide a better cost along this path and produce a better performance. In addition, we can observe that the performance of cost features is sensitive to the quality

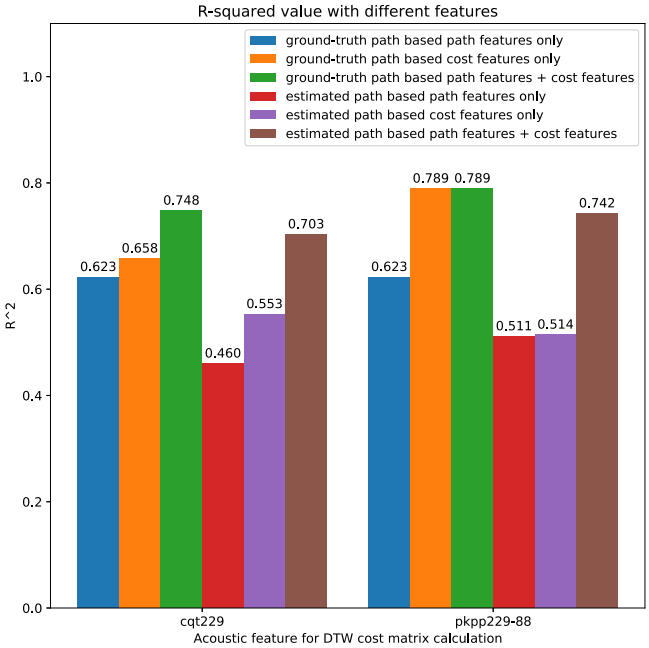


Fig. 9. R-squared values of different feature combinations with LOO configuration. The estimated path-based and estimated cost-based features are extracted from the DTW cost matrix with DTW alignment path.

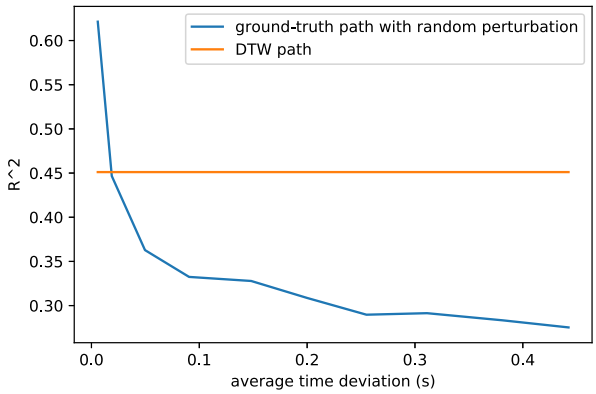


Fig. 10. The R-squared values of the manually labeled path with random perturbations. We also add an orange line to show result of the DTW path without random perturbation for comparison.

of the alignment path, as the PKPP features show a much better performance than the CQT features with the same ground-truth path, but it is not the case when the estimated DTW path is adopted. Therefore, our future works will focus on other advanced methods to further improve the accuracy of the estimated alignment path. In this work, although the DTW estimated path is not as good as the manually labeled one, combining both path-based and cost-based features together achieves significant

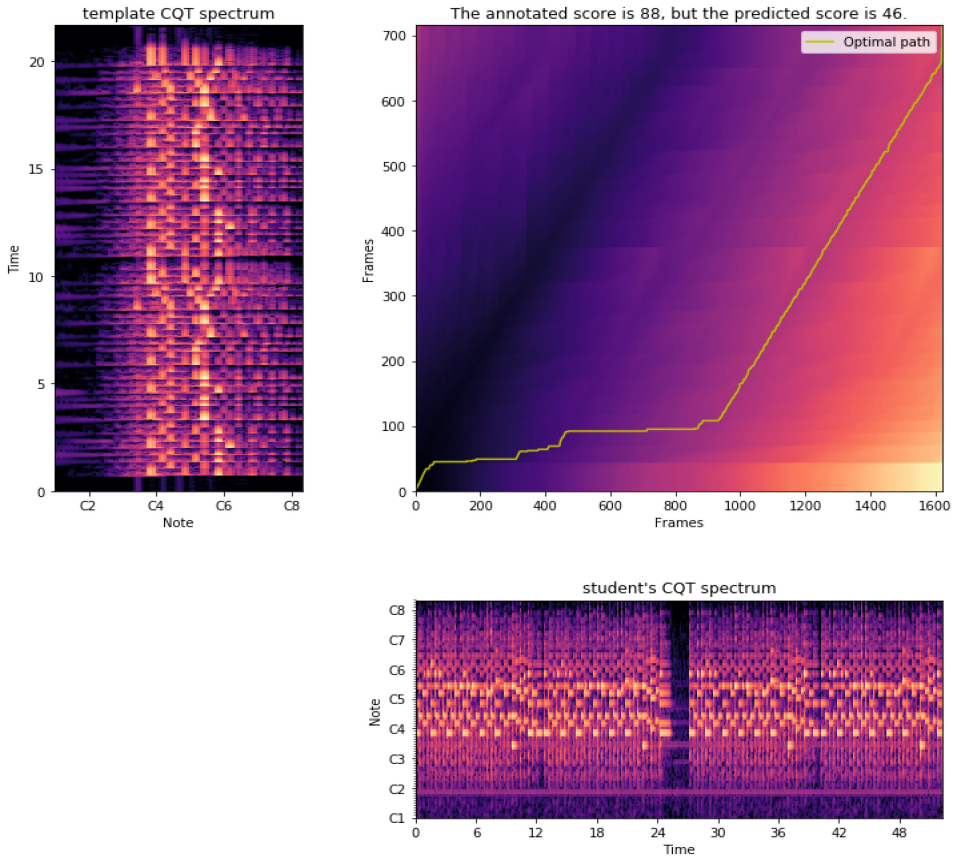


Fig. 11. An outlier whose annotated performance score is 88, but the predicted score is 46. The reason is that we assume that all students completely played a song once, but this student played this song twice.

improvement which show that they are complementary to each other.

To further investigate the influence of the alignment path quality to the final score prediction, we add some random perturbations to the manually labeled paths to evaluate the performance with path features only. We set a fixed small probability p such that for each point in an alignment path, the next n points will be replaced with the worst path with a probability of $p = 0.05$. This probability is quite large since we check this random perturbation for each point in the alignment path and each alignment path contains thousands of points. As Figure 10 shows, with average time deviation increases (as n increases), the final R-squared values decrease rapidly even with a small average time deviation. This means that the overall regression performance is quite sensitive to the quality of alignment. Furthermore, although our DTW based alignment also achieves around 0.13 average time deviation, the corresponding R-squared values is much higher than the one using the ground-truth alignment with random perturbation.

VI. Conclusion

In this paper, we present a modularized system and an end-to-end framework for the piano performance assessment task. For the modularized system, we use a CNN acoustic model to extract the PKPP, align the PKPP sequences using the DTW algorithm, extract the global similarity features, and finally predict a performance score. Our acoustic model extracts a robust PKPP vector for each frame, replacing the relatively low-level CQT spectrum feature, and applying the DTW algorithm to produce an accumulated matrix and a warping path. For the end-to-end model, we use an end-to-end attention-based deep neural network approach to predict the performance score. We proposed a speed-up and pitch-shift data augmentation strategy, which is proved to be useful in our experiments. We first train the model on 11 songs, then fine-tune the model on the remaining unseen test song. After fine-tuning, the performance of the end-to-end method is comparable to the modularized system under the song-dependent setup.

One of our significant contributions is that we propose a robust acoustic model and apply this model to the baseline system. Our results show that our modularized system with the acoustic model provides better performance than the baseline system. We also explore the leave-one-song-out cross-validation method in our song-independent system. Another contribution is that we propose an end-to-end model with a well-designed training strategy. Besides, we employ a data augmentation method, which is also important for this evaluation task since we only have very limited data. We believe that the end-to-end method has good potential in the future when large scale training data is available.

Limitations of our system do exist. The quality of the alignment path is crucial for both path features and cost features in the modularized system. The current DTW estimated path is not accurate enough and our future works will focus on other advanced methods to further improve the accuracy of the estimated alignment path. Moreover, we plan to manually annotate

the ground truth alignment paths for all the recordings, and study the relation and contribution of alignment path accuracy and different selection of cost features towards the overall system performance.

Another limitation is that the end-to-end model cannot learn song-independent information well. Without fine-tuning, the model does not give satisfactory results. In the future, we plan to collect more data and propose a more robust deep neural network-based approach to solve the problem of song-independent prediction.

REFERENCES

[1] B. E. Russell, "The empirical testing of musical performance assessment paradigm," Ph.D. dissertation, Univ. Miami, 2010.

[2] A. Vidwans, S. Gururani, C. Wu, V. Subramanian, R. Swaminathan, and A. Lerch, "Objective descriptors for the assessment of student music performances," in *Proc. Int. Conf. Semantic Audio*, 2017.

[3] L. Huanhuan, "Computer assisted music instrument tutoring applied to violin practice," Master Thesis, National University of Singapore, [Online]. Available: https://scholarbank.nus.edu.sg/handle/10635/16795, 2009.

[4] K. A. Pati, S. Gururani, and A. Lerch, "Assessment of student music performances using deep neural networks," *Appl. Sci.*, vol. 8, no. 4, p. 507, 2018.

[5] B. Bozkurt, O. Baysal, and D. Yüret, "A dataset and baseline system for singing voice assessment," in *Proc. Int. Symp. Comput. Music Multidisciplinary Res.*, 2017, pp. 25–28.

[6] S. Morita, N. Emura, M. Miura, S. Akinaga, and M. Yanagida, "Evaluation of a scale performance on the piano using spline and regression models," in *Proc. Int. Symp. Perform. Sci.*, 2009, pp. 77–82.

[7] S. Akinaga, M. Miura, N. Emura, and M. Yanagida, "Toward realizing automatic evaluation of playing scales on the piano," in *Proc. Int. Conf. Music Perception Cogn*. Citeseer, 2006, pp. 1843–1847.

[8] M. Piszczalski and B. A. Galler, "Automatic music transcription," *Comput. Music J.*, pp. 24–31, 1977.

[9] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks," in *Proc. Int. Soc. Music Inf. Conf.*, 2013, pp. 335–340.

[10] A. P. Klapuri, "Automatic music transcription as we know it today," *J. New Music Res.*, vol. 33, no. 3, pp. 269–282, 2004.

[11] Wikipedia, "Piano Roll." [Online]. Available: https://en.wikipedia.org/wiki/Piano_roll

[12] C. Raphael, "Automatic transcription of piano music," in *Proc. 3rd Int. Conf. Music Inf. Retrieval*, Paris, France, Oct. 2002.

[13] M. Marolt, "A connectionist approach to automatic transcription of polyphonic piano music," *IEEE Trans. Multimedia*, vol. 6, no. 3, pp. 439–449, Jun. 2004.

[14] E. Benetos and S. Dixon, "A shift-invariant latent variable model for automatic music transcription," *Comput. Music J.*, vol. 36, no. 4, pp. 81–94, 2012.

[15] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[16] A. Khlif and V. Sethu, "An iterative multi range non-negative matrix factorization algorithm for polyphonic music transcription," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2015, pp. 330–335.

[17] S. A. Abdallah and M. D. Plumbley, "Polyphonic music transcription by non-negative sparse coding of power spectra," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, vol. 4, 2004, pp. 318–325.

[18] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 18, no. 3, pp. 528–537, Mar. 2010.

[19] N. Bertin, R. Badeau, and E. Vincent, "Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 18, no. 3, pp. 538–549, Mar. 2010.

[20] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, 2012, pp. 121–124.

[21] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *Proc. Int. Conf. Mach. Learn.*, pp. 1881–1888, 2012.

[22] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2016, pp. 475–481.

[23] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 24, no. 5, pp. 927–939, May 2016.

[24] C. Hawthorne *et al.*, "Onsets and frames: Dual-objective piano transcription," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 50–57.

[25] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 18, no. 6, pp. 1643–1654, Aug. 2010.

[26] S. Ewert, M. Muller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process*., 2009, pp. 1869–1872.

[27] B. Li, Z. Duan, B. Li, and Z. Duan, "An approach to score following for piano performances with the sustained effect," *IEEE/ACM Trans. Audio, Speech Lang. Process.*, vol. 24, no. 12, pp. 2425–2438, Dec. 2016.

[28] S. Wang, S. Ewert, and S. Dixon, "Robust joint alignment of multiple versions of a piece of music," in *Proc. 15th Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, 2014, pp. 83–88.

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

[30] C.-C. Chiu *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process*., 2018, pp. 4774–4778.

[31] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A. neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3156–3164.

[32] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6000–6010.

[33] J. Pan *et al.*, "An audio based piano performance evaluation method using deep neural network based acoustic modeling," in *Proc. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 3088–3092.

[34] M. Müller, "Dynamic Time Warping," *Inf. Retrieval Music Motion*, pp. 69–84, 2007.

[35] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[36] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks," in *Proc. Int. Soc. Music Inf. Retrieval*, 2015, pp. 121–126.

[37] Y. Guo, Q. Qian, and Y. Yan, "Robust voice activity detection based on adaptive sub-band energy sequence analysis and harmonic detection," in *Proc. 8th Annu. Conf. Int. Speech Commun. Assoc.*, pp. 2949–2952, 2007.

[38] J. C. Brown, "Calculation of a constant q spectral transform," *J. Acoustical Soc. Amer.*, vol. 89, no. 1, pp. 425–434, 1991.

[39] B. McFee *et al.*, "Librosa: Audio and music signal analysis in python," in *Proc. 14th Python Sci. Conf.*, 2015, pp. 18–25.

[40] S. Nakagawa and H. Nakanishi, "Speaker-independent english consonant and japanese word recognition by a stochastic dynamic time warping method," *IETE J. Res.*, vol. 34, no. 1, pp. 87–95, 1988.

[41] B.-H. Juang, "On the hidden markov model and dynamic time warping for speech recognition," *ATT Bell Lab. Tech. J.*, vol. 63, no. 7, pp. 1213–1243, 1984.

[42] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, 2007.

[43] S. Gururani and A. Lerch, "Automatic sample detection in polyphonic music," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2017 pp. 264–271,.

[44] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. 34th Int. Conf. Mach. Learn.-Volume 70*, 2017, pp. 1243–1252.

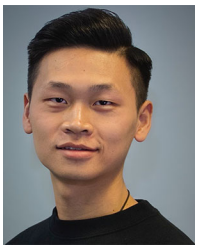
[45] Y. Zhang, M. Yu, N. Li, C. Yu, J. Cui, and D. Yu, "Seq2Seq attentional siamese neural networks for text-dependent speaker verification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process*, 2019, pp. 6131–6135.

[46] L. I.-K. Lin, "A concordance correlation coefficient to evaluate reproducibility," *Biometrics*, vol. 45, no. 1, pp. 255–268, 1989.

[47] M. Schmitt, N. Cummins, and B. W. Schuller, "Continuous emotion recognition in speech - do we need recurrence?" in *Proc. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 2808–2812.

[48] E. Jones *et al.*, "SciPy: Open source scientific tools for python," 2001.

Weiqing Wang received the B.S. degree in computer science from Sun Yat-sen University, Guangzhou, China, in 2018. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. From 2018 to 2019, he was a Research Assistant with Duke Kunshan University Suzhou, China. His research interests include the areas of speaker diarization and speaker verification.



Jing Pan received the B.E. degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2016 and the M.S. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2018. His research interests include the areas of speech processing and natural language processing.

Hua Yi received the M.A. degree in music from the Shandong University of Arts, Jinan, China, in 2011. She is currently a Piano Instructor with the Department of Early Childhood Education, Shandong Yingcai University, Jinan, China. Her research interests include the areas of piano instruction and learning.

Zhanmei Song received the Ph.D. degree in early childhood education from East China Normal University, Shanghai, China, in 2012. She is currently a Professor with the College of Education, Wenzhou University, Wenzhou, China. Her research interests include compensation education for children left-behind and kindergarten curriculum. She is the Associate General Secretary of the National Association of ECE, Chinese Society of Education.

Ming Li (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in May 2013. He is currently an Associate Professor of electrical and computer engineering with Duke Kunshan University, Suzhou, China and an Adjunct Professor with the School of Computer Science, Wuhan University, Wuhan, China. His research interests include the areas of audio, speech, language processing, multimodal behavior signal analysis, and interpretation.