IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 June 2024, accepted 5 July 2024, date of publication 8 July 2024, date of current version 16 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3425167

RESEARCH ARTICLE

Polyphonic Piano Music Transcription System Exploiting Mutual Correlations of Different Musical Note States

TAEHYEON KIM[1], DONGHYEON LEE[1], MAN-JE KIM[2], (Member, IEEE), AND CHANG WOOK AHN[1], (Member, IEEE)

[1]AI Graduate School, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea
[2]Convergence of AI, Chonnam National University, Gwangju 61186, South Korea

Corresponding authors: Man-Je Kim (jaykim0104@jnu.ac.kr) and Chang Wook Ahn (cwan@gist.ac.kr)

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT, Ministry of Science and ICT) (RS-2024-00347902) and the Ministry of Education (RS-2023-00247900), and the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01842, Artificial Intelligence Graduate School Program (GIST, Gwangju Institute of Science and Technology) & the Artificial Intelligence Convergence Innovation Human Resources Development (RS-2023-00256629)).

ABSTRACT Generally, polyphonic piano music transcription systems are designed to estimate and determine pitch activities along with various note states for each audio frame. While the music transcription system has multiple uses in the Music Information Retrieval (MIR) field, due to the complicated structures of the note events, precisely predicting various note states is still regarded as a challenging task. Accordingly, approaches to designing neural network architectures have evolved to facilitate the joint prediction of each note state. However, recent models have not been able to efficiently exploit mutual correlations among different note states. The key contribution of our work is that we verified mutual correlations between the different note states and reflected them in the model architecture. It enables the transcription system to recognize clearer note events and produce high-quality real-world results. We propose a kernel-sharing feature extractor module for exploiting those mutual correlations in the feature extraction step. Moreover, to make a system recognize the shape of the pitch envelope, we added some connections between the note state-specific detector modules in the note state detection step. The efficacy of our architecture was thoroughly validated in a series of experiments using the publicly available MAESTRO datasets proposed by Google Magenta. Furthermore, ablation studies are performed to demonstrate notions of those mutual correlations and show the impact and significance of the suggested approach.

INDEX TERMS Polyphonic piano music transcription, joint estimation system, mutual correlations, musical note states.

I. INTRODUCTION

Bridging the gap between audio-based and symbolic music representations is essential for advancing computational music information processing. Symbolic music representations, such as MIDI format and piano-roll representation, have great practical utility in various symbolic Music Information Retrieval (MIR) tasks. For instance, symbolic music composition [1], [2], [3], [4] using language models has been the subject of much attention. Further, in order to perform Artist/Genre Classification [5], [6], [7], symbolic music features related to the artist's style and genre are used to train classification models. These capabilities highlight the importance of an Automatic Music Transcription (AMT) system [8] to detect musical note events in music audio samples. Polyphonic Piano Music Transcription is particularly challenging and critical in AMT challenges due to the capability of the piano to produce a broad spectrum of notes, multi-layered melodies, and intricate harmonies simultaneously. Therefore, through the deep analysis of piano

The associate editor coordinating the review of this manuscript and approving it for publication was Tai Fei.

music cases, it is possible to advance the development of more sophisticated music transcription algorithms.

To accurately capture the overlapping and time-varying musical notes in polyphonic piano music, Deep Neural Network (DNN) architectures have evolved to identify various note states individually and capture the timing of note states more precisely. The notable breakthrough and performance improvement in this field occurred with the Onsets and Frames transcription [9], designed to predict onsets and frame-wise pitches jointly. Based on the Onsets and Frames model architecture, there have been subsequent approaches [10], [11], [12], [13] to developing parallel Convolutional Recurrent Neural Network (CRNN) structures aimed at the joint estimation of the note states up to recently.

Though the earlier architectures showed impressive performance, further enhancements in practical results necessitate an in-depth exploration of the roles and effects of each distinct note state within the transcription process. This study focuses on the design method of neural architecture to improve the joint estimation of note states. By analyzing the shape of the ADSR envelope, we estimate the mutual correlations between different note states and analyze their impact on performance. Then, we propose a polyphonic piano music transcription system that can generate improved results with more clear note events. The main contributions of this work can be highlighted as follows:

- To the best of our knowledge, this is the first known work that attempts to verify the mutual correlations between different note states in music transcription tasks via modification of the DNN architecture. We verified the effective mutual correlation within the combination of the onset, offset, velocity, and frame-wise pitches. We implemented comparative experiments with a kernel-sharing feature extractor module to achieve that.
- Technically, we augmented an additional frame-wise pitch detector module on four parallel note state-specific detector modules. Namely, we leverage a total of five note state-specific detector modules. Then, we added the connection from the onset and offset detector to the second frame-wise pitch detector so that the entire model can infer the pitch envelope's approximate shape using the onset and offset timing.
- We multilaterally analyzed our approaches with several comprehensive ablation experiments and several metric values of results. Further, by comparing the visual piano-roll results, the practical effects in real-world applications were directly shown, compared, and analyzed. These results demonstrate that exploiting mutual correlations and timing information is effective in enhancing the output quality of the previous.

The remainder of this paper is arranged as follows: In Section II, we provide the background information for the previous research. In Section III, we introduce our music transcription system that exploits mutual correlations of onsets, offsets, and velocities. Section IV explains the experimental method. Section V provides comparative experimental results and a practical analysis of inference results for test samples. Lastly, Chapter 6 provides our comments on the overall results of our method.

II. RELATED WORKS

A. MULTI-LABELED NOTE STATES CLASSIFICATION SYSTEM

The music transcription system aims to estimate concurrent pitches in each frame so that it transcribes the input spectrogram into an output piano-roll representation. To achieve this goal, in the Multi-Labeled Note States Classification (MLNSC) system, the outputs are the presence probabilities of pitches for the given log-mel spectrogram, denoted as $X \in \mathbb{R}^{d_{time} \times d_{freq}}$, where d_{time} is the number of d_{freq} of each audio clip and d_{freq} is the number of frequency bins. Then, these calculated probabilities are compared with the ground-truth frame-wise piano roll $I_{frame} \in \{0, 1\}^{d_{time} \times d_{pitch}}$, where d_{pitch} is the number of pitch classes and $I_{frame}(t, p)$ is equal to 1 if pitch p is active in frame t and is 0 otherwise.

As DNNs have become actively used for handling piano transcription tasks, they are commonly employed to model functions mapping the log-mel spectrogram to the frame-wise roll. Namely, DNNs are trained to predict the frame-wise presence probability of notes. Along with this, binary cross entropy loss is calculated on $I_{frame}(t, p) \in \{0, 1\}$ and $P_{frame}(t, p) \in [0, 1]$, where $P_{frame}(t, p)$ is the probability output by the DNNs at frame t and pitch class p. For the joint estimation of different note states, the same method is applied for calculating the loss values for the prediction of note onsets and offsets. However, for velocity prediction, the loss calculation incorporates an extra criterion based on the presence of an onset.

B. HIGH-RESOLUTION TIME REGRESSION SYSTEM

The goal of modeling the MLNSC system is to accurately represent the presence or absence of a note event for each frame. Therefore, traditional approaches used a discrete binary representation to denote $I_{frame}(t, p)$, indicating whether the note was activated. However, because of the hop size in the sampling process, discrete representation can be imprecise to express the exact timing of onset or offset on the discrete frame time axis.

To address this issue, the High-Resolution Time Regression (HRTR) System [12] was proposed with an algorithm for determining the precise continuous onset and offset times of each note. Instead of classifying the presence probabilities in discrete time for each frame, they regressed the time distance from its nearest onset or offset timing for each frame. Accordingly, they encoded the time distance Δ_i by a function d:

$$d(\Delta_i) = \begin{cases} 1 - \dfrac{|\Delta_i|}{J}, & |\Delta_i| \leq J \\ 0, & |\Delta_i| > J \end{cases} \tag{1}$$

where J is the sharpness of the target and also means the target range of the interval on the frames. Then, using this distance encoding, one recomputes the target encoding $I(t, p)$ as $D(t, p)$ for the onset and offset target.

C. PREVIOUS APPROACHES

Initial research in piano transcription focused on the application of discriminative models like support vector machines [14], which determined whether notes were present or absent within frame times. Additionally, to tackle the estimation of multiple pitches, a probabilistic approach based on spectral smoothness [15] was introduced. After that, an integrated approach utilizing both frequency and time domain analyses was suggested [16]. They assumed that musical signals could be represented as a linear combination of waveforms from individual piano notes. Thereafter, non-negative matrix factorizations (NMFs) were actively adopted to decompose the given spectrogram into note events [17], [18].

Next, deep learning (DL) [19], [20], [21], [22] approaches were consistently exploited. Lately, with the introduction of the MAESTRO dataset [23], deep neural networks (DNNs) have been able to learn from a large-scale dataset. Hence, the transcription performance of DNNs has been significantly improved. Accordingly, DNNs such as convolutional neural networks (CNNs) [24] and recurrent neural networks (RNNs) [25] have been actively used to handle the AMT problem [26], [27], [28], [29], [30], [31], [32] in the MLNSC system.

The aforementioned Onsets and Frames [9] architecture significantly reduced false positive values in note event detection by ensuring that note pitches don't generate without the presence of an onset. Then, in the follow-up study [23], the Onset and Frames model was expanded so that it could predict four different note states: onsets, offsets, velocities, and frame-wise pitches. In addition, there have been attempts to apply the adversarial training scheme to train the model [10], as well as to apply an additive attention mechanism to the model [11]. However, MLNSC systems had limited resolution due to the hop size of the frames. In the aforementioned way, the HRTR system [12] overcame the limitation of the transcription resolution problem. Additionally, HPT-T [13] is the model that simply changed RNNs to Transformer for velocity prediction in the existing architecture. As described above, parallel CRNN architectures inspired by Onsets and Frames models have been actively studied up until recently. In this work, we exploited the Onsets and Frames model and its follow-up CRNN-based models to validate our model.

III. PROPOSED METHOD

A. MOTIVATION

The parallel CRNN architectures have proven to be a powerful approach for frame-wise polyphonic piano transcription. Despite their remarkable performance, these approaches only connect RNNs and do not take into account the integration of CNNs. Due to their structural problems, those approaches have limitations in extracting local features related to the mutual correlations of different note states. However, within a single piano note event, there might be crucial mutual correlations between different note states.

For a piano note event, the specific timings of onset and offset are shown in Figure 1. The note onset timing refers to the point where a note begins, and the note offset timing refers to the point where the note ends and the amplitude starts to decrease. Namely, in the ADSR envelope, the starting points of the Attack phase and Release phase are the exact

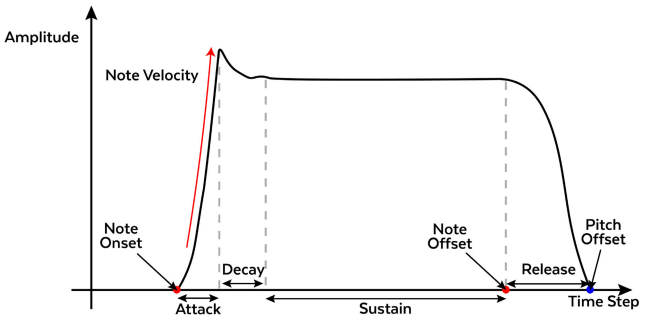

FIGURE 1. An ADSR (A: Attack, D: Decay, S: Sustain, R: Release) envelope, which describes how amplitude changes approximately over time for a piano note event [33].

Algorithm 1 Presence Probabilities Prediction Process of Note States in Proposed Architecture

Input: Log Mel Spectrogram $X \in \mathbb{R}^{d_{time} \times d_{freq}}$
Output: Presence Probabilities $P_{frame}, P_{velocity}, P_{onset}, P_{offset}$

▷ **Feature Extraction Step**
$F_{frame} \leftarrow$ Basic Feature Extractor(X)
$F_{combined} \leftarrow$ Kernel-Sharing Feature Extractor(X)
$F_{onset}, F_{offset}, F_{velocity} \leftarrow$ Split Feature Map $F_{combined}$
$\mathcal{F} \leftarrow \{F_{onset}, F_{offset}, F_{velocity}, F_{frame}\}$

▷ **Note State Detection Step**
for each $F \in \mathcal{F}$ **do**
  if $F = F_{frame}$ **then**
    $H_{frame} \leftarrow$ 1st Frame-wise Pitch Detector(F)
  else if $F = F_{onset}$ **then**
    $P_{onset} \leftarrow$ Onset Detector(F)
  else if $F = F_{offset}$ **then**
    $P_{offset} \leftarrow$ Offset Detector(F)
  else
    $P_{velocity} \leftarrow$ Velocity Detector(F)
  end if
end for

$C_{frame} \leftarrow$ Concat($[H_{frame}, P_{onset}, P_{offset}]$)
$P_{frame} \leftarrow$ 2nd Frame-wise Pitch Detector(C_{frame})

return $P_{frame}, P_{velocity}, P_{onset}, P_{offset}$

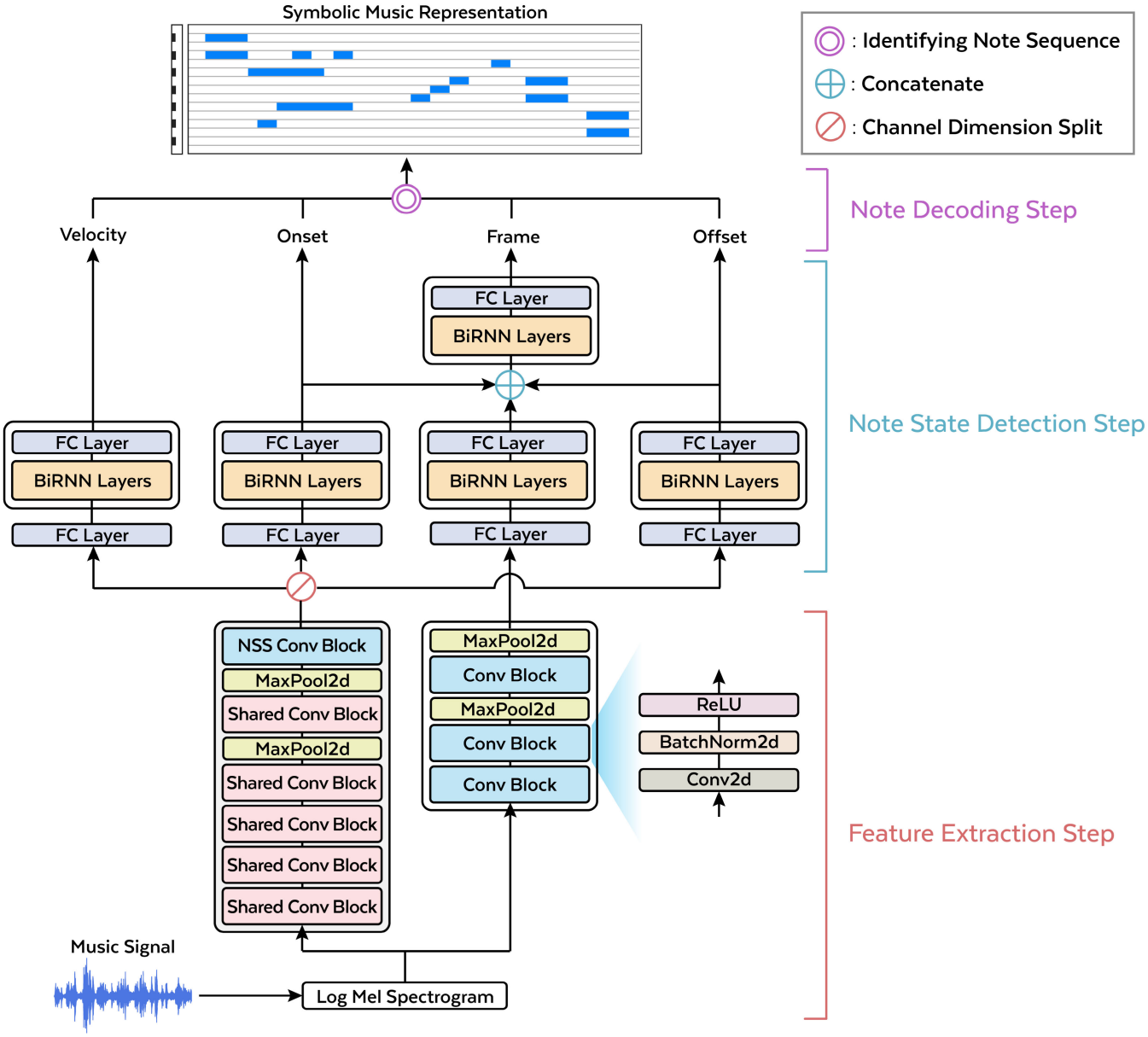

FIGURE 2. Polyphonic piano music transcription system considering the mutual correlation of three different note states. (onset, offset, velocity).

note onset and note offset timing, respectively. Therefore, within the ADSR envelope, knowing any two factors among the onset, offset, and duration of a note enables us to infer the other one approximately. Furthermore, from the perspective of expressing musical context, the note duration and the note velocity have a significant correlation. For instance, the soft and emotional parts of music are generally characterized by long note duration and low velocity. On the contrary, the staccato part, which arouses light-hearted emotions, is represented with high velocity and short note durations to emit a strong accent. Thus, based on these insights, we anticipate that there is a certain relationship between the onset, offset, and velocity of the notes in the primary transcription system. Accordingly, we propose a CRNN architecture optimized for exploiting mutual correlations among the onset, offset, and velocity of the notes.

B. ENTIRE ARCHITECTURE

Based on the above-mentioned motivation, we reflected our assumption in our music transcription system. The entire prediction process of different note states in our proposed architecture is described in Algorithm 1. Firstly, in the feature extraction step, we utilize the basic feature extractor module to extract feature maps only related to frame-wise pitch prediction. Whereas, for the other note states, we apply kernel-sharing feature extractor module and split along the channel dimension. Afterward, before the output feature maps progress to the note state detection step, to adjust the

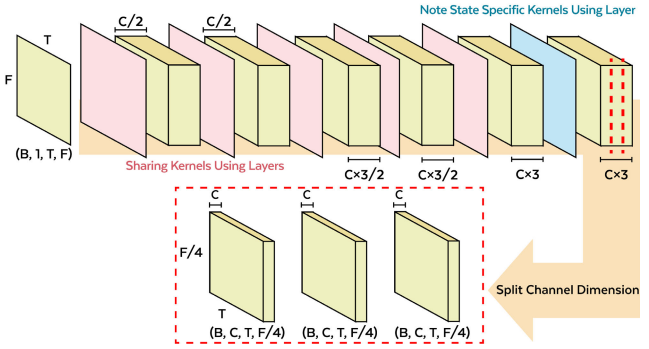

FIGURE 3. Kernel-sharing feature extractor module for sharing 3 different note states. (B: d_{batch}, C: $d_{channel}$, T: d_{time}, F: d_{freq}).

dimension of the feature maps, they are flattened across the frequency and channel axes and then passed to a fully connected layer. After passing the fully connected layer, it is fed into the note state-specific detector modules consisting of RNNs, a fully connected layer, and a sigmoid layer, which yield $d_{pitch} = 88$ dimension outputs accordant with the number of distinct pitches in MIDI format for acoustic piano. Especially, predictions of onset and offset are leveraged to predict frame-wise pitches with the second frame-wise pitch detector module.

C. BASIC FEATURE EXTRACTOR MODULE

For extracting individual feature maps corresponding to each note state, we use the base feature extraction module. As outlined in Algorithm 1, this module is employed specifically for frame-wise pitch detection. In each convolutional block, sequential implementation of a 2-D convolution, 2-D batch normalization [34], and ReLU activation [35] occurs. Accordingly, given a series of inputs $X \in \mathbb{R}^{d_{time} \times d_{freq}}$, the first convolutional block is computed as:

$$X_{frame} = ReLU(BatchNorm2d(Conv2d(X))) \quad (2)$$
$$= ConvBlock(X) \quad (3)$$

and the last two convolutional blocks are followed by a max-pooling layer [36] as follows:

$$F_{frame} = MaxPool2d(ConvBlock(X)) \quad (4)$$

To detect features more frame-wise and frequency-wise precisely, we introduced (3, 3) dimension kernels in convolutional layers. Here, the pooling layer shrinks the feature map's spatial size while preserving crucial information. Therefore, the max pooling layer is introduced instead of the average pooling layer to emphasize the prominent note states-related features by taking the maximum in the local area.

D. KERNEL-SHARING FEATURE EXTRACTOR MODULE

Meanwhile, we introduce a kernel-sharing feature extractor module to extract feature maps jointly related to different note states. These feature maps can provide predictive information useful across multiple contexts such as intrinsic correlations among different note states. Namely, shared kernels integrate information across different note states, allowing the model to learn more complex musical patterns effectively. The two structural differences from the basic feature extractor module are as follows: (i) channel dimension transition and (ii) location of two max pooling layers. The input representation for the feature extractor has the dimension of (d_{batch}, 1, d_{time}, d_{freq}), where d_{batch} is the batch size, 1 is the channel size, d_{time} is the number of frames, and d_{freq} is the number of frequency bins. Channel dimensions are progressively increased through the convolutional blocks, as detailed in Figure 3. Consequently, while the original architecture outputs a feature map shaped (d_{batch}, $d_{channel}$, d_{time}, $d_{freq}/4$), the proposed CNN architecture yields a feature map shaped (d_{batch}, $d_{channel} \times N$, d_{time}, $d_{freq}/4$), where $d_{channel}$ is the channel dimension of the final output feature map and $N(= 3)$ represents the number of note states under consideration (onset, offset, and velocity). During the process, shared kernels are used to generate a feature map that is jointly related to different note states. Following this layer, the output feature maps denoted as $F_{combined} \in \mathbb{R}^{d_{batch} \times (d_{channel} \times N) \times d_{time} \times d_{freq}/4}$ are then equally divided along the channel dimension into N separate feature maps, $F_{onset}, F_{offset}, F_{velocity} \in \mathbb{R}^{d_{batch} \times d_{channel} \times d_{time} \times d_{freq}/4}$. Namely, we distribute the kernel set in the last convolutional layer into N subsets ($\Gamma^{[i]}$, where $|\Gamma^{[i]}| = d_{channel} \times d_{channel}$ and $i \in \{0, \ldots, N-1\}$) so that the kernels of each subset $\Gamma^{[i]}$ can be allocated into each feature map, $F_{onset}, F_{offset}, F_{velocity}$. Accordingly, it was expected that $\Gamma^{[i]}$ would serve as the note state-specific kernels. Meanwhile, according to the increase in channel dimension, the parameter size and the computational complexity also increase. Therefore, we adjusted the position of the max pooling layer to reduce the whole model's complexity.

E. NOTE STATE-SPECIFIC DETECTOR MODULE

After the feature extraction step, we use bidirectional RNN layers to detect each specific note state. As shown in Figure 2, four note state-specific detector modules (first frame-wise pitch detector, onset detector, offset detector, and velocity detector) are designed in parallel, and one additional frame-wise pitch detector is augmented. For given four feature maps $F \in \mathbb{R}^{d_{batch} \times d_{channel} \times d_{time} \times d_{freq}/4}$ for each note state, each detector module is computed as follows:

$$P = FCLayer(BiRNNLayers(FCLayer(F)) \quad (5)$$
$$= Detector(FCLayer(F)), \quad (6)$$

where $P \in \mathbb{R}^{d_{batch} \times d_{time} \times d_{pitch}}$ which is predicted outputs for each note state. Predicted outputs from the onset detector, offset detector, and velocity are passed to the sigmoid layer so that the range of its value can be within (0, 1) and directly compared with ground truth. Whereas, the predicted output from the first frame-wise pitch detector should be passed to the second frame-wise pitch detector.

F. EXPLOITATION OF TIMING FEATURES FOR PITCH ENVELOPE APPROXIMATION

When we implement time-series analysis with the second frame-wise pitch detector, we exploit the timing features from the onset detector and offset detector. In order to achieve that, we concatenate outputs from three detectors: the first frame-wise pitch detector, the onset detector, and the offset detector, in the feature dimension. Therefore, for the concatenated feature map input $I_{frame} \in \mathbb{R}^{d_{batch} \times d_{time} \times (d_{pitch} \times 3)}$, the second frame-wise pitch detector is implemented as follows:

$$C_{frame} = Concat([H_{frame}, P_{onset}, P_{offset}]) \tag{7}$$

$$P_{frame} = Detector(C_{frame}) \tag{8}$$

The reason we exploit the timing features for pitch detection is to give the model approximate information about the shape of the pitch envelope. The pitch isn't generated without the presence of an onset and starts to decrease after note offset presence. Therefore, exploiting two timing features can be useful to predict frame-wise pitches more precisely.

G. PROBLEM FORMULATION WITH HIGH-RESOLUTION TIME REGRESSION LOSS

For jointly learning different note states, we adopted the summing of the losses of each note for its total loss function as follows:

$$l_{note} = l_{frame} + l_{velocity} + l_{onset} + l_{offset} \tag{9}$$

For the loss value of the frame-wise pitches, we calculate as:

$$l_{frame} = \sum_{t=1}^{d_{time}} \sum_{p=1}^{d_{pitch}} l_{bce}(y_{frame}(t,p), \hat{y}_{frame}(t,p)) \tag{10}$$

where the predicted probability output $\hat{y}_{frame}(t,p) = P_{frame}(t,p) \in [0,1]$ and ground truth $y_{frame}(t,p) = I_{frame}(t,p) \in \{0,1\}$.

Then, for the velocity, we calculate the loss value as follows:

$$l_{velocity} = \sum_{t=1}^{d_{time}} \sum_{p=1}^{d_{pitch}} I_{onset}(t,p) l_{bce}(y_{velocity}(t,p), \hat{y}_{velocity}(t,p)) \tag{11}$$

where the predicted probability output $\hat{y}_{velocity}(t,p) = P_{velocity}(t,p) \in [0,1]$ and ground truth $y_{velocity}(t,p) = I_{velocity}(t,p) \in [0,1]$.

Especially, using $d(\Delta_i)$ for onset and offset prediction, $D \in [0,1]^{T \times F}$ is defined as onset and offset regression targets at frame t and pitch class p. Accordingly, the regression loss for onset and offset is defined as follows:

$$l_{onset} = \sum_{t=1}^{d_{time}} \sum_{p=1}^{d_{pitch}} l_{bce}(y_{onset}(t,p), \hat{y}_{onset}(t,p)) \tag{12}$$

$$l_{offset} = \sum_{t=1}^{d_{time}} \sum_{p=1}^{d_{pitch}} l_{bce}(y_{offset}(t,p), \hat{y}_{offset}(t,p)) \tag{13}$$

where the predicted probability outputs $\hat{y}_{onset}(t,p) = P_{onset}(t,p) \in [0,1]$, $\hat{y}_{offset}(t,p) = P_{offset}(t,p) \in [0,1]$ and ground truths $y_{onset}(t,p) = D_{onset}(t,p) \in \{0,1\}$, $y_{offset}(t,p) = D_{offset}(t,p) \in \{0,1\}$.

Each loss value (l_{frame}, l_{onset}, l_{offset} and $l_{velocity}$) is calculated in different ways with the binary cross-entropy loss functions l_{bce}. In terms of velocity prediction, the term I_{onset} implies that velocities are predicted specifically at the moments when each onset occurs. In real performance, the velocity of a note is closely related to the intensity when the player strikes the keys. Therefore, the onset of a note (when the key is first struck) contains significantly more information about the velocity compared to other timings of the note. Therefore, to concentrate on this crucial information, ground truth onsets I_{onset} are employed to modulate the velocity prediction. Next, by using regression targets on onset and offset prediction, it becomes feasible to obtain continuous-time target data $y(t,p)$. Therefore, this approach enables the precise prediction of onset and offset timings, which is directly linked to more accurate capturing of the dynamic changes in musical pieces, thereby significantly refining the quality of the transcription process.

H. INFERENCE METHOD

As shown in Figure 2, we convert the music signals into log-mel spectrograms and input them into our proposed architecture. Then, via the Feature Extraction Note State Detection steps, the whole architecture calculates each note state-specific regression output. Those outputs are processed into a high-resolution note event sequence in the Note Decoding Step. Firstly, adopting the decoding algorithm from previous work [12], local maximum prediction values with adjacent two values of onsets and offsets are used to predict the precise timing through geometric approaches. Then, using the timing information of onsets and offsets, the activations of frame-wise pitches are identified. Based on the thresholds θ_{on} and θ_{off}, for the frame detected to be onset or offset, we decide it is activated if the pitch value is over the threshold. For the offset prediction, there is an additional condition for frame-wise pitch threshold θ_{frame}. Thus, we regard the frame as an offset if the pitch value is lower than θ_{frame}. After that, to ensure all onsets and offsets are paired, we insert offsets within a sequence of consecutive onsets to segment them. Furthermore, the velocity value is rescaled to $P_{onset}(t,p) \in [0,127]$ from $P_{onset}(t,p) \in [0,1]$ in accordance with the range of velocity in MIDI format.

IV. EXPERIMENTS

A. DATASET

To train and validate our architecture, we used the MAESTRO [23] v3 dataset. It contains 198.7 hours of MIDI-synchronized solo piano recordings captured with a time resolution of less than 3 ms. The recordings were recorded by performing the Yamaha Disklaviers pianos equipped with an integrated high-precision MIDI capture

TABLE 1. Transcription results for comparison of the proposed architecture with baselines in this work. (* are the results represented in the original papers).

		Frame			Note			Note w/ Offset			Note w/ Offset & Vel		
Model Architecture	# Parameter	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)
Onsets and Frames ([23])*	23.5 M	92.11	88.41	90.15	98.27	92.61	95.32	82.95	78.24	80.5	79.89	75.37	77.54
Adversarial ([10])*	26.9 M	93.10	89.80	**91.40**	98.10	93.20	95.60	83.50	79.30	81.30	82.30	78.20	80.20
High-Resolution ([12])*	34.1 M	88.71	90.73	89.62	98.17	95.35	96.72	83.68	81.32	82.47	82.10	79.80	80.92
HPT-T ([13])*	41.4 M	-	-	90.09	97.88	95.72	96.77	84.13	82.31	**83.20**	82.85	81.07	**81.90**
Ours	54.9 M	89.05	90.43	89.66	98.70	95.84	**97.23**	84.09	81.71	82.87	82.62	80.29	81.42

TABLE 2. Changes of channel dimension of the feature maps in the convolutional architecture. (MLNSC: Multi-Labeled Note States Classification, HRTR: High-Resolution Time Regression).

	MLNSC System($d_{channel}$=96)		HRTR System($d_{channel}$=128)	
	original	proposed	original	proposed
input	1	1	1	1
1st	48	48	64	64
2nd	48	48	64	64
3rd	96	48×N	128	64×N
4th	-	48×N	-	64×N
5th	-	96×N	-	128×N
6th	-	96×N	-	128×N
split	-	(96, ..., 96)	-	(128, ..., 128)
output	**96**	**(96, ..., 96)**	**128**	**(128, ..., 128)**

and playback system. This setup ensures that the MIDI data is precise and reflective of the true performance dynamics. Thus, using the MAESTRO dataset can provide high-quality inputs that are vital for learning robust and generalizable features.

We divided the dataset into train/validation/test splits according to the provided configuration. Each split has the number of samples as 962/137/177 and the total duration as 159.2/19.4/20.0 hours. Most recordings are sampled at 44.1 kHz, but there are some exceptions where they are sampled at 48 kHz.

B. PREPROCESSING

As input to the proposed model, using the MAESTRO dataset, we computed the log-mel spectrograms. Since the stereo audio recordings have two channels, we downmixed them into one channel. Then, we downsampled them to 16 kHz for consistency of input data. We did a Short-Time Fourier Transform (STFT) using the Hann window with a 2048 window size and constant padding mode. Finally, to get the log-mel spectrogram, we applied 229 mel banks and a log scale.

C. EXPERIMENT CONFIGURATION

For training our model, we set a batch size of 8, a learning rate of 6e-4 with the Adam optimizer [37], and a clip gradient norm of 3. The learning rate decreases by 0.2 in every 10k iterations. Then, the training was implemented on four RTX 2080 GPU cards. Additionally, for the supplement experiments with various changes to our model, we trained and validated each model in the MLNSC system. The main and supplement experiments are implemented with the same hyperparameters and obtain average scores of 3 repeated trials for reliable results. The models converged after 300k and 500k iterations, and each training took around 2.5 days and 7.5 days for the MLNSC system and HRTR system, respectively.

As shown in Table 2, due to the different conventional output channel dimensions across baseline models of each different transcription system, we used different channel dimension settings in each transcription system. Following previous studies, we used channel dimensions 96 and 128 for the MLNSC system and HRTR system, respectively.

For the decoding step, we also use different threshold values in each transcription system. For the MLNSC system, we set all the thresholds to 0.5. On the other hand, for the HRTR system, we set θ_{on}, θ_{off}, and θ_{frame} to 0.3, 0.3, and 0.5, respectively. Moreover, we set the default value of J to 5. These values were verified to be optimal values through the comparative experiment in the previous study [12].

D. EVALUATION

In this experiment, by using the mir_eval [38] library, we evaluate the performance of each model with both frame-level metrics and note-level metrics. First of all, the frame-level metrics, referred to as *Frame* in this work, concentrate on comparing the existence of notes in each specific frame time. Therefore, the frame-level metrics are determined by a binary assessment of the congruence between a piano-roll representation of predictions and targets. Secondly, the note onset metrics termed *Note* in this work are the most basic metrics. It evaluates only the accuracy of pitches and onset timings within 50ms for each particular note, disregarding note offsets. Next, the *Note with Offset* metrics require not only matching onsets and pitches but also matching note offsets. The tolerance of offsets should be within either 0.2 times the duration of the reference note or 50 ms of each other. Lastly, the *Note with Offset & Velocity* incorporates velocity, thereby capturing the dynamics of individual note pieces more effectively. The precision, recall, and F1 scores are calculated per recording, and the average of scores was used for the final metrics.

V. RESULTS

A. MAIN EXPERIMENT

Table 1 shows the effectiveness of our proposed architecture compared with similar previous approaches. Firstly, for the *Note* F1 score, our model obtained the best score of 97.23% among all models. This represents a meaningful improvement

TABLE 3. Mutual Correlations between Onset, Offset, and Velocity in MLNSC system.

Sharing State	Frame			Note			Note w/ Offset			Note w/ Offset & Vel		
	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)
(a) None Sharing	94.06	83.75	88.52	98.95	89.82	94.08	83.74	76.16	79.69	82.22	74.84	78.28
(b) **Onset-Offset-Velocity**	94.04	85.10	89.27	98.67	91.34	**94.80**	84.59	78.44	**81.34**	83.33	77.34	**80.16**
(c) Onset-Velocity	93.96	84.39	88.84	98.66	91.25	94.75	83.48	77.34	80.24	82.51	76.49	79.34
(d) Onset-Offset	93.91	85.47	89.42	98.75	91.34	94.84	84.58	78.36	81.30	82.74	76.72	79.56
(e) Offset-Velocity	93.97	84.45	88.87	98.93	89.97	94.15	84.55	77.06	80.56	83.31	75.97	79.40
(f) All Note States	94.05	85.04	89.24	98.70	90.96	**94.60**	84.62	78.13	**81.18**	83.29	76.94	**79.93**

TABLE 4. Impact of Frame-wise Pitches on Mutual Correlations in MLNSC system.

Sharing State	Frame			Note			Note w/ Offset			Note w/ Offset & Vel		
	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)	Precision(%)	Recall(%)	F1(%)
(a) None Sharing	94.06	83.75	88.52	98.95	89.82	94.08	83.74	76.16	79.69	82.22	74.84	78.28
(b) Onset-Frame	94.10	83.97	88.66	98.77	90.01	94.11	83.98	76.67	80.09	82.62	75.49	78.83
(c) Offset-Frame	93.27	84.48	88.58	98.97	90.06	94.22	83.46	76.12	79.55	81.97	74.81	78.16
(d) Velocity-Frame	94.24	84.19	88.84	98.89	90.15	94.23	84.14	76.85	80.26	82.83	75.70	79.03

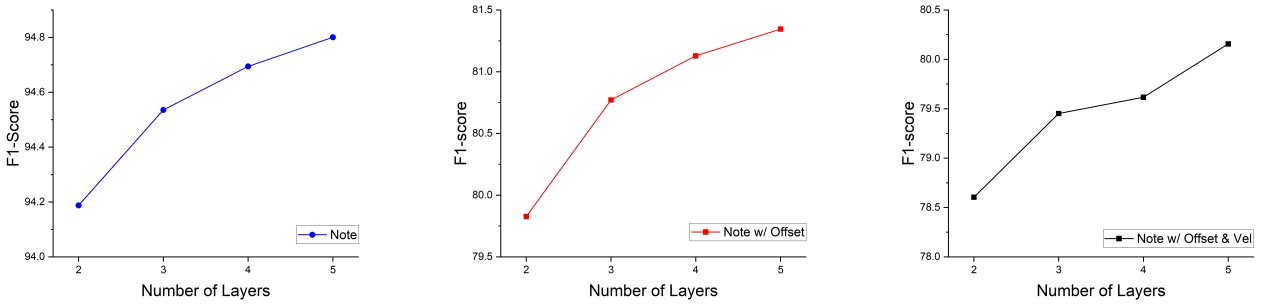

FIGURE 4. F1 score of *Note*, *Note with Offset*, *Note with Offset & Velocity* according to the number of sharing layers.

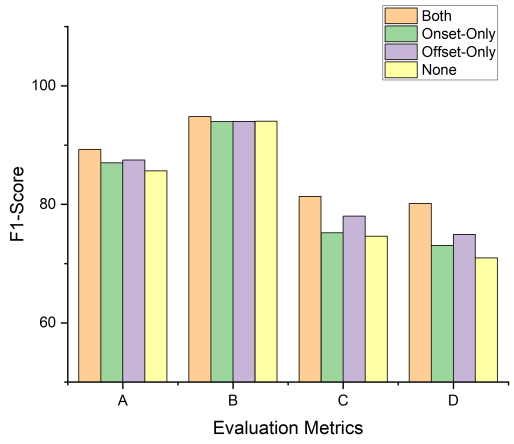

FIGURE 5. F1 score of evaluation metrics (A: *Frame*, B: *Note*, C: *Note with Offset*, D: *Note with Offset & Velocity*) according to the connection states.

over the baseline models within the HRTR system. Further, for the *Note with Offset* and *Note with Offset & Velocity* F1 scores, our model showed the second-highest score of 82.87% and 81.42%, respectively. Although our model couldn't show the best performance among all models, it shows a performance improvement in the *Note with Offset* F1 score from 82.47% to 82.87% and Note with *Offset & Velocity* F1 score from 80.92% to 81.42% compared to the High-Resolution model. Accordingly, we argue that our approach is effective in similar model architecture conditions.

One key observation is that the models focusing on musical context showed improved performance in *Note with Offset* F1 score. In the original paper of HPT-T [13], they applied a Transformer instead of RNNs for each frame-wise pitch, onset, offset, or velocity prediction. Despite the Transformer's ability to capture global musical context through its attention mechanism, it did not help predict frame-wise pitches, onset timing, or offset timing. They suggested this limitation arises because multi-pitch estimation and offset timing prediction are highly associated with short-term time dependencies. On the other hand, we can see that global musical context and note velocity prediction have a meaningful correlation. Interestingly, even though the Transformer was only integrated into the velocity detection module, there was an observed improvement in the *Note with Offset* F1 score, as indicated in Table 2. These results can also support our assumption that knowing two factors (onset and velocity) enables the model to infer the other factor (offset).

B. VERIFICATION OF MUTUAL CORRELATIONS BETWEEN ONSET, OFFSET, AND VELOCITY

To further analyze the distinct roles and effects of various note states, we conducted supplement experiments within the MLNSC system. By varying the combinations of the different note states, we compared them and derived the best combination. These were aimed at exploring the interaction in each combination of different note states to identify

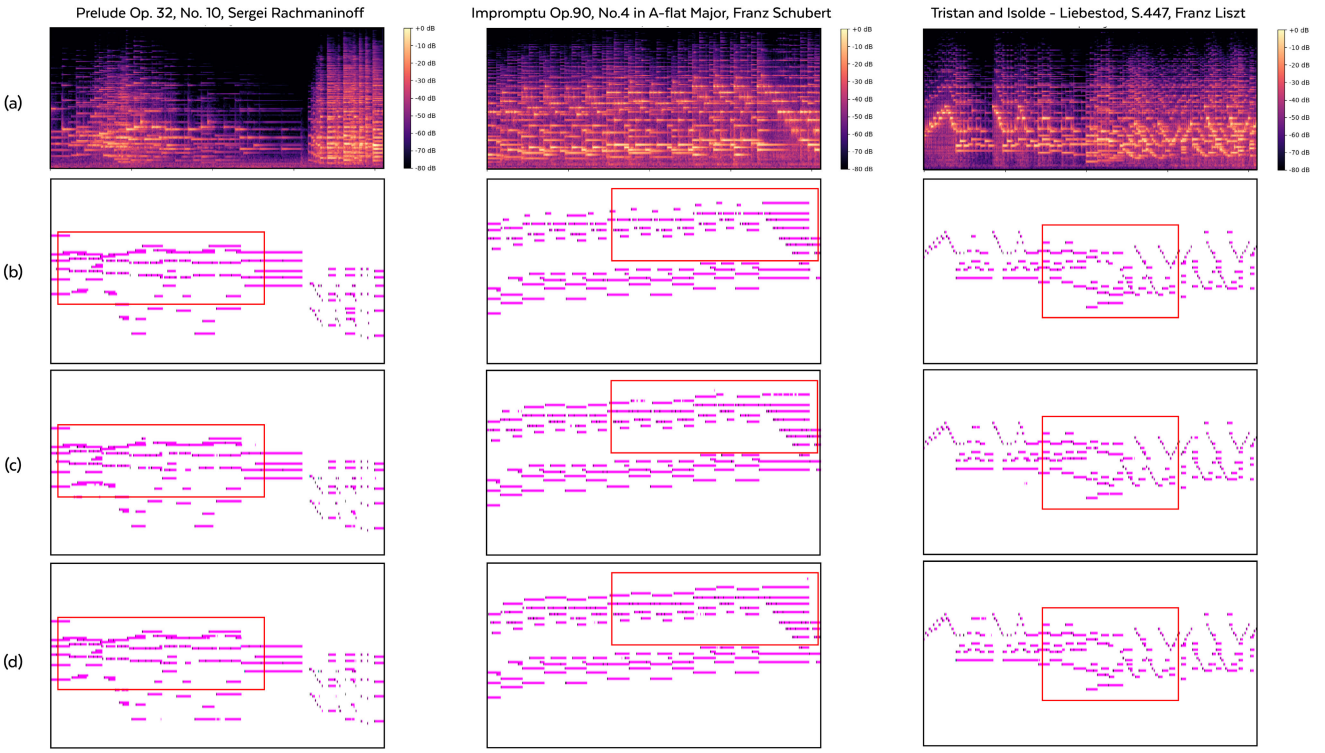

FIGURE 6. Comparison of (a) mel spectrogram, (b) ground truth, (c) none-sharing model, and (d) ours, to validate the impact of the kernel-sharing method.

the most effective configuration. The combinations tested included: (a) the model with no shared note states; (b) sharing Onset, Offset, and Velocity; (c) sharing Onset and Velocity; (d) sharing Onset and Offset; (e) sharing Offset and Velocity; and (f) sharing across all note states.

Comparing the results of (c), (d), and (e) relative to (a), depending on the note state we combine, we can confirm that the prediction performance for that corresponding note state particularly improves. First of all, (c) shows an improvement in the *Note* F1 score of 0.67% and *Note with Offset & Velocity* F1 score of 1.06%. Secondly, (d) shows an improvement in the *Note* F1 score of 0.76% and *Note with Offset* F1 score of 1.61%. Lastly, (e) shows an improvement in the *Note with Offset* F1 score of 0.87% and *Note with Offset & Velocity* F1 score of 1.12%. Consequently, among all the combinations, (b) emerges as the superior performer.

However, (f) surprisingly shows a decline in performance compared to (b). The decreases are observed across the *Note* F1 score by 0.2%, the *Note with Offset* F1 score by 0.16%, and the *Note with Offset & Velocity* F1 score by 0.23%, as indicated in the last row of Table 3. The difference between (f) and (b) is whether the frame-wise pitch is included or not. This led to further experiments aimed at identifying which specific combinations of frame-wise pitches and other note states might be causing this unexpected degradation in model performance.

Table 4 illustrates the effects of merging frame-wise pitches with other note states on the transcription system's performance. A notable point is that, unlike other combinations, (c) shows performance degradation compared to (a). Specifically, the scores of *Note with Offset* and *Note with Offset and Velocity*, which are the evaluation metrics related to offset prediction, decreased by 0.14% and 0.12%, respectively. This result can be explained by considering the inherent characteristics of the pitch envelope of the musical notes, where the note onset is typically marked by a peak, as shown in Figure 1. On the contrary, there is no noticeable point around the note offset in the pitch envelope. This characteristic makes the detection of offsets challenging, particularly in the context of overlapping notes. Consequently, the feature maps associated with frame-wise pitches proved less effective in aiding the accurate timing estimation of offsets.

C. ABLATION STUDY OF THE PROPOSED ARCHITECTURE

To discuss the impact of specific elements in our architecture, we conducted ablation studies. Firstly, we focused on the influence of the number of sharing layers within the kernel-sharing feature extractor module. The results, as illustrated in Figure 4, indicate a clear correlation between the count of sharing layers and the model's efficacy. Constrained by computing resources, we could only extend the number of shared layers to five. Despite computational constraints that capped our ability to extend beyond five shared layers, we observed a consistent uptick in performance correlating with each additional shared layer implemented. This trend

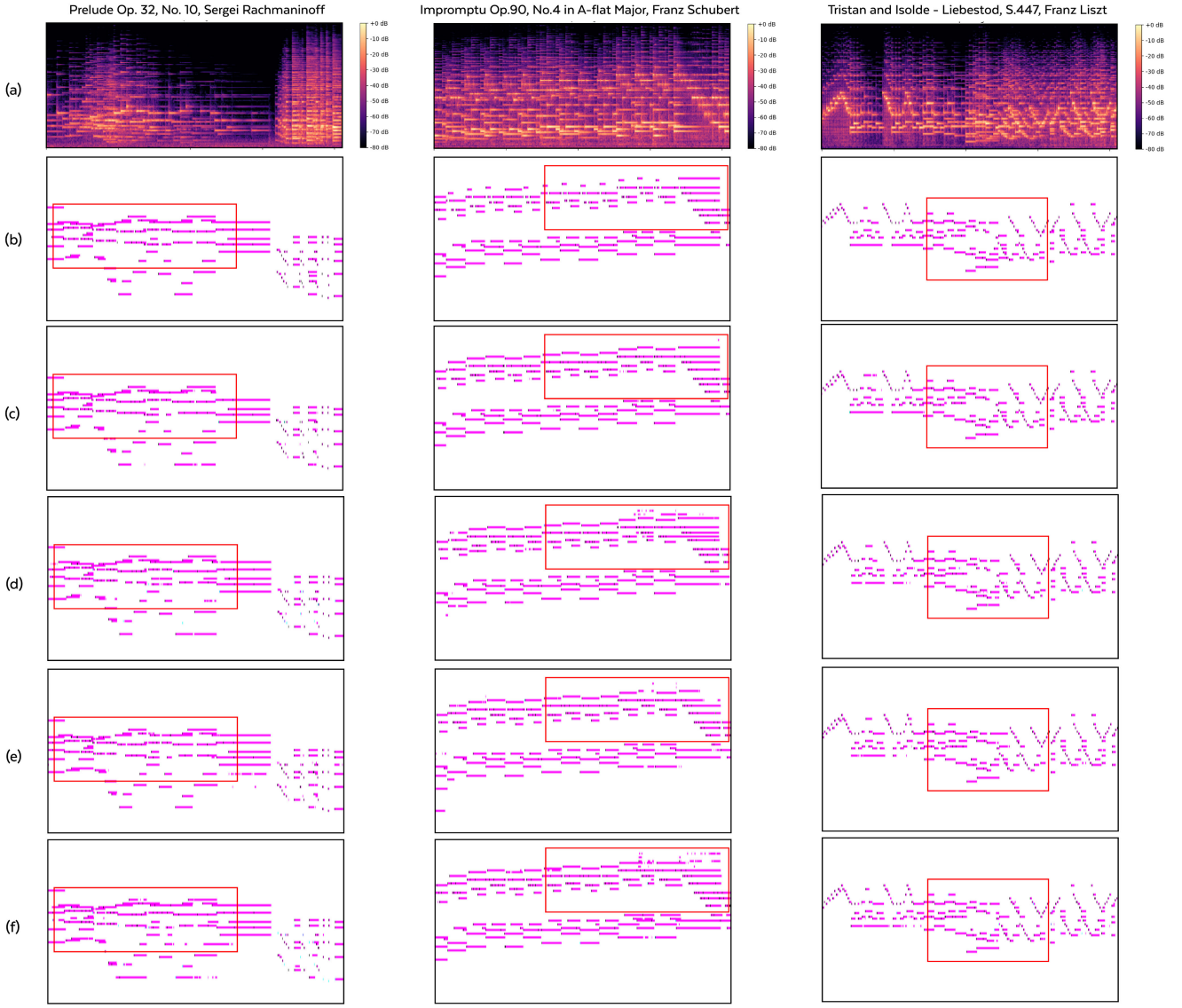

FIGURE 7. Comparison of (a) mel spectrogram, (b) ground truth, (c) both connections exist, (d) offset information only exists, (e) onset information only exists, and (f) no connection exists, to validate the impact of the connections between note state-specific detector modules.

underscores the potential benefits of deepening integration within the feature extraction module.

In the continuation of our ablation studies, we focused on the impact of connections between note state-specific detector modules. Figure 5 illustrates the comparison results of performance for the following 4 cases: (1) presence of both connections; (2) presence of only the onset connection; (3) presence of only the offset connection; and (4) absence of any connections. The performance outcomes were assessed using four evaluation metrics: *Frame* (A), *Note* (B), *Note with Offset* (C), *Note with Offset & Velocity* (D).

Interestingly, for metric B, there is no meaningful performance difference between (2), (3), and (4). For the reason that the note onset is typically marked by a peak, as shown in Figure 1, the presence or absence of additional information doesn't significantly impact the onset prediction. However, due to the absence of noticeable characteristics around the note offset, additional information is needed to predict offset timings. Accordingly, a significant contrast was observed in metrics C and D, related to predicting note offset. In conclusion, it was evident that restricting the timing information of either note onsets or note offsets led to a degradation in performance, with the impact standing out in the prediction of offset. Moreover, comparing the model without note onset information and the model without note offset information, the latter model performed worse compared to the former model.

D. QUALITATIVE ANALYSIS OF THE PROPOSED METHOD

In order to assess the practical effectiveness of our proposed architecture, we performed a comparative analysis using piano-roll visualizations. As illustrated in Figure 6, where

the x and y axes represent time and pitch, respectively, we compared the output from the non-sharing model (c) and our model (d) against the ground truth (b). First of all, the left column shows that (d) bears a closer resemblance to (b) than to (c). It is particularly noticeable in the time interval with highly overlapping notes where (c) frequently misses or prematurely ends the notes. On the right column, (d) appears to be superior in identifying the exact moments for pitch offsets. Furthermore, in the center column, a key observation is that (d) tends to generate relatively consistent notes, unlike (c), which produces fragmented notes. Although there are a few inaccuracies in the timing of offsets in (d)'s output, the notes appear to be in complete form. We thus argue that the utilization of our model leads to appropriate degrees of note duration that result in generating more consistent and natural-sounding instances.

In addition to our qualitative analysis of the proposed model, Figure 7 presents a side-by-side comparison of its various configurations: (c) both connections exist; (d) only the offset information exists; (e) only the onset information exists; and (f) no connections exist. In the output of (d) and (f), especially in the center column, what is particularly important to note are the effects of the restricted onset information. Restriction leads to the generation of finely segmented notes, especially visible at higher pitches. Meanwhile, other issues come to our attention in cases where the offset information is restricted, as seen in the output of (e). The models restricted to the offset information produce instances with comparatively irregular and inconsistent note lengths. Consequently, the absence of either significantly affects the model's ability to estimate the position and shape of the pitch envelope, leading to the generation of relatively low-quality output results.

VI. CONCLUSION

In this paper, we suggest our assumption of mutual correlations between different note states. This assumption was verified by comparing various combinations of different note states via the kernel-sharing feature extractor module. The kernel-sharing feature extractor module shares a kernel in each channel to capture common features represented in the given log-mel spectrogram. We also confirmed the impacts of connections in the note state detection step. A series of supplement experiments explained the novel notions in polyphonic piano music transcription systems. Moreover, the application of our method appears to be successful in improving model performance. Our approach therefore has the possibility of providing further enhancements to previous approaches. In addition, the method is not bound to the piano but can also be useful with regard to other instruments. Overall, we claim that our approach can be considered in wider domains of AMT and provides keystones for further related studies.

However, as discussed in the main experiment section, our model exhibits lower performance in predicting specific note states (offset and velocity) compared to the previous model. Our design primarily leverages CNNs and RNNs; however, we anticipate that integrating RNNs with self-attention mechanisms could enhance our model's ability to account for the global context of music. This integration is expected to improve our model's capability in predicting velocity, which will even affect the prediction of other musical elements such as onset and offset timing. In addition, as the research progresses, it can be seen that the number of parameters and amount of computation of the transcription model are gradually increasing. Therefore, employing techniques like quantization or pruning could be highly beneficial. Implementing such methods will not only make the model more efficient but also enhance its scalability and applicability in real-world scenarios.

REFERENCES

[1] N. Meade, N. Barreyre, S. C. Lowe, and S. Oore, "Exploring conditioning for generative music systems with human-interpretable controls," in *Proc. 10th Int. Conf. Comput. Creativity*. Charlotte, NC, USA: Association for Computational Creativity, Jun. 2019, pp. 148–155.

[2] C. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer: Generating music with long-term structure," in *Proc. 7th Int. Conf. Learn. Represent.*, New Orleans, LA, USA, May 2019, pp. 1–15.

[3] H.-W. Dong, K. Chen, S. Dubnov, J. McAuley, and T. Berg-Kirkpatrick, "Multitrack music transformer," in *Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[4] X. Wei, J. Chen, Z. Zheng, L. Guo, L. Li, and D. Wang, "A multi-scale attentive transformer for multi-instrument symbolic music generation," in *Proc. INTERSPEECH*, Aug. 2023, pp. 5391–5395.

[5] M. G. Armentano, W. A. De Noni, and H. F. Cardoso, "Genre classification of symbolic pieces of music," *J. Intell. Inf. Syst.*, vol. 48, no. 3, pp. 579–599, Jun. 2017.

[6] T. Tsai and K. Ji, "Composer style classification of piano sheet music images using language model pretraining," in *Proc. 21st Int. Soc. Music Inf. Retr. Conf.*, 2020, pp. 176–183.

[7] S. Kim, H. Lee, S. Park, J. Lee, and K. Choi, "Deep composer classification using symbolic representation," in *Proc. Late-Breaking Demo Session 21st Int. Soc. Music Inf. Retr. Conf.*, 2020, pp. 1–3.

[8] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 20–30, Jan. 2019.

[9] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proc. Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Sep. 2018, pp. 50–57.

[10] J. W. Kim and J. P. Bello, "Adversarial learning for improved onsets and frames music transcription," in *Proc. 20th Conf. Int. Soc. Music Inf. Retr. (ISMIR)*, 2019, pp. 670–677.

[11] K. W. Cheuk, Y.-J. Luo, E. Benetos, and D. Herremans, "Revisiting the onsets and frames model with additive attention," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.

[12] 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 Trans. Audio, Speech, Language Process.*, vol. 29, pp. 3707–3717, 2021.

[13] L. Ou, Z. Guo, E. Benetos, J. Han, and Y. Wang, "Exploring transformer's potential on automatic piano transcription," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 776–780.

[14] G. E. Poliner and D. P. W. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. 1, pp. 1–9, Dec. 2006.

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

[16] J. P. Bello, L. Daudet, and M. B. Sandler, "Automatic piano transcription using frequency and time-domain information," *IEEE Trans. Audio, Speech Language Process.*, vol. 14, no. 6, pp. 2242–2251, Nov. 2006.

[17] K. O'Hanlon and M. D. Plumbley, "Polyphonic piano transcription using non-negative matrix factorisation with group sparsity," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 3112–3116.
[18] L. Gao, L. Su, Y.-H. Yang, and T. Lee, "Polyphonic piano note transcription with non-negative matrix factorization of differential spectrogram," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 291–295.
[19] M. Marolt, "Transcription of polyphonic piano music with neural networks," in *Proc. 10th Medit. Electrotech. Conf. Inf. Technol. Electrotechnol. Medit. Countries*, vol. 2, 2000, pp. 512–515.
[20] S. Van Herwaarden, M. Grachten, W. B. De Haas, H.-M. Wang, Y.-H. Yang, and J. H. Lee, "Predicting expressive dynamics in piano performances using neural networks," in *Proc. 15th Conf. Int. Soc. Music Inf. Retr. (ISMIR)*, 2014, pp. 45–52.
[21] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 5, pp. 927–939, May 2016.
[22] M. A. Román, A. Pertusa, and J. Calvo-Zaragoza, "A holistic approach to polyphonic music transcription with neural networks," in *Proc. 20th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, Delft, The Netherlands, Nov. 2019, pp. 731–737.
[23] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. A. Huang, S. Dieleman, E. Elsen, J. H. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *Proc. 7th Int. Conf. Learn. Represent.*, New Orleans, LA, USA, May 2019, pp. 1–12.
[24] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *CoRR*, vol. abs/1511.08458, pp. 1–11, Dec. 2015.
[25] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306.
[26] D. Troxel, "Music transcription with a convolutional neural network," in *Proc. Music Inf. Retr. Eval. Exchange (MIREX)*, 2016, pp. 1–2.
[27] A. Cogliati, Z. Duan, and B. Wohlberg, "Context-dependent piano music transcription with convolutional sparse coding," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 12, pp. 2218–2230, Dec. 2016.
[28] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 121–124.
[29] 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. 17th Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, New York, NY, USA, Aug. 2016, pp. 475–481.
[30] R. Kelz, S. Böck, and C. Widnaer, "Multitask learning for polyphonic piano transcription, a case study," in *Proc. Int. Workshop Multilayer Music Represent. Process. (MMRP)*, Jan. 2019, pp. 85–91.
[31] T. Kwon, D. Jeong, and J. Nam, "Polyphonic piano transcription using autoregressive multi-state note model," in *Proc. 21st Conf. Int. Soc. Music Inf. Retr. (ISMIR)*, 2020, pp. 454–461.
[32] C. Thomé and S. Ahlbäck, "Polyphonic pitch detection with convolutional recurrent neural networks," in *Proc. MIREX*, 2017, pp. 1–4.
[33] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications*, vol. 5. Cham, Switzerland: Springer, 2015.
[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 448–456.
[35] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[36] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *Proc. IEEE Int. Conf. Signal Image Process. Appl. (ICSIPA)*, Nov. 2011, pp. 342–347.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.
[38] 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 *Proc. 15th Conf. Int. Soc. Music Inf. Retr. (ISMIR)*, vol. 10, 2014, p. 2014.

TAEHYEON KIM received the B.S. degree in electrical engineering and computer science from Gwangju Institute of Science and Technology (GIST), Republic of Korea, in 2023, where he is currently pursuing the Ph.D. degree in artificial intelligence with the AI Graduate School. His research interests include artificial intelligence, audio content analysis, and music signal processing.

DONGHYEON LEE received the B.S. and M.S. degrees in electrical engineering and computer science from Gwangju Institute of Science and Technology (GIST), where he is currently pursuing the Ph.D. degree with the School of Artificial Intelligence. His research interests include decision-making in multi-agent environments, with a particular emphasis on neural network architectures, reinforcement learning, and dynamic optimization. He demonstrated his expertise by securing third place in the 2018 CIG Fighting Game AI Competition.

MAN-JE KIM (Member, IEEE) received the B.S. degree in computer science from Sejong University, South Korea, in 2017, the M.S. degree from the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Republic of Korea, in 2019, and the Ph.D. degree from the AI Graduate School, GIST, in 2023. He is currently an Assistant Professor with the Convergence of AI, Chonnam National University. His research interests include reinforcement learning, human-like AI, video game AI, algorithms for game AI, and control intelligence. He received third place in the CIG Fighting Game AI Competition in 2015, 2017, and 2018. He was a Co-Organizer of the IEEE CIG 2015 Starcraft AI Competition.

CHANG WOOK AHN (Member, IEEE) received the Ph.D. degree from the Department of Information and Communications, Gwangju Institute of Science and Technology (GIST), Republic of Korea, in 2005. From 2005 to 2007, he was with the Samsung Advanced Institute of Technology, South Korea. From 2007 to 2008, he was a Research Professor with GIST. From 2008 to 2016, he was an Assistant/Associate Professor with the Department of Computer Engineering, Sungkyunkwan University (SKKU), Republic of Korea. He is currently a Professor with the School of Artificial Intelligence, GIST. His research interests include genetic algorithms/programming, multiobjective optimization, neural networks, and quantum machine learning.