Note-level Automatic Guitar Transcription Using Attention Mechanism

Sehun Kim*, Tomoki Hayashi*†, Tomoki Toda*
*Nagoya University, Nagoya, Japan
†Human Dataware Lab. Co., Ltd., Nagoya, Japan
*{kim.sehun, hayashi.tomoki}@g.sp.m.is.nagoya-u.ac.jp,
†hayashi@hdwlab.co.jp, *tomoki@icts.nagoya-u.ac.jp

***Abstract*—We propose a method that effectively generates a note-level transcription from a guitar sound signal. In recent years, there have been many successful guitar transcription systems. However, most of them generate a frame-level transcription rather than a note-level transcription. Furthermore, it is usually difficult to effectively model long-term characteristics. To address these problems, we propose a novel model architecture using an attention mechanism along with a convolutional neural network (CNN). Our model is capable of modeling both short-term and long-term characteristics of a guitar sound signal and a corresponding guitar transcription. A beat-informed quantization is implemented to generate a note-level transcription. Furthermore, multi-task learning with frame-level and note-level estimations is also implemented to achieve robust training. We conducted experimental evaluations on our method using a publicly available acoustic guitar dataset. We confirmed that 1) the proposed method significantly outperforms the conventional method based on a CNN in frame-level estimation performance and that 2) the proposed method can also generate note-level guitar transcription while preserving high estimation performance.**

***Index Terms*—automatic guitar transcription, note-level, attention mechanism, multi-task learning**

I. Introduction

The guitar is a popular instrument for both professional musicians and hobbyists. However, when a guitar player wants to play a tune performed by another guitarist, unless there is already a music score, a player has to transcribe the correct pitch and fingering by either listening to it or watching a performance video and assume by looking at their finger position. This process of manually transcribing a guitar performance is a non-trivial task even for skilled musicians and can be time-consuming and inaccurate [1].

To address this, there have been many studies on automatic guitar transcription [1]–[6]. Automatic guitar transcription is the task of generating a symbolic transcription of the music from an audio recording. Since guitar is a polyphonic instrument, the transcription of a guitar performance is relatively difficult compared with that of a monophonic instrument. Moreover, since a guitar has six strings and their ranges of possible pitches overlap, it is difficult to predict exactly which string is used when a note is played. This ambiguity is a characteristic of multistring instruments such as the guitar. For this reason, a tablature score, as shown in Fig. 1 (c), which is a type of music score that contains the timing, string, and fret positioning of each note, has been a popular choice for annotating a guitar performance.

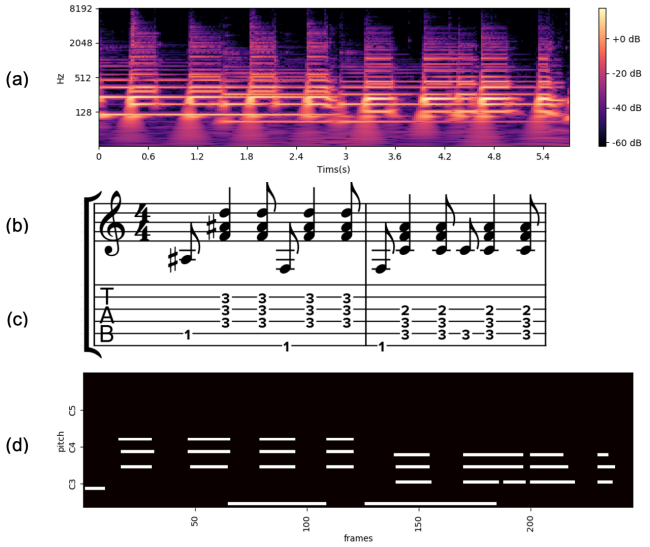

Fig. 1: Examples of (a) spectrogram of a guitar sound, (b) standard music score, (c) tablature score, and (d) frame-level transcription.

Wiggins and Kim proposed a convolutional neural network (CNN)-based system called TabCNN [2], which is capable of directly estimating a frame-level[1] tablature of a guitar performance, i.e., generating Fig. 1 (d) from Fig. 1 (a). Although TabCNN is capable of generating a tablature transcription, it has the limitations of not being able to model long-term characteristics of a guitar performance and the output consisting of a frame-level transcription instead of a note-level transcription. Since a music score that can directly be read by humans, such as in Fig. 1 (b) and Fig. 1 (c), consists of annotation in musical notes rather than frames, note-level annotation is more desirable in terms of generating a human-readable music score.

To overcome the limitations of conventional methods, we propose a novel method[2] that generates a note-level tablature from a spectrogram of a guitar sound signal and a given beats-per-minute (BPM) information. Note that our method does not

[1]Term used to describe that the time resolution is in units of frames.
[2]**Source code available**: https://github.com/KimSehun725/Tab-estimator

detect onset. Therefore, the output is a saliency representation of a tablature. Our contributions are summarized as follows.

- We introduce an attention mechanism, which is demonstrated to be effective at modeling long-term characteristics without making the size of a network very large [7].
- We implement a beat-informed quantization, which enables our model to generate a note-level transcription.
- We use both frame-level and note-level outputs to perform multi-task learning to achieve robust training.
- Experimental evaluation results demonstrate that our proposed method not only outperforms a state-of-the-art guitar transcription system in frame-level estimation, but is also capable of generating a note-level transcription while preserving high estimation performance.

II. Related work

A. Automatic tablature estimation

In the goal of developing an automatic tablature estimation system, several methods mainly based on audio signal processing have been proposed [6]. Also, there have been several approaches that use probabilistic models [5], [8]. In [5], two-step approach is used. The first step is to estimate the pitch of each note being played. The second step is to estimate the finger positioning by combining the estimated pitch information and the physical limitations of the possible fingering to estimate the best fingering position. Since these methods process information sequentially, information from downstream components cannot inform upstream components.

In an attempt to overcome the limitations of these multi-step approach, the second approach directly estimates a tablature from an audio signal using a deep neural network (DNN). Inspired by the CNN-based polyphonic music transcription model [9], Wiggins et al. proposed a CNN-based architecture that uses constant-Q transform (CQT) [10] as an input acoustic feature to estimate the frame-level tablature and named it TabCNN [2].

B. Note-level music transcription

Shibata et al. presented an automatic piano transcription system that converted polyphonic audio recordings into musical scores [11]. They used CNNs to generate a MIDI-like sequence and used the metrical hidden Markov model (HMM) for rhythm quantization. Hiramatsu et al. proposed a bidirectional long-short term memory (BiLSTM)-based network that effectively converted a MIDI-like note sequence into a note-based annotation [12]. Cogliati et al. proposed a method of converting a MIDI file to a musical score by using a combination of a rule-based and probabilistic models [13].

Although these methods can be applied to a guitar transcription system, they require a multistep approach, making it difficult to jointly optimize all models through the entire process.

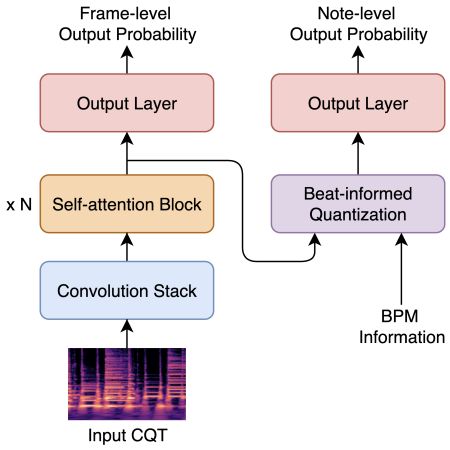

Fig. 2: Overview of our proposed model architecture.

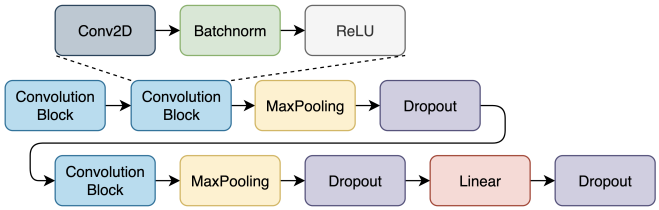

Fig. 3: Network structure of the convolution stack used in our proposed model.

III. Proposed method

The structure of our model is generally inspired by TabCNN [2] and the Conformer block architecture [14] with the addition of a beat-informed quantization layer. An overview of our proposed model architecture is shown in Fig. 2. The proposed network contains four main parts: a convolution stack, self-attention blocks, a beat-informed quantization layer, and two output layers. The convolution stack is used as a local feature extractor and the self-attention block is used for seeking global interactions based on the features extracted from the convolution stack. Beat-informed quantization is used to effectively quantize latent features corresponding to the given BPM information without sacrificing much information. Lastly, two output layers each generate frame-level and note-level outputs, and we use these two outputs to implement multi-task learning.

A. Convolution stack

The convolution stack consists of several 2D convolution layers, max pooling layers, dropout layers, and a linear layer. The structure of the convolution stack is shown in Fig. 3. First, input features go through two convolution blocks, which consist of sequential operations of 2D convolution, batch normalization, and the Rectified Linear Unit (ReLU) activation function. Next, the latent features generated from two sequential convolution blocks are subsampled by a max pooling layer. Then, another convolution block and a max pooling layer further refine the latent features extracted from the previous steps. Lastly, a linear layer is added as an output

layer to reduce the dimension. In addition, three dropout layers are placed after the max pooling layers and the linear layer to prevent overfitting.

B. Self-attention block

For the self-attention blocks, we employ Conformer [14], which is a convolution-augmented Transformer [15] architecture. The Conformer block consists of a dot-product self-attention module with relative positional encoding [16], followed by a convolution module that contains pointwise and depthwise convolution operations. In addition, both modules have a residual connection from the previous step. The self-attention and convolution modules are sandwiched by two feedforward modules with a half-step residual connection.

C. Beat-informed quantization

Beat-informed quantization is a non-trainable operation. Its goal is to quantize a frame-level sequence into a note-level sequence, i.e., quantize in time with the BPM scaling factor. In this operation, the quantization ratio $K \in \mathbb{R}$ can be calculated as

$$K = \frac{60 \cdot f_s}{q/4 \cdot B \cdot h}, \tag{1}$$

where f_s denotes the sampling rate, q denotes the tatum, i.e., the minimum quantization unit in the qth note, B denotes the BPM, and h denotes the hop length.

Simple ways to approximately quantize with a ratio of a real number are to perform integer ratio interpolation followed by integer ratio decimation, and to perform a simple subsampling process that reduces the data size by selecting a subset of the original data [17]. However, these methods discard information between samples when $K > 1$. Furthermore, as K increases, more information is discarded.

In an attempt to preserve as much information as possible when performing beat-informed quantization, we propose a method of beat-informed quantization. The method is expressed as

$$\begin{aligned}\hat{\mathbf{X}}(t_n) = \frac{1}{K}\big[&(\lceil Kt_n \rceil - Kt_n)\mathbf{X}(\lfloor Kt_n \rfloor) \\ &+ \sum_{t_f=\lceil Kt_n \rceil}^{\lfloor K(t_n+1) \rfloor - 1} \mathbf{X}(t_f) \\ &+ \{K(t_n+1) - \lfloor K(t_n+1) \rfloor\}\, \mathbf{X}(\lfloor K(t_n+1) \rfloor)\big],\end{aligned} \tag{2}$$

where $\mathbf{X}$ denotes input latent features generated from the self-attention blocks with a framewise sequence length, $\hat{\mathbf{X}}$ denotes an output with a notewise sequence length, and t_f and t_n denote framewise and notewise times, respectively.

D. Multi-task learning

In order to obtain probability distribution over each string, we use a linear layer with the string-wise softmax function [2] for the output layers of our system.

We train our model by multi-task learning with frame-level and note-level estimations. In addition, we employ the guided attention loss [18] to make the training process more stable and converge faster. The loss function of our system $\mathcal{L}_{total}$ is espressed as

$$\mathcal{L}_{total} = \mathcal{L}_{frame} + \mathcal{L}_{note} + \mathcal{L}_{att}, \tag{3}$$

where $\mathcal{L}_{frame}$ denotes the frame loss, $\mathcal{L}_{note}$ denotes the note loss, and $\mathcal{L}_{att}$ denotes the guided attention loss.

The frame and note losses of our model can be respectively expressed as

$$\begin{aligned}\mathcal{L}_{frame} = -\frac{1}{6 \cdot 21 \cdot T} \sum_{s=1}^{6} \sum_{f=1}^{21} \sum_{t=1}^{T} \{&y_{s,f,t} \log(\hat{y}_{s,f,t}) \\ &+ (1 - y_{s,f,t}) \log(1 - \hat{y}_{s,f,t})\},\end{aligned} \tag{4}$$

$$\begin{aligned}\mathcal{L}_{note} = -\frac{1}{6 \cdot 21 \cdot N} \sum_{s=1}^{6} \sum_{f=1}^{21} \sum_{n=1}^{N} \{&z_{s,f,n} \log(\hat{z}_{s,f,n}) \\ &+ (1 - z_{s,f,n}) \log(1 - \hat{z}_{s,f,n})\},\end{aligned} \tag{5}$$

where y denotes the frame-level ground truth label, $\hat{y}$ denotes the frame-level prediction from the model, z denotes the note-level ground truth label, $\hat{z}$ denotes the note-level prediction from the model, s denotes the string number, f denotes the fret classes, and T and N denote the framewise and notewise lengths, respectively.

The guided attention loss can be expressed as

$$\mathcal{L}_{att}(\mathbf{A}) = \frac{\alpha}{T^2} \sum_{t_1=1}^{T} \sum_{t_2=1}^{T} \mathbf{A}(t_1, t_2)[1 - \exp\{\frac{-(t_1/T - t_2/T)^2}{2g^2}\}], \tag{6}$$

where $\mathbf{A}$ denotes the attention weight matrix and α denotes the scaling coefficient. In addition, t_1 and t_2 denote the source and target frames, respectively, and T denotes the total number of frames. Lastly, g is a hyperparameter for controlling the strength of the effect of the guided attention.

IV. Experimental evaluation

A. Experimental conditions

As our training and testing dataset, we used GuitarSet [19], which contains acoustic guitar recordings and corresponding annotations. Since GuitarSet contains performances of six different guitar players, we used a sixfold cross-validation method with a rotating test player to evaluate our model. Furthermore, we set the training/validation ratio to 0.9.

The training conditions of the baseline model (TabCNN) and our proposed model are shown in Table I. For the baseline model, we followed the original training conditions proposed in [2]. When training our proposed model, instead of using a fixed learning rate in the entire training process, we used a step decay learning rate (Step LR) scheme. In our experiment, we set 0.005 as the initial learning rate and reduced it by half after every 32 epochs. For the optimizer, we employed Rectified Adam (RAdam) [20]. All network parameters were initialized using Xavier's initializer [21].

We implemented the self-attention block using the ESPNet2 framework [22], and we used one layer of self-attention block and one attention head for the self-attention block, and we set

TABLE I: Comparison of training conditions.

	TabCNN	Proposed
Training epochs	8	192
Minibatch size	128	32
Optimizer	Adadelta	RAdam
Learning rate	1.0	Step LR
Number of CQT bins	192	192
Bins per octave	24	24
Hop length	512	512
Downsampling rate	22050Hz	22050Hz

TABLE II: Frame-level tablature estimation metrics for our proposed system, compared with a baseline system. For all metrics, we report the mean and standard deviation over the entire dataset.

	Precision	Recall	F1	TDR
TabCNN (frame-level)	**0.809** ± 0.029	0.696 ± 0.061	0.748 ± 0.047	0.899 ± 0.033
Proposed (frame-level)	0.789 ± 0.027	**0.780** ± 0.040	**0.781** ± 0.029	0.918 ± 0.020
Proposed (note-level)	0.781 ± 0.031	0.777 ± 0.039	0.775 ± 0.029	**0.919** ± 0.021

$g = 0.4$, $\alpha = 1$ for the guided attention loss. In addition, the dimension of the attention mechanism was set to 64. Finally, we set the tatum q to the 16th note.

For the metrics to evaluate our system, we used precision, recall, F1 score, and tablature disambiguation rate (TDR) [2]. TDR is computed by dividing the total number of correctly identified string-fret combinations by the total number of correctly identified pitches. TDR measures how frequently pitches that are correctly identified are assigned to the correct fingering positions. The equation for calculating TDR is

$$\tau = \frac{\mathbf{e}^T(\mathbf{Z}_{\mathrm{gt}} \circ \mathbf{Z}_{\mathrm{pred}})\mathbf{e}}{\mathbf{e}^T(\mathbf{Y}_{\mathrm{gt}} \circ \mathbf{Y}_{\mathrm{pred}})\mathbf{e}}, \tag{7}$$

where τ denotes TDR, $\mathbf{e}$ denotes a vector of all ones, $\mathbf{Z}$ denotes the tablature, and $\mathbf{Y}$ denotes the pitches. Subscripts gt and pred denote the ground truth and the prediction from the guitar transcription model, respectively.

B. Results

We compare the baseline model with our proposed model in Table II. Regarding frame-level tablature estimation, our proposed model outperforms the baseline model for the recall, F1 score, and TDR. The baseline model shows slightly better results only for precision. In addition, the proposed method also achieves high performance in note-level estimation while limiting the performance degradation in precision, recall, and F1 score to less than 1% compared with those in the frame-level estimation. Note that the TDR of 0.918 indicates that over 91% of correctly identified pitches are assigned to the correct fingering.

A sample output of our system is shown in Fig. 4. Note that by looking at the attention map, we can see that the attention mechanism not only attends to the exact corresponding time frame when generating the output, it also attends to frames around it in a rectangular region roughly corresponding to each note or chord.

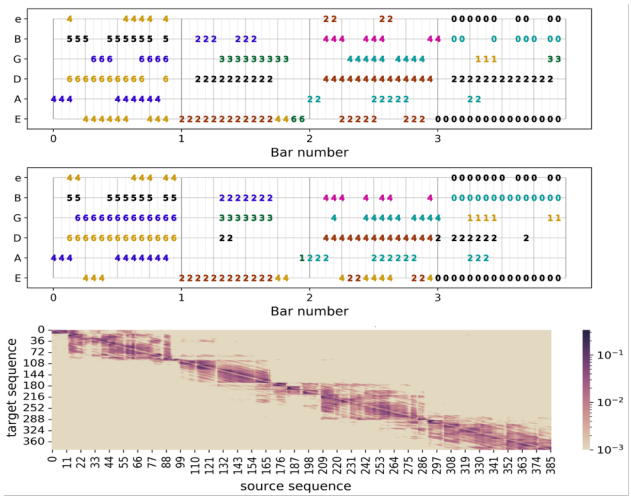

Fig. 4: Sample set of (top) ground truth label, (middle) estimation result from our system, and (bottom) corresponding attention map. The same color represents the same pitch name.

By analyzing the estimation result, we found that the most common type of error was the insertion or deletion of a note in a higher octave. This type of error was also reported in [2]. This is likely due to the fact that a sound of a guitar often contains a strong harmonic overtone and the network fails to detect the correct string by its timbre.

C. Ablation study

We evaluated the effects of various elements of our system via an ablation study. To quantitatively analyze the effect of each element of our proposed method, we used Cohen's d for the F1 score of the note-level tablature estimation as an effect size to evaluate the effectiveness of each element. Cohen's d is computed as

$$d = \frac{|\bar{x_1} - \bar{x_2}|}{\sqrt{(\sigma_1^2 + \sigma_2^2)/2}}, \tag{8}$$

where $\bar{x_1}$ and $\bar{x_2}$ denote sample means, and σ_1^2 and σ_2^2 denote unbiased variances. It has been suggested that d = 0.2 represents a 'small' effect size, 0.5 represents a 'medium' effect size, and 0.8 represents a 'large' effect size [23]. We compared our proposed model (a) with the vanilla Transformer encoder instead of Conformer, (b) without the attention mechanism, (c) with the selection-based quantization method instead of the proposed beat-informed quantization method, (d) with the mel-spectrogram as the input feature instead of the CQT, (e) without the guided attention, and (f) without multi-task learning ($\mathcal{L}_{total} = \mathcal{L}_{note} + \mathcal{L}_{att}$).

The result of the ablation study for our proposed model architecture is shown in Table III. Additionally, the effect size of each component is shown in Table IV. The result of the ablation study shows that (b), (c), and (d) have a large effect, (a) has a medium to large effect, (e) has a small effect on the performance of the model, and (f) multi-task learning is essential for training our model. This result demonstrates the effectiveness of 1) the attention mechanism, 2) the convolutional augmentation on the attention mechanism,

TABLE III: Ablation study on note-level tablature estimation for our proposed model. For all metrics, we report the mean and standard deviation over the entire dataset.

	Precision	Recall	F1	TDR
Proposed	**0.781** ± 0.031	**0.777** ± 0.039	**0.775** ± 0.029	0.919 ± 0.021
(a) With vanilla Transformer encoder	0.768 ± 0.035	0.748 ± 0.055	0.753 ± 0.038	0.905 ± 0.029
(b) Without attention mechanism	0.754 ± 0.033	0.703 ± 0.039	0.724 ± 0.032	0.880 ± 0.026
(c) With selection-based quantization method	0.692 ± 0.031	0.617 ± 0.032	0.646 ± 0.024	**0.921** ± 0.018
(d) With mel-spectrogram as input feature	0.726 ± 0.023	0.696 ± 0.056	0.706 ± 0.038	0.880 ± 0.030
(e) Without guided attention	0.777 ± 0.026	0.776 ± 0.046	0.772 ± 0.032	0.917 ± 0.023
(f) Without multi-task learning	Failed to train			

TABLE IV: Effect size of (a)-(e) regarding F1 score with reference to our proposed model.

	Effect size
(a) With vanilla Transformer encoder	0.65
(b) Without attention mechanism	1.67
(c) With selection-based quantization method	4.85
(d) With mel-spectrogram as input feature	2.04
(e) Without guided attention	0.10

3) our proposed beat-informed quantization method, and 4) using CQT as an input feature instead of the mel-spectrogram. Although the guided attention had little impact on the performance of the model, it helped make the training procedure more stable and converge faster.

V. Conclusion

In this paper, we have proposed a novel automatic guitar transcription method that uses an attention mechanism, beat-informed quantization, and a multi-task learning scheme. Our method not only significantly outperforms the conventional method of generating a frame-level transcription, but is also capable of generating a note-level transcription while preserving high estimation performance. The results of experimental evaluations have shown the effectiveness of 1) the attention mechanism, 2) our proposed beat-informed quantization method, and 3) multi-task learning with frame-level and note-level estimations.

Acknowledgment

This work was partly supported by JST CREST Grant Number JPMJCR19A3, Japan.

References

[1] S. Goldstein and Y. Moses, "Guitar music transcription from silent video," in *British Machine Vision Conference (BMVC) 2018, Newcastle, UK, September 3-6*, 2018.

[2] A. Wiggins and Y. E. Kim, "Guitar tablature estimation with a convolutional neural network," in *Proc. 20th International Society for Music Information Retrieval Conference (ISMIR) 2019, Delft, The Netherlands, November 4–8*, 2019, pp. 284–291.

[3] E. J. Humphrey and J. P. Bello, "From music audio to chord tablature: Teaching deep convolutional networks to play guitar," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6974–6978.

[4] X. Fiss and A. Kwasinski, "Automatic real-time electric guitar audio transcription," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 373–376.

[5] K. Yazawa, K. Itoyama, and H. G. Okuno, "Automatic transcription of guitar tablature from audio signals in accordance with player's proficiency," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3122–3126.

[6] C. Kehling, J. Abeßer, C. Dittmar, and G. Schuller, "Automatic tablature transcription of electric guitar recordings by estimation of score and instrument-related parameters," in *Proc. 17th International Conference on Digital Audio Effects (DAFx-14)*, 2014.

[7] J. Salazar, K. Kirchhoff, and Z. Huang, "Self-attention networks for connectionist temporal classification in speech recognition," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7115–7119.

[8] G. Hori, H. Kameoka, and S. Sagayama, "Input-output HMM applied to automatic arrangement for guitars," *Journal of Information Processing*, vol. 21, no. 2, pp. 264–271, 2013.

[9] R. Bittner, B. McFee, J. Salamon, P. Li, and J. Bello, "Deep salience representations for F0 estimation in polyphonic music," in *Proc. 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 63–70.

[10] J. Youngberg and S. Boll, "Constant-Q signal analysis and synthesis," in *1978 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, 1978, pp. 375–378.

[11] K. Shibata, E. Nakamura, and K. Yoshii, "Non-local musical statistics as guides for audio-to-score piano transcription," *Information Sciences*, vol. 566, pp. 262–280, 2021.

[12] Y. Hiramatsu, E. Nakamura, and K. Yoshii, "Joint estimation of note values and voices for audio-to-score piano transcription," in *Proc. 22nd International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 278–284.

[13] A. Cogliati, D. Temperley, and Z. Duan, "Transcribing human piano performances into music notation," in *Proc. 17th International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 758–764.

[14] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, pp. 5998–6008, 2017.

[16] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. 57th Annu. Meeting of the Assoc. for Computational Linguistics*, Jul. 2019, pp. 2978–2988.

[17] R. Crochiere and L. Rabiner, "Interpolation and decimation of digital signals—A tutorial review," *Proc. IEEE*, vol. 69, no. 3, pp. 300–331, 1981.

[18] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4784–4788.

[19] Q. Xi, R. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "Guitarset: A dataset for guitar transcription," in *Proc. 19th International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, Sep. 2018, pp. 453–460.

[20] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *International Conference on Learning Representations*, 2020.

[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research (PMLR), vol. 9, 13–15 May 2010, pp. 249–256.

[22] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, J. Shi, S. Watanabe, K. Wei, W. Zhang, and Y. Zhang, "Recent developments on espnet toolkit boosted by conformer," in *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5874–5878.

[23] J. Cohen, *Statistical power analysis for the behavioral sciences*. Routledge, 1988.