STREAMING PIANO TRANSCRIPTION BASED ON CONSISTENT ONSET AND OFFSET DECODING WITH SUSTAIN PEDAL DETECTION

Weixing Wei[1] **Jiahao Zhao**[1] **Yulun Wu**[2] **Kazuyoshi Yoshii**[3]
[1]Graduate School of Informatics, Kyoto University, Japan
[2]School of Computer Science and Technology, Fudan University, China
[3]Graduate School of Engineering, Kyoto University, Japan
{wei.weixing.23w, zhao.jiahao.56h}@st.kyoto-u.ac.jp, yoshii.kazuyoshi.3r@kyoto-u.ac.jp

ABSTRACT

This paper describes a streaming audio-to-MIDI transcription method that can sequentially translate a piano recording into a sequence of note-on and note-off events. The sequence-to-sequence learning nature of this task may call for using a Transformer model, which has been used for offline transcription and could be extended for streaming transcription with a causal restriction of the attention mechanism. We assume that the decoder of this model suffers from the performance limitation. Although time-frequency features useful for onset detection are considerably different from those for offset detection, the single decoder is trained to output a mixed sequence of onset and offset events without guarantee of the correspondence between the onset and offset events of the same note. To overcome this limitation, we propose a streaming encoder-decoder model that uses a convolutional encoder aggregating local acoustic features, followed by an autoregressive transformer decoder detecting a variable number of onset events and another decoder detecting the offset events of the active pitches with validation of the sustain pedal at each time frame. Experiments using the MAESTRO dataset showed that the proposed streaming method performed comparably with or even better than the state-of-the-art offline methods while significantly reducing the computational cost.

1. INTRODUCTION

Automatic music transcription (AMT) is a central topic in the field of music information retrieval (MIR), which refers to converting a music recording into a symbolic musical score (MusicXML format) or a piano-roll representation (MIDI format) [1]. It has remarkably been improved with the technical progress of deep learning techniques and the public availability of large-scale music datasets. In this paper, we focus on streaming audio-to-MIDI AMT because it remains relatively unexplored unlike streaming automatic

Attribution: W. Wei, J. Zhao, Y. Wu, and K. Yoshii, "Streaming Piano Transcription Based on Consistent Onset and Offset Decoding with Sustain Pedal Detection", in *Proc. of the 25th Int. Society for Music Information Retrieval Conf.,* San Francisco, United States, 2024.

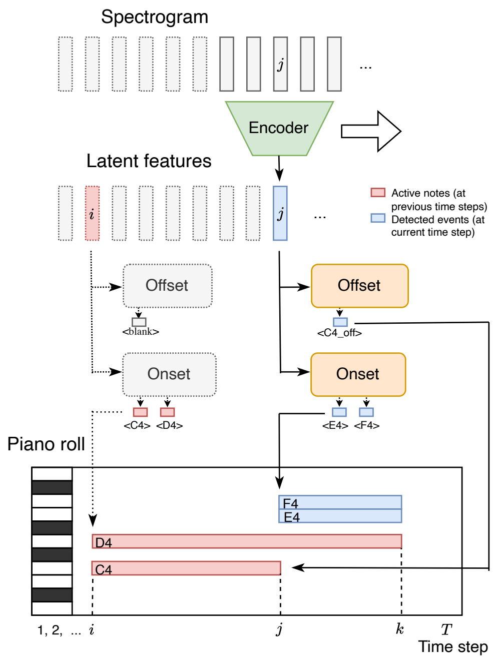

Figure 1. An overview of the proposed streaming audio-to-MIDI piano transcription method aware of onset-offset correspondence.

speech recognition (ASR) [2–4] and forms the basis of real-time music applications such as performance evaluation and interactive jam session. The previous research in [5] applied auto-regressive convolutional recurrent neural network (CRNN) frame-by-frame for piano transcription. The auto-regressive CRNN model can be easily adapted for the online scenario [6]. But the transcription performance for note offsets still has significant room for improvement.

Inspired by the sequence-to-sequence learning for ASR, many studies on AMT have recently attempted to use the Transformer [7] by serializing the polyphonic information of the estimation target [8, 9]. AMT is essentially different with ASR in a sense that the onsets, durations, and pitches of musical notes should be estimated, while the temporal information of output tokens (e.g., words and characters) is not considered in ASR. For audio-to-MIDI piano transcription, one may define the input and output of the Transformer as a sequence of raw audio features (e.g., mel and constant-Q spectrograms) and a sequence of note-on and note-off events sorted in time and pitch, respectively. The performance of this naive approach, however, is potentially limited. Despite the significant differences in features needed for detecting onsets and offsets, the Transformer decoder estimates these events in a mixed manner. In addition, the correspondence between the onset and offset events of the same note is not guaranteed.

For streaming AMT, one can use the *causal* Transformer that restricts the self-attentive region to a certain number of past frames, which could reduce the computational cost of the basic self-attention mechanism that increases quadratically with the input length. Nonetheless, due to the strong coupling between note events, Transformer-based transcription methods often underperform the state-of-the-art frame-level methods [10, 11], especially in offset detection and velocity estimation.

To overcome these limitations, we propose a streaming audio-to-MIDI piano transcription method based on a novel encoder-decoder architecture (Fig. 1). The encoder is implemented with a convolutional neural network (CNN) that sequentially aggregates latent features from local regions of an input piano recording. The two Transformer decoders that operate framewise are then separately used for detecting a variable number of onset events and offset events for the active pitches with guarantee of onset-offset correspondence. For further improvement, the offset decoder is trained to judge the activation of the sustain pedal in a way of multitask learning.

The main contribution of this study is to develop an efficient streaming encoder-decoder model and pave a way for interactive and responsive applications based on real-time music transcription. We experimentally show that our method performs comparably with a state-of-the-art offline transcription method and outperforms existing sequence-to-sequence transcription methods.

2. RELATED WORK

This section reviews related work on automatic music transcription and sequence-to-sequence transcription.

2.1 Automatic Piano Transcription

Automatic piano transcription (APT) is the most popular form of AMT. Early methods rely on handcrafted features and rule-based algorithms [12–15], while modern methods use deep learning models such as CNNs [16–19], recurrent neural networks (RNNs) [20, 21], and transformers [22, 23]. In APT, the framewise transcription has still been the mainstream approach due to its superior performance and accuracy [10, 24]. In this approach, audio features such as short-time Fourier transform (STFT) spectrograms are mapped to a binary matrix of dimensions $T \times N$ indicating the presence of pitches over time frames, where T represents the number of frames and N the number of pitches. Early transcription methods, mostly based on CNNs, perform comparably at the frame level but underperform in term of note-level.

Onsets and Frames [19] is a major breakthrough in APT that learns to sequentially predict note onsets and pitches in a multitask framework. To improve the performance, a music language model (MLM) based on a bidirectional long short-term memory (BiLSTM) network is used for modeling the temporal dependency of musical notes. This study has triggered many extensions. Kong et al. [25], for example, proposed a high-resolution piano transcription (HPT) model that simultaneously deals with onset, offset, velocity, and frame prediction tasks. The predicted velocities are used as conditional information to predict onsets, and the predicted onsets and offsets are used to predict frame-wise pitches, forming a hierarchical structure.

Our previous work [24] proposed HPPNet that uses harmonic dilated convolution for constant-Q transform (CQT) spectrograms and an enhanced frequency grouped LSTM (FG-LSTM) as a MLM. This model exhibits improved performance in both frame-level and note-level predictions. To capture long-term temporal and spectral dependencies, Toyama et al. [10] proposed a two-level hierarchical frequency-time transformer (hFT-Transformer) and achieved the state-of-the-art performance on the prediction of note with offset and velocity.

2.2 Sequence-to-Sequence Transcription

Sequence-to-sequence models are able to learn a mapping between input and output sequences of variable lengths and have actively been investigated in many fields such as natural language processing (NLP) and automatic speech recognition (ASR). Such models have recently been implemented with the Transformer or the self-attention mechanism due to its excellent performance. Awiszus et al. [26], for example, proposed a piano transcription model based on an LSTM and a Transformer for frame-level multi-pitch estimation. The performance of this method, however, is limited due to the lack of training data and using improper relative time shifts.

Inspired by this study, Hawthorne et al. [8] proposed a note-level piano transcription model that uses Transformer encoder and decoder in a way similar to the T5 model [27]. The encoder extracts latent features from an input spectrogram and the decoder refers to the input in an autoregressive manner, and the token with the highest probability is selected at each frame. This method achieved promising performance on the MAESTRO dataset and was later extended to multi-track music transcription [9]. However, this sequence-to-sequence transcription method still faces limitations. It encodes all types of note events and absolute time location of each event into a single sequence. This increases the complexity of sequence-to-sequence transformation and also constrains the length of the input sequence.

3. PROPOSED METHOD

This section explains the proposed method of streaming audio-to-MIDI piano transcription based on a single encoder and onset and offset decoders.

3.1 Streaming Transcription

As shown in Algorithm 1, the model takes a spectrogram $\mathbf{X} \in \mathbb{R}^{T \times F_i}$ as input, where T represents the number of frames and F_i represents the number of frequency bins. It outputs an onset sequence list $\mathbf{Y}$ and an offset sequence list $\overline{\mathbf{Y}}$, where each element $\mathbf{Y}_t$ in $\mathbf{Y}$ represents the detected onsets sequence of frame t with sequence length k_t, and each element $\overline{\mathbf{Y}}_t$ in $\overline{\mathbf{Y}}$ represents the detected offsets sequence with sequence length n_t in frame t.

The model consists of one encoder and two decoders (Fig. 2). The encoder is implemented with a CNN that efficiently extracts and aggregates local features from the audio spectrogram $\mathbf{X}$. The two separate decoders are then used at each frame for detecting a variable number of onset times and judging the offset of the detected notes by focusing on different aspects of the latent features.

More specifically, at each frame t, the encoder takes as input the audio spectrogram around frame t with a receptive field of a fixed size M and outputs a hidden embedding sequence $\mathbf{H}_t \in \mathbb{R}^{F_h \times D}$ in the frequency domain with a sequence length of F_h and the hidden embedding size of D. In addition, positional encodings are incorporated into the encoder hidden states $\mathbf{H}_t$. Then the decoders receive $\mathbf{H}_t$ with the cross attention (encoder-decoder attention).

For onset detection, the onset sequence $\mathbf{Y}_t$ at frame t is initialized with the beginning-of-sequence token (BOS). The onset events are then detected using the onset decoder Decoder$_{on}$ iteratively until the end-of-sequence token (EOS) is obtained, considering the current encoder hidden state $\mathbf{H}_t$, the onset sequences $\mathbf{Y}_{1:t-1}$ detected in previous times, the current onset sequence at frame t, and decoder positional encodings. The detected onset events are finally added to the active onsets set $\mathbf{A}$. The process is repeated throughout the input sequence $\mathbf{X}$.

The offset events are detected using the offset decoder Decoder$_{off}$, considering the current encoder hidden state $\mathbf{H}_t$, the active onsets set $\mathbf{A}$, and decoder positional encodings. Then active onsets corresponding to the detected offsets are removed from $\mathbf{A}$ indicating the end of notes. It should be emphasized that the offset decoder does not perform sequence prediction. Instead, it predicts the offset for each onset that has been activated in the past time steps all at once.

Algorithm 1 Streaming piano transcription. The length of output onset sequence equals to the number of the detected onsets, while the length of offset sequence has an additional output for pedal offset indexed as 0.

Input: Source sequence $\mathbf{X} = (x_1, x_2, ..., x_T)$
Output:
Onset sequence $\mathbf{Y} = (Y_1^{1:k_1}, Y_2^{1:k_2}, ..., Y_T^{1:k_T})$
Offset sequence $\overline{\mathbf{Y}} = (\overline{Y}_1^{0:n_1}, \overline{Y}_2^{0:n_2}, ..., \overline{Y}_T^{0:n_T})$
Parameters:
Receptive field of encoder: M
Initialize positional encodings: $\mathbf{PE}_{\text{enc}}$ and $\mathbf{PE}_{\text{dec}}$
Initialize active onsets set: $\mathbf{A} = \{\}$
for $t = 1$ **to** T **do**
    $H_t \leftarrow$ Encoder$(X_{t-\frac{M}{2}:t+\frac{M}{2}})$
    $H_t \leftarrow H_t + \mathbf{PE}_{\text{enc}}$
    // Offset decoder
    $n_t \leftarrow \mathbf{A}$.size()
    $\overline{Y}_t \leftarrow$ Decoder$_{off}(H_t, \mathbf{A}, \mathbf{PE}_{\text{dec}})$
    Delete onsets in $\mathbf{A}$ corresponding to offsets in $\overline{Y}_t$
    // Onset decoder
    $k_t \leftarrow 0$
    $Y_t^{k_t} \leftarrow$ BOS
    $y \leftarrow$ BOS
    while y != EOS **do**
        $y \leftarrow$ Decoder$_{on}(H_t, Y_{1:t-1}, Y_t^{0:k_t}, \mathbf{PE}_{\text{dec}})$
        if y == EOS **then**
            break
        end if
        $k_t \leftarrow k_t + 1$
        $Y_t^{k_t} \leftarrow y$
    end while
    $\mathbf{A}$.add(Y_t)
end for

3.2 Encoder

The encoder is based on the harmonic dilated convolution originally used for HPPNet [24] and uses the same configuration proposed for the acoustic model of HPPNet. It extracts local acoustic features with a fixed receptive field and feeds them to the decoders. There are three sets of convolutional layers with different kernel sizes: three layers with a kernel size of 7×7, one harmonic dilated convolution layer with a kernel size of 1×3, and five layers with a kernel size of 5×3. The resulting receptive field in the time dimension is $M = 39$.

For streaming piano transcription, we use the shifting window approach for sequentially feeding an input spectrogram to the encoder. Instead of feeding the entire spectrogram at once, we segment it into smaller chunks or windows to simulate real-time processing. These windows are shifted along the time axis, allowing the model to gradually analyze incoming audio data. We define the size of each window based on the desired temporal context for transcription. Typically, smaller window sizes facilitate faster processing but may sacrifice some contextual information, whereas larger window sizes provide more context but may introduce latency. To ensure continuity of transcription and avoid information loss at window boundaries, we apply overlap between consecutive windows.

3.3 Decoder

Both the onset and offset decoders are the same as the decoder of T5 [27] (Fig. 2). In the decoder architecture, the embedding size is set to $D_{dec} = 256$, and decoder layers to $L = 6$, attention head number to $N_{head} = 8$. The multilayer perceptron (MLP) dimension is set to $D_{mlp} = 1024$. A maximum decoder sequence length $N_{seq} = 64$. The length of the decoder output varies with the number of activated onsets. During the training phase, we use padding and masking to fix the output tokens length of offset decoder to 16.

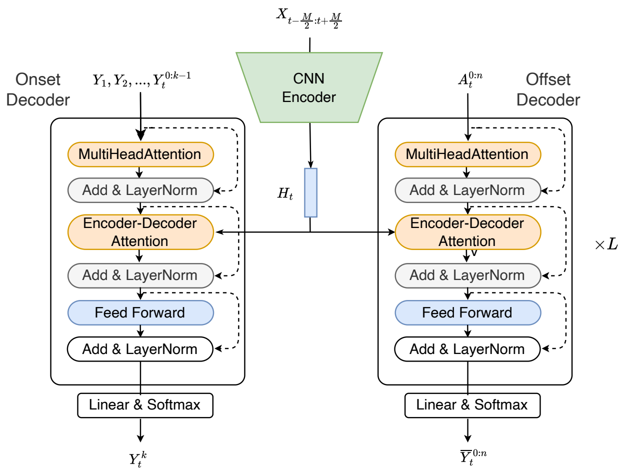

Figure 2. The implementation of the streaming transcription model that uses one encoder for latent feature extraction and two decoders for onset and offset detection.

3.4 Consistent Decoding

Existing piano transcription models that applied onset and offset detection [9, 25] often face issues with mismatched detected onsets and offsets. This is due to the little constrains in the detection processes for onsets and offsets. Although this issue can be addressed with post-processing methods, we prefer to solve it end-to-end within the model. Our proposed architecture makes a constriction to the offset decoder to detect offsets for detected onsets only, and also detects sustain pedal release events to improve performance of note offsets detection.

The onset decoder sequentially outputs onset events in an autoregressive manner while the offset decoder detects all the offset events at once for the active notes detected by the onset decoder with judgement of the sustain pedal. If the offset event for an active note is not detected at the current frame, a special token $BLANK$ is obtained as described in Section 4.1.3. The onset decoder considers only notes detected in the past and current frames. The sustain pedal plays a crucial role in expressive piano performance and considerably affects offset detection. The lifting time of the sustain pedal is highly relevant to the absolute offset times and thus determines the duration and decay characteristics of musical notes.

The input of the onset decoder in each step at frame t consists of the onset tokens detected in the previous step and the onset tokens detected at previous frames. This enables to capture long-term dependency between musical notes. By incorporating information from previous frames, the decoder can better understand the context of the current onset detection and facilitate the recognition of typical patterns and structures in the music sequence over time.

4. EVALUATION

This section reports a comparative experiment conducted for evaluating the performance of the proposed and conventional piano transcription methods.

Time	Target Tokens
1	<EOS> <blank>
2	<EOS> <blank>
...	
i	<C4><D4><EOS> <blank>
i+1	<C4><D4><EOS> <blank><blank><blank>
i+2	<EOS> <blank><blank><blank>
...	
j	<E4><F4><EOS> <blank><C4_off><blank>
...	
k	<EOS> <pedal_off><D4_off><E4_off><F4_off>

Table 1. Target tokens for onset decoder(red) and offset decoder(blue).

4.1 Experimental Conditions

We explain the dataset used for evaluation and the input and output data of the proposed method.

4.1.1 Dataset

We used the MAESTRO dataset V3.0.0 [28] composed of about 200 hours of virtuosic piano performances captured with fine alignment between note events and audio recordings. The split of the dataset into training, validation, and test sets was defined officially. The validation set was used for selecting the best-performing trained model based on its performance on unseen data. The dataset also provides information about the states (on or off) of the sustain pedal. The pedal information is crucial for accurately transcribing piano performances as it affects the *actual* durations and offset times of sustained notes.

4.1.2 Input

The original audio recordings were resampled with a sampling rate of 16 kHz. To increase the variation of the training data and reduce the memory footprint, 10-sec segments were randomly clipped from the recordings and the CQT spectrograms were computed on the fly with the nnAudio library [30]. We used the CQT for its capability of capturing both higher and lower-frequency components in the logarithmic frequency domain suitable for analyzing music signals. The lowest frequency was set to 27.5 Hz corresponding to the lowest key of the standard 88-key piano. One octave was divided into 48 frequency bins and the total number of frequency bins was 352. This ensures a fine frequency resolution over the entire audible frequency range. The hop length was set to 320 samples (20 ms), taking the balance between the time resolution and the computational efficiency. After obtaining the CQT spectrogram, the amplitude values were converted to decibels (dB) using transforms available in the torchaudio library.

4.1.3 Output

The vocabulary of output tokens used in our study was the same as that used for the music transformer 3 (MT3) [8, 9] except that time location tokens were not used. This contributes to reducing the length of the output sequence and stabilizing the training. The output vocabulary consists of the following tokens:

Model	Params	Frame-level			Note-level (onset only)			Note-level (onset + duration)		
		P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)
Onsets & Frames [28]	26M	92.11	88.41	90.15	98.27	92.61	95.32	82.95	78.24	80.50
Semi-CRFs [29]	9M	93.79	88.36	90.75	98.69	93.96	96.11	90.79	86.46	88.42
HPPNet-sp [24]	1.2M	92.79	93.59	<u>93.15</u>	98.45	95.95	<u>97.18</u>	84.88	82.76	83.80
hFT-Transformer [10]	5.5M	92.82	93.66	**93.24**	99.64	95.44	**97.44**	92.52	88.69	**90.53**
Streaming Seq2Seq (ours)	16M	91.91	91.73	91.75	98.30	94.83	96.52	91.08	87.89	<u>89.44</u>

Table 2. The transcription performances of the existing and proposed methods on MAESTRO V3.0.0 test set.

Model	Segment	Encoder Input Seq-Length	Decoder Output Seq-Length	Latency	Note F1	Note w/ Offset F1
Seq2Seq [8]	4088 ms	511	1024	4088 ms	96.01	83.94
Streaming Seq2Seq (ours)	-	39	64	380 ms	**96.52**	**89.44**

Table 3. The transcription performances of sequence-to-sequence transcription models on MAESTRO V3.0.0 test set.

Onsets and offsets (128+128 tokens) Each token represents the presence of an onset or offset of the corresponding pitch given as a MIDI note number.

Pedal states (2 tokens) Two tokens representing the presence and absence of the sustain pedal.

BLANK (1 token) A special token representing silence or absence of any musical event.

BOS and EOS (2 tokens) Special tokens representing the beginning and end of the output sequence.

The onset decoder and offset decoder both need only part of the vocabulary. But we kept the full vocabulary for all decoders to maintain consistency in the model architecture, regardless of whether there is only one decoder or multiple decoders. We set the length of each onset and offset events into 2 frames. During the transcription process, if consecutive onsets or offsets were detected, we only kept the first one and discard the duplicates. To estimate note events from the output of the decoders we used a simple greedy regression algorithm. We then selected the nearest corresponding offsets after the onsets to determine the duration of the notes. If offsets were not detected, we selected the nearest pedal offset as the offsets for the notes or a maximum duration of 4 seconds.

4.1.4 Training

We used the cross entropy loss for training the proposed model. It represents the negative log-posterior probability over output tokens for the ground-truth annotation. For optimization, we utilized the AdamW optimizer [31], which is a variant of the Adam optimizer with weight decay regularization. The mini-batch size was set to 16 and the learning rate was set to 6e-4. A dropout rate of 0.1 was applied to the decoder layers to prevent overfitting. Training was iterated for 200,000 steps with early stopping.

4.1.5 Metrics

The performance of piano transcription was evaluated with the mir_eval library [32] in terms of the precision and recall rates and F1 score at the frame and note levels. In the note-level evaluation, an estimated note was judged as correct if its onset time was detected correctly or if both the onset time and duration were estimated correctly. The error tolerance in onset estimation was set to 50 msec as in many studies. The error tolerance in duration estimation was set to the larger of 50 msec or 20% of the ground-truth duration. These metrics were averaged over the test set.

4.2 Experimental Results

We report the experimental results obtained through comparative and ablation studies.

4.2.1 Comparison with Existing Methods

We conducted a comprehensive experiment that compared our method with state-of-the-art methods such as frame-level and event-level transcription methods (Table 2). We found that our method achieved competitive performance and surpassed an event-level transcription method named Semi-CRFs in terms of both the note-level F1-scores with and without duration evaluation. This superiority indicates the robustness of our method in capturing the musical onset events and their corresponding offsets.

4.2.2 Sequence to Sequence Transcription

For comparison, we tested the generic transformer-based sequence-to-sequence transcription model [8] (Table 3). Audio recordings were split into segments of 4088 msec to be fed to the encoder. Since different segments were transcribed independently, the long-term correlation between note events is hard to learn from the data. Moreover, increasing the segment length would exponentially increase the computational complexity of the self-attention layers. It would increase the number of absolute-time-location tokens and further complicates the estimation of time locations for note events.

Thanks to the streaming encoder-decoder architecture, the proposed model kept the actual input length constant

Decoder	Onset	Offset	Pedal	Note-level (onset only)			Note-level (onset+duration)		
				P (%)	R (%)	F1(%)	P (%)	R (%)	F1(%)
1	✓	✓	✓	98.32	93.36	95.73	89.91	85.41	87.56
2	✓	✓		98.23	94.75	96.44	88.11	85.00	86.51
2	✓	✓	✓	98.30	94.83	**96.52**	91.08	87.89	**89.44**

Table 4. Ablation study on MAESTRO V3.0.0 test set.

and significantly reduced the computational complexity of the self-attention layers. The length of the encoder input was set to 39 and the maximum length of the decoder output was set to 64. This enables the processing of variable-length audio recordings without the need for segmentation and offers the potential for real-time transcription. Compared with the generic model, our streaming model showed better performance in terms of the note-level F1-scores with and without duration evaluation. This indicates the potential application to streaming and sequence-to-sequence music transcription scenarios.

4.2.3 *Latency*

The latency of a streaming model refers to the gap between the actual time of an onset or offset event and the time of the event output. Putting the actual computational speed aside, the latency of a non-streaming model is equal to the length of the input sequence because the whole sequence needs to be processed for generating outputs. In contrast, for streaming models, the latency is equal to the length of future frames in the input data stream.

In Table 3, our streaming model had a latency of 380 msec. The CNN-based encoder takes 19 future frames and 19 past frames as input. Even with a short input context, the streaming model still achieved competitive performance on piano transcription. This indicates that onset and offset events could be detected without heavily relying on long-term dependency of acoustic features.

4.2.4 *Ablation Study*

To verify the effectiveness of sustain pedal detection and that of the separated decoders for onset and offset detection, we conducted an ablation study. Besides the proposed model, we trained a model without pedal detection and another model that uses a single decoder for onset, offset, and pedal detection. The training and evaluation were performed in the same way.

Table 4 shows the performances of the compared methods. We found that removing the pedal detection slightly decreased the note-level F1-score without duration estimation, but significantly degraded the note-level F1-score with duration estimation. This suggests that pedal detection plays a crucial role in estimating note durations. Similarly, using a single decoder for both onset and offset detection degraded both the note-level F1-scores with and without duration estimation, compared with the proposed model. This demonstrated the effectiveness of incorporating pedal detection and a separated decoder for onset and offset prediction for better piano transcription.

5. CONCLUSION

In this paper, we have presented a novel streaming audio-to-MIDI piano transcription method. We tackled an open problem of detecting note onset and offset events from a piano recording in an online manner. Our method is based on a streaming encoder-decoder architecture that combines a convolutional encoder for aggregating local acoustic features with separate transformer decoders for detecting onset and offset events at each time step while validating the use of the sustain pedal.

In extensive experiments with the MAESTRO dataset, our method attained competitive performance, compared with the state-of-the-art offline methods. Our model also outperformed the generic transformer-based sequence-to-sequence model in terms of both accuracy and latency. The ablation study showed the effectiveness of incorporating pedal detection and that of using the separated decoders for onset and offset detection. Our method uses a limited number of incoming frames for detecting the onset and offset events and paved a way for latency-critical practical applications. We achieved a system latency of 380 msec and plan to thoroughly investigated the trade-off between the latency and the transcription performance. Additionally, decoding every frame may not be necessary. Some scenarios might not require such high temporal precision. The setting of the time step also requires further exploration for real-time scenarios.

6. ACKNOWLEDGMENTS

This work was partially supported by JST SPRING No. JPMJSP2110, JST FOREST No. JPMJFR2270, and JSPS KAKENHI Nos. 24H00742 and 24H00748.

7. REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2018.

[2] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6645–6649.

[3] C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[4] X. Chen, Y. Wu, Z. Wang, S. Liu, and J. Li, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 5904–5908.

[5] T. Kwon, D. Jeong, and J. Nam, "Polyphonic piano transcription using autoregressive multi-state note model," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, J. Cumming, J. H. Lee, B. McFee, M. Schedl, J. Devaney, C. McKay, E. Zangerle, and T. de Reuse, Eds., 2020, pp. 454–461.

[6] D. Jeong and S. Telecom, "Real-time automatic piano music transcription system," *Late Breaking/Demo of the International Society for Music Information Retrieval Conference (ISMIR)*, pp. 4–6, 2020.

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

[8] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. H. Engel, "Sequence-to-sequence piano transcription with transformers," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021, pp. 246–253.

[9] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. H. Engel, "MT3: multi-task multitrack music transcription," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2022.

[10] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W. Liao, and Y. Mitsufuji, "Automatic piano transcription with hierarchical frequency-time transformer," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023, pp. 215–222.

[11] W. T. Lu, J. Wang, and Y. Hung, "Multitrack music transcription with a time-frequency perceiver," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[12] A. Khlif and V. Sethu, "An iterative multi range non-negative matrix factorization algorithm for polyphonic music transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 330–335.

[13] S. A. Abdallah and M. D. Plumbley, "Unsupervised analysis of polyphonic music by sparse coding," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 179–196, 2006.

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

[15] J. Nam, J. Ngiam, H. Lee, and M. Slaney, "A classification-based polyphonic piano transcription approach using learned feature representations," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pp. 175–180.

[16] R. Kelz, M. Dorfer, F. Korzeniowski, S. Böck, A. Arzt, and G. Widmer, "On the potential of simple framewise approaches to piano transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2016, pp. 475–481.

[17] R. Kelz, S. Böck, and G. Widmer, "Deep polyphonic ADSR piano note transcription," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 246–250.

[18] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE ACM Transactions on Audio Speech and Language Processing (TASLP)*, vol. 24, no. 5, pp. 927–939, 2016.

[19] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018, pp. 50–57.

[20] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 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 *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

[22] L. Ou, Z. Guo, E. Benetos, J. Han, and Y. Wang, "Exploring transformer's potential on automatic piano transcription," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 776–780.

[23] R. Wu, X. Wang, Y. Li, W. Xu, and W. Cheng, "Piano transcription with harmonic attention," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 1256–1260.

[24] W. Wei, P. Li, Y. Yu, and W. Li, "HPPNet: Modeling the harmonic structure and pitch invariance in piano transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 709–716.

[25] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE ACM Transactions*

on Audio Speech and Language Processing (TASLP), vol. 29, pp. 3707–3717, 2021.

[26] B. S. M. Awiszus, "Automatic music transcription using sequence to sequence learning," Ph.D. dissertation, Master's thesis, Karlsruhe Institute of Technology, 2019.

[27] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, pp. 140:1–140:67, 2020.

[28] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019, pp. 1–6.

[29] Y. Yan, F. Cwitkowitz, and Z. Duan, "Skipping the frame-level: Event-based piano transcription with neural Semi-CRFs," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 34, 2021, pp. 20 583–20 595.

[30] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, "nnAudio: An on-the-fly GPU audio to spectrogram conversion toolbox using 1d convolutional neural networks," *IEEE Access*, vol. 8, pp. 161 981–162 003, 2020.

[31] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[32] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "Mir_eval: A transparent implementation of common mir metrics," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 1–6.