Multi-Instrument Automatic Music Transcription With Self-Attention-Based Instance Segmentation

Yu-Te Wu, Berlin Chen, *Member, IEEE*, and Li Su, *Member, IEEE*

***Abstract*—Multi-instrument automatic music transcription (AMT) is a critical but less investigated problem in the field of music information retrieval (MIR). With all the difficulties faced by traditional AMT research, multi-instrument AMT needs further investigation on high-level music semantic modeling, efficient training methods for multiple attributes, and a clear problem scenario for system performance evaluation. In this article, we propose a multi-instrument AMT method, with signal processing techniques specifying pitch saliency, novel deep learning techniques, and concepts partly inspired by multi-object recognition, instance segmentation, and image-to-image translation in computer vision. The proposed method is flexible for all the sub-tasks in multi-instrument AMT, including multi-instrument note tracking, a task that has rarely been investigated before. State-of-the-art performance is also reported in the sub-task of multi-pitch streaming.**

***Index Terms*—Automatic music transcription, deep learning, multi-pitch estimation, multi-pitch streaming, self-attention.**

I. Introduction

AUTOMATIC music transcription (AMT), the task to convert acoustic music signals into music notation, is an enabling technology to music information retrieval (MIR), music generation, music search, music education, and musicology [1], [2]. The granularity of music notation for these application scenarios could vary largely from local properties such as pitch, onset, offset, dynamics, and timbre, to global properties such as voice, meter, and structure. To deal with such a complicated problem, AMT research is usually broken down into four different levels: frame-level transcription on pitches, which is also known as multi-pitch estimation (MPE); note-level transcription on pitches, onset, and duration, also known as note tracking (NT); stream-level transcription on notes and stream attributes, also known as multi-pitch streaming (MPS); and notation-level transcription on human-readable scores [1]. Comprehensive reviews on AMT can be found in [1], [2].

AMT of polyphonic music signals was considered to be the Holy Grail in music listening [3] due to the diverse problem scopes, high complexity of polyphonic signals, highly overlapped harmonic components, and the lack of labeled data, all of which have been long-standing issues repeatedly mentioned in the literature [1], [2], [4]. Most of the previous AMT studies focused only on single-instrument transcription (e.g., piano solo), or at the level of MPE, the latter is however an over-simplified task that cannot bring to true symbolic music notation by itself. Combining MPE with onset/offset detection [5]–[7] or instrument classification [8] to perform the NT or the MPS task was relatively less seen until the emergence of deep learning. Deep learning provides unprecedented flexibility in multi-task learning (MTL) [9] as it allows one to optimize multiple objective functions at the same time in an end-to-end manner. For example, the state-of-the-art piano transcription method employs two bidirectional long-short term memory (BLSTM) recurrent neural networks (RNN) to jointly predict the note-level attributes including pitch, temporal continuity of pitch, onsets, offsets, and dynamics by dual objective functions for onsets and frames [10], [11]. Such a rich set of labels guides the model much better than a single set of labels does, thereby boosting the performance up to the level suitable for automatic music generation [12]. Similar principles also apply for state-of-the-art multi-instrument recognition [13], [14] and multi-instrument MPE [15], where pitch and instrument classes are predicted jointly with the models based on convolutional neural networks (CNN). These facts indicate new opportunities for joint transcription of note-level attributes (i.e., pitch, onset, offset) and instrument-level attributes (i.e., instrument classes) of notes in multi-instrument sounds, a further step to conquer the challenge of AMT but relatively less investigated. In what follows, we refer to such a task as the *multi-instrument AMT* task.[1]

Multi-instrument AMT is a special case of the MPS task, where each stream represents an instrument class.[2] Previous studies on MPS mostly focus on identifying the stream (e.g., instrument) attribute for each pitch event obtained from MPE; these however are still frame-level transcription. To the best of our knowledge, transcribing both note and instrument of acoustic music signals has not been investigated in the literature except the few pioneer tries [8]. To distinguish this task from the typical frame-level MPS task (referred to as MPS), in this paper, we

Manuscript received April 23, 2020; revised July 30, 2020 and September 14, 2020; accepted September 14, 2020. Date of publication October 13, 2020; date of current version October 26, 2020. This work was supported in part by MOST Taiwan, under Contract MOST 106-2218-E-001-003-MY3. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stefan Bilbao. *(Corresponding author: Li Su.)*

Yu-Te Wu and Berlin Chen are with the Department of Computer Science and Information Engineering, National Taiwan Normal University, Taiepi 116, Taiwan (e-mail: freedombluewater@gmail.com; berlin@ntnu.edu.tw).

Li Su is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan (e-mail: lisu@iis.sinica.edu.tw).

Digital Object Identifier 10.1109/TASLP.2020.3030482

[1]In this article, multi-instrument AMT is the transcription task that the input signal and the output symbols are both multi-instrument (see Table I).

[2]Generally speaking, a *stream* can be a voice, a soundtrack, or an instrument class in polyphonic music. In this article, we consider the special case that a stream is an instrument, in order to facilitate the discussion.

will refer to the note-level MPS task as *note streaming* (NS). The challenges of the MPS and NS tasks are multi-fold. From the viewpoint of music signal processing, the model needs to discriminate the timbres of similar instruments (e.g., violin and viola) from a sound mixture. From the view of music language modeling, the model needs to track a stream of notes with various note lengths, transition, silence, polyphony, and even interweaving with other streams [16]. Moreover, the available data size of different instruments are usually highly imbalanced; this could result in a bias towards the majority instrument classes such as piano or violin.

This work also represents the first attempt to systematically approach the MPS and NS tasks of multi-instrument AMT with multi-task deep learning. The proposed solutions are partly inspired by computer vision (CV) techniques. Deep-learning-based CV techniques such as the U-net [17] have been shown useful in music source separation [18], melody extraction [19], MPE [20], and MPS [15]. These music processing tasks are considered to be the *semantic segmentation* task on 2-D signal representations, which usually make use of multiple channels to emphasize pitch saliency in the time-frequency plane [15], [20]. In a similar vein, the MPS and NS tasks also bear some resemblance to *instance segmentation* or *multi-object detection* problems in CV [21]. More specifically, the goal of NS is then to identify each note instance of each time-frequency pixel for every known note. When a note instance is bounded with the onset, offset, and pitch values, it is analogous to multi-object detection. When a note is a pitch contour with known start and endpoints, it behaves like instance segmentation. It is a feasible idea to solve the MPS/NS problem with related CV techniques.

We base our method on the previous work in [15], but with some key extensions. First, to improve the sequence modeling of music signals, we incorporate an effective self-attention sequence modeling mechanism [22] into the U-net model in [15] and propose a novel AMT model with enhanced performance. Second, the label smoothing (LS) technique is introduced into the training process in order to mitigate the data imbalance issue and improve the performance on the rarely-seen instrument classes. More importantly, regarding the complexity of AMT when pertaining to instrument information, we first specify the problem scenarios and the sub-tasks in multi-instrument AMT, and then propose the evaluation methodology for the MPS and NS task based on these problem scenarios, named the *instrument-agnostic* and the *instrument-informed* scenarios. Finally, based on [4], we propose a newly extended dataset for multi-instrument AMT research. To be specific, this paper contributes to several aspects, including:

- to define the problem scenarios and evaluation methods for multi-instrument AMT;
- to propose an improved multi-instrument AMT system which is suitable for all the problem scenarios and achieves state-of-the-art performance on multi-instrument note tracking;
- to show the effectiveness of self-attention mechanism on multi-instrument AMT by improving instrument identification accuracy; and
- to propose an extended dataset for multi-instrument AMT.

The rest of this paper is organized as follows. Section II specifies the problem scenarios of multi-instrument AMT. After that, Section III reviews the related work on multi-instrument AMT. Section IV presents the proposed methods, followed by Sections V–VII that report the performance study and discuss the associated results. Finally, we conclude our work in Section VIII.

II. Problem Scenarios

To investigate how the information of instrument classes challenges the multi-instrument AMT task, we consider three different transcription scenarios for multi-instrument music:

- *Instrument-informed* transcription: the instrument classes existing in the test music piece are assumed known. This is usually the case when the genre of music is known; for example, if the music piece to be transcribed is a violin sonata, then we may directly dismiss all channels other than violin and piano to simplify the task.
- *Instrument-agnostic* transcription: the most challenging case that the instrument classes existing in the test music piece are unknown, and need to be predicted by the model. Here, the only assumption is *closed-set* recognition of instrument classes, which means that the instrument classes of the test music piece are a subset of the classes in the training set. *Open-set* recognition (i.e., predicting the instrument classes not existing in the training data) is out of the scope of this paper.
- Instrument classes are not transcribed, no matter whether the test music piece is single- or multi-instrument. This scenario is equivalent to the conventional MPE (at frame-level) or NT (at note-level) tasks in AMT research.

Combining the different levels of AMT subtasks (i.e., MPE, NT, MPS, and NS) discussed in Section I with the three transcription scenarios mentioned above results in the following tasks: 1) MPE, 2) NT, 3) instrument-informed MPS, 4) instrument-informed NS, 5) instrument-agnostic MPS, and 6) instrument-agnostic NS. Comparison of these scenarios is highlighted in Table I.

III. Related Work

In this Section, a review focusing on multi-instrument AMT and Multi-Instrument reCognition (MIC) tasks is given. Readers are encouraged to refer to [1] for a comprehensive and updated review of general AMT research.

Most of the AMT systems proposed in the literature have been dedicated only for single-instrument inputs such as piano solo, or only for mono-track outputs where instrument information is ignored. There have been still widely-used AMT datasets which are equipped with multi-instrument signals alongside complete instrument/track labels for every note, such as the MIREX Multiple Fundamental Frequency Estimation (MF0) test set,[3] Bach10 dataset [23], Su dataset [4], RWC classical music dataset [24], MusicNet dataset [25], and Slakh dataset [26], to name but a few. The instrument labels provided by these datasets were however seldom used, except in certain studies on MPS [27]–[29].

[3][Online]. Available: https://www.music-ir.org/mirex

TABLE I
PROBLEM SCENARIOS OF MULTI-INSTRUMENT AMT

Data		Instrument info.	Scale of transcription		Type
Input (audio)	Output		Frame	Note	
Single- or multi-instrument	Single-instrument	–	Multi-pitch estimation (MPE)	Note tracking (NT)	Single-instrument AMT
	Multi-instrument	Informed	Multi-pitch streaming (MPS)	Note streaming (NS)	Multi-instrument AMT
		Agnostic			

Ironically, it has been long argued that instrument information is indispensable in an AMT model. The spectral patterns of an instrument class are key factors to guide the model in locating the time and pitch of that instrument class in a given mixture spectrum. This idea has been repeatedly proven successful in AMT systems using spectrogram factorization models such as non-negative matrix factorization (NMF) [30] and probabilistic latent component analysis (PLCA) [31]. In these studies, the mixture spectrogram is decomposed based on instrument-aware templates, which are usually trained on or sampled from single-note data with various instrument classes. For example, the AMT system demonstrated in [31] used templates containing the spectral patterns of 11 classes of instruments. If single-note data are not provided, constraints on spectral envelope and spectral smoothness can also be imposed to guide the model to capture patterns of different instrument classes [8], [30], [32]. In these cases, the instrument-awared templates are utilized only for decomposing the input spectrogram. Only a few studies in this direction took a further step to multi-instrument AMT, which calls for a more elaborate investigation on the outputs contributed by the templates of individual instruments. Examples include Independent Subspace Analysis (ISA) and factorial Hidden Markov Models (FHMM) [33], harmonic temporal clustering (HTC) [34], MPS with high-order HMM [35], Viterbi algorithm [36], constrained clustering [27]–[29], PLCA [8], [37], [38], neural networks [15], and others.

The MIC task is intended to identify the frame-level activity of instruments in the recordings of single- or multi-instrument ensembles. The datasets used in the research of MIR include the MedleyDB dataset [39], Open-MIC dataset [40], Mixing Secret dataset [41], IRMAS dataset [42], and others. Some of these datasets such as MedleyDB also provide pitch annotation for specific tracks. The aforementioned multi-instrument AMT datasets can all be used for MIC research as the instrument class label for every note is also provided. As a task closely related to multi-instrument AMT, the MIC task has also been solved by feature classification, spectrogram factorization [8], and more recently deep learning methods [43]–[45]. Recently, an MTL approach that leverages pitch information to enhance frame-level MIC is also proposed [13], [14], and in this case, the MIC task is equivalent to the MPS task [14].

IV. METHOD

The proposed multi-instrument AMT system includes three stages: pre-processing, the neural network model, and post-processing. Given the input signal $\mathbf{x}$ that is a mono-channel music signal, the system predicts a set of note events $\mathcal{N} := \{\mathbf{n}_i\}_{i=1}^{|\mathcal{N}|}$ from $\mathbf{x}$. The neural network model predicts a finite set of instrument classes $\mathcal{S}$ which depends on the class labels provided in the training data. A note event $\mathbf{n}_i$ contains four attributes, as denoted by $\mathbf{n}_i := (p_i, t_i^{\text{on}}, t_i^{\text{off}}, s_i)$, where $p_i \in [21, 108]$ is the pitch value in terms of MIDI number, $t_i^{\text{on}} \in \mathbb{R}^+$ is the onset time, $t_i^{\text{off}} \in \mathbb{R}^+$ is the offset time, and $s_i \in \mathcal{S}$ is the instrument class of $\mathbf{n}_i$.

A. Data Representation

Regarding multi-instrument AMT as an instance segmentation task on a time-frequency representation of the input signal, to discriminate the objects (i.e., salient regions on the time-frequency representation) of fundamental frequencies from the objects of the harmonics on the time-frequency representation is important. Following [15], [46], the data representation adopted in this work is based on the Combined Frequency and Periodicity (CFP) approach [47]. The data representation $\mathbf{Z}$ employed as the input of the model is derived from the short-time Fourier transform (STFT) matrix of $\mathbf{x}$, which is denoted by $\mathbf{X}$. For simplicity, only the STFT in the positive frequency range is used. $\mathbf{Z}$ contains two channels, namely the *frequency representation* $\mathbf{Z}_f$, and the *periodicity representation* $\mathbf{Z}_f$. They are derived from $\mathbf{X}$ with the generalized cepstrum [48], high-pass filters $\mathbf{W}_f$ and $\mathbf{W}_t$ [46], [47], and log-frequency filterbanks $\mathbf{Q}_f$ and $\mathbf{Q}_q$:

$$\mathbf{Z}_f := \mathbf{Q}_f \left|\mathbf{W}_f \mathbf{X}\right|^{\gamma_f} , \tag{1}$$

$$\mathbf{Z}_q := \mathbf{Q}_q \left|\mathbf{W}_q \mathbf{F}^{-1} \mathbf{Z}_f\right|^{\gamma_q} , \tag{2}$$

where $\mathbf{Z}_f, \mathbf{Z}_q \in \mathbb{R}^{K \times N}$, and N and K are the number of time and frequency bins in $\mathbf{Z}$, respectively. The generalized cepstrum of $\mathbf{Z}_f$ is $|\mathbf{F}^{-1}\mathbf{Z}_f|^{\gamma}\ \mathbf{F}$, where $\mathbf{F}$ denotes the discrete Fourier transform (DFT) matrix, and $|\cdot|^{\gamma}$ is an element-wise power-scaled nonlinear function such that $|x|^{\gamma} = x^{\gamma}$ if $x \geq 0$, and $|x|^{\gamma} = 0$ if $x < 0$. As suggested in [49], we set the parameters $(\gamma_f, \gamma_q) = (0.24, 0.6)$. $\mathbf{W}_f$ and $\mathbf{W}_t$ are two high-pass filters to remove low-varying components in $\mathbf{Z}$, which are in general irrelevant to pitch. $\mathbf{Q}_f$ and $\mathbf{Q}_q$ are two triangular filterbanks: $\mathbf{Q}_f$ maps a feature from the frequency domain to the log-frequency domain, and $\mathbf{Q}_q$ maps a feature from the time domain to the log-frequency domain. Both filterbanks have 352 triangular filters, ranging from 27.5 Hz (A0) to 4,435 Hz (one quarter semitone below #C8), and the resolution is 48 semitones per octave. In sum, $\mathbf{Z}_f$ reveals the fundamental frequencies and their harmonics in a signal, and $\mathbf{Z}_q$ reveals the fundamental frequencies and their sub-harmonics [47], [50]. That means, the 'consensus' of $\mathbf{Z}_f$ and $\mathbf{Z}_q$ is a representation of pitch [46], [47], [50]. The peaks appearing on both $\mathbf{Z}_f$ and $\mathbf{Z}_q$ tend to be at the positions true pitch activation, as exemplified in Fig. 1, where the common peaks of $\mathbf{Z}_f$ and $\mathbf{Z}_q$ effectively locate the position of the three pitches in the signal. In this way, considering the

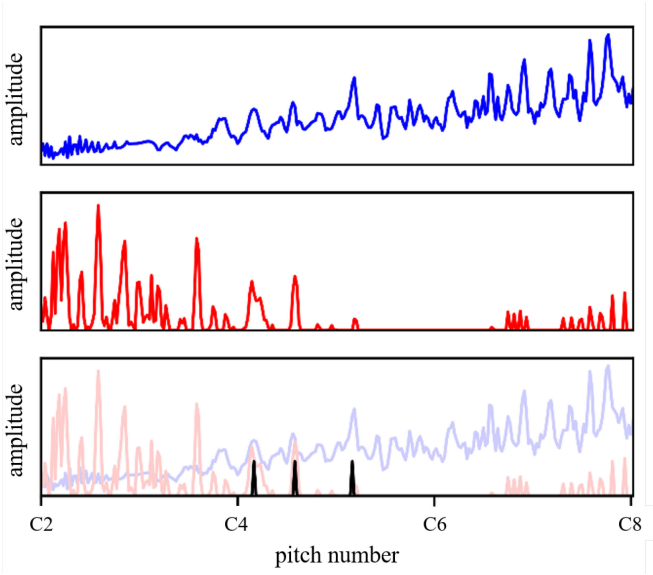

Fig. 1. Illustration of powered spectrum ($\mathbf{Z}_f$, upper), generalized cepstrum ($\mathbf{Z}_q$, middle), and ground truth (lower in gray, overlayed with $\mathbf{Z}_f$ and $\mathbf{Z}_q$). Amplitudes re-scaled to arbitrary unit for better illustration. Data sampled from the frame at 11.0 sec in the test clip '2628.wav' in the MusicNet dataset.

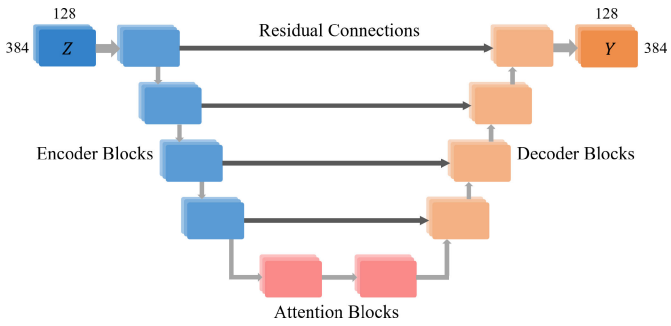

Fig. 2. The architecture of the proposed model.

TABLE II
DETAILED MODEL ARCHITECTURE

	Conv	Attn
Input	Input Feature: $K \times N \times \lvert\mathcal{S}\rvert$	
Encoder	**Conv:** out_channel(32)/ size(7, 7)/ stride(1, 1)	
	Encoder block: (32)/(3, 3)/(2, 2) **Encoder block:** (32)/(3, 3)/(1, 1)	
	Encoder block: (64)/(3, 3)/(2, 2) 2× **Encoder block:** 64/(3, 3)/(1, 1)	
	Encoder block: 128/(3, 3)/(2, 2) 3× **Encoder block:** 128/(3, 3)/(1, 1)	
	Encoder block: 256/(3, 3)/(2, 2) 4× **Encoder block:** 256/(3, 3)/(1, 1)	
Bottleneck	**ASPP**: out_channel(512)/ size(3, 3)/stride(1,1)/ dilation(1) **ASPP:** (512)/(3, 3)/(1,1)/(2) **ASPP:** (512)/(3, 3)/(1,1)/(4)	**Attn**: out_channel(64)/ query(100, 32)/flange(8, 8) num_of_head(8) **Attn:** (128)/(64, 16)/(8, 8)/(8)
Decoder	**Conv:** 256/(1, 1)/(1, 1)	
	Decoder block: 128/(3, 3)/(2, 2)	
	Decoder block: 64/(3, 3)/(2, 2)	
	Decoder block: 32/(3, 3)/(2, 2)	
	Decoder block: 32/(3, 3)/(2, 2)	
Output	Output Prediction: $K \times N \times \lvert\mathcal{S}\rvert$	
# Param.	12,571,907	7,920,343

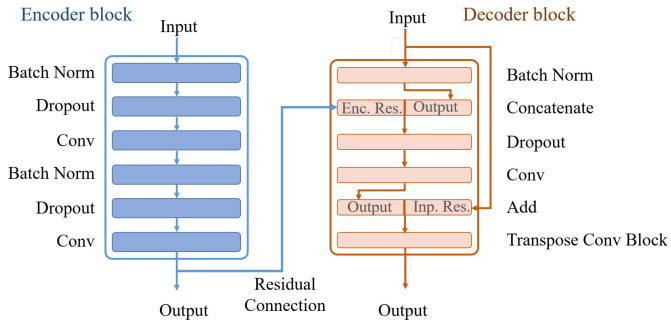

Fig. 3. Layers inside an encoder block and a decoder block. The residual connection is the same as the one shown in Fig. 2.

harmonics or sub-harmonics out of the pitch range of interest is no longer mandatory. This data representation has been shown useful in MPE, and has been applied in other data representations such as Harmonic Constant-Q Transform (HCQT) [20] and multi-layered cepstrum (MLC) [51].

The input audio recordings are mono-channel with a sampling rate of 44.1 kHz. The STFT is computed with a Blackman-Harris window whose size is 0.18 second, namely 7,939 samples. The hop size of the STFT is 0.02 seconds.

B. Model

Fig. 2 illustrates the proposed multi-instrument AMT model, and Table II shows the detailed settings of the model architecture. In general, the model contains an encoder, a decoder, and two self-attention blocks that connect the encoder and the decoder. The encoder contains four block groups, each of which consists of 2, 3, 4, and 5 encoder blocks, respectively. Each group of encoder blocks has a corresponding decoder block, and are connected with a skip connection link in between them [19], thereby forming a U-net structure [17]. The structure of the encoder block/ decoder block pair is shown in Fig. 3. The skip connection is implemented by concatenating each encoder block output to its corresponding decoder block. The first two dimensions of each hidden layer are $K \times N$ the same as the input. That means the output channel and the input channel has the same dimension.

This model is originated from DeepLabV3+ [52], a state-of-the-art image semantic segmentation method constructed with fully convolutional neural networks with an encoder-decoder architecture. In [52], the atrous spatial pyramid pooling (ASPP) mechanism is used in between the encoder and decoder. It employs *dilated convolution* to enlarge the reception field, such that a convolution kernel can capture objects in various scales by varying the size of dilation r:

$$\mathbf{y}[i,j] = \sum_{m,l} \mathbf{u}[i + rm, j + rl]\mathbf{w}[m,l]\,, \tag{3}$$

where $\mathbf{u}$ and $\mathbf{y}$ denote the input and output 2-D feature maps, respectively, $\mathbf{w}$ is the convolution filter to be learned, and $[i,j]$ indicates the location on the feature maps. The standard convolution is a special case when $r = 1$. ASPP then performs dilated convolution with multiple dilation sizes and pool the resulting

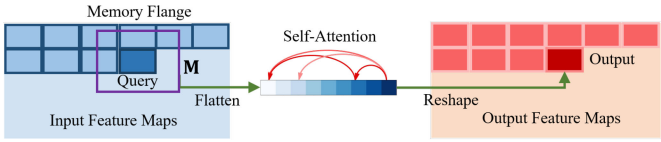

Fig. 4. Mechanism of the self-attention block. The input feature map is partitioned into overlapped blocks, and the blocks are fed into the self-attention layers in raster-scan order.

feature maps together. ASPP has been applied in melody extraction and AMT in [19] and [15].

Since temporal modeling is important in music transcription, in this paper we also capitalized on the self-attention mechanism, which has been proven effective in various sequence modeling problems. As shown in Fig. 2, the ASPP blocks in the DeepLabV3+ model are replaced by self-attention blocks. We adopt the Image Transformer [22], a self-attention-based image generation framework, to implement the self-attention layer. It is instantiated with the multi-head self-attention mechanism:

$$\text{Attention}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right)=\text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{T}}{\sqrt{n}}\right)\mathbf{V}\,, \tag{4}$$

where $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are linear-transformed feature maps from the original input, and n represents the number of heads. In this paper, we set $n=8$. To apply the self-attention mechanism on the feature map, the most direct way is to flatten the feature map with size $d\times K\times N$ to a sequence with size $d\times KN$, where d is the feature dimension and KN is the length of the sequence. This is however impractical because in our case, the length of the sequence is more than 50,000 ($K=384$ and $N=128$) and a desktop computer cannot afford the memory consumption of the quadratic term $\mathbf{Q}\mathbf{K}^{T}$. To overcome this issue, [22] proposed the idea to divide a feature map into *query blocks* to represent local information, and apply self-attention to the sequence of query blocks. This not only saves memory space but also enables parallel computations, as the mechanism works in batches of blocks. Specifically, the input of a self-attention layer, $\mathbf{M}$, is essentially a query block with a *memory flange* that pads the receptive field, as illustrated in Fig. 4. Before $\mathbf{M}$ is fed into (4), its time and pitch dimensions are flattened. The flattened vectors are multiplied by three different learnable weights, $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$, and the outcomes are therefore the $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ for (4), respectively. All the query blocks are processed in batch. A self-attention block is then formulated as [22]:

$$\mathbf{q}_a=\text{layernorm}\left(\mathbf{q}+\text{dropout}(\text{Attention}\left(\mathbf{Q},\mathbf{K},\mathbf{V}\right))\right), \tag{5}$$

$$\mathbf{q}'=\text{layernorm}\left(\mathbf{q}_a+\text{dropout}(\mathbf{W}_1\text{ReLu}(\mathbf{W}_2\mathbf{q}_a))\right), \tag{6}$$

where $\mathbf{Q}=\mathbf{W}_Q\mathbf{q}$, $\mathbf{K}=\mathbf{W}_K\mathbf{q}$ and $\mathbf{V}=\mathbf{W}_V\mathbf{q}$. $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$, $\mathbf{W}_1$, and $\mathbf{W}_2$ are the parameters to learn. In (5), $\mathbf{q}$ is the flattened feature map bounded by memory flange. Equation (6) is a feed-forward neural network, where $\mathbf{W}_1$ and $\mathbf{W}_2$ are the parameters shared across all the positions in a layer. In this paper, two such self-attention blocks are employed to connect the encoder and the decoder, as shown in Fig. 3. It is noted that different from [22], [53], positional embedding vectors are not used in this work. The sequential ordering information from the input can be propagated to the output directly through the residual connection layers in the U-Net.

C. Output

The output of the proposed model is a multi-channel representation where each channel is also a time-frequency image with size $K\times N$. The model predicts $|\mathcal{S}|$ classes of the instrument, and for each instrument class $s\in\mathcal{S}$, we use two event-type channels to represent the likelihood a note event occurs at specific time and pitch, one for note onset (denoted as $\mathbf{Y}_s^{\text{on}}$) and the other for pitch activation (i.e., the process between note onset and note offset events, denoted as $\mathbf{Y}_s^{\text{act}}$). Note offset and the whole note event are identified in the post-processing stage based on these two channels. We further add one channel in the output to represent the classes of "others." In total, there are $2|\mathcal{S}|+1$ channels at the output of the model.

The label distributions of polyphonic music signals are highly unbalanced. First, most of the pitch activation and onset labels on the time-frequency plane are zero-valued (i.e., silence). A note event on a piano roll is merely a line in a two-dimensional array, and its onset even just occupies one pixel. The model trained with such labels tends to predict all the examples as zero when using pixel-wise binary cross-entropy as the loss function. We therefore introduce the focal loss [54] to solve this problem. In computer vision, focal loss has been proven effective in the one-stage dense object detection problem with extremely dense examples of background classes but sparse examples of foreground classes. In audio processing, focal loss has also been shown useful in vocal melody extraction [19] and MPE [15]. For a prediction value p and its ground truth y at a pixel, the focal loss is defined as

$$\text{FL}(p_t)=-\alpha_t(1-p_t)^{\gamma}\log(p_t)\,. \tag{7}$$

The focal loss is parameterized by a weighting factor $\alpha\in[0,1]$ and a focusing factor $\gamma\in[0,1]$, and in (7) we have $\alpha_t=\alpha$ if $y=1$ and $\alpha_t=1-\alpha$ otherwise, and $p_t=p$ if $y=1$ and $p_t=1-p$ otherwise. The parameter α is employed to balance the loss from the activation and silence examples, and the term $(1-p_t)^{\gamma}$ is employed to balance the loss from the examples which are correctly predicted and those which are wrongly predicted. As [54] recommended, we set $\alpha_t=0.25$ and $\gamma=2$ in this work. The total loss function is therefore the sum of the values in (7) over all the pixels and channels (including onset and also activation channels).

Besides the data imbalance between activation and silence events, the distribution of the instrument classes is also highly imbalanced. As we will see in Table III, most of the instrument classes are in the extreme minority. In this case, a model tends to be over-confident in predicting all instruments to the majority classes. This issue not only occurs in the case of multi-instrument transcription. Rather, in the case of piano solo transcription, there is still data imbalance between the class 'piano' and the class 'others.' To deal with this issue, the label smoothing (LS) method smooths the label distribution by imposing a penalty to the over-confident output. More specifically, given a dataset with samples x, label y_s, and with $|\mathcal{S}|$ different classes, $1\leq s\leq|\mathcal{S}|$.

TABLE III
INSTRUMENT CLASSES, ABBREVIATIONS, AND THE PORTION OF NOTE LENGTH IN EACH MULTI-INSTRUMENT DATASET

Instrument class	% in MusicNet	% in Ext-Su	% in URMP
Piano (pn)	59.24	28.62	–
Violin (vn)	16.23	30.89	34.71
Cello (vc)	9.37	10.09	14.16
Viola (va)	8.16	13.35	15.38
Clarinet (cl)	2.2	6.08	10.86
Horn (hn)	1.3	2.24	–
Bassoon (bn)	1.39	3.77	–
Flute (fl)	0.57	2.85	15.37
Oboe (ob)	0.72	1.51	6.01
Contrabass (db)	0.29	0.61	3.52
Harpsichord (hpd)	0.53	–	–

With the predicted label $\hat{y}_s$, the ideal label distribution is then $D(\hat{y}_s|x) = \Phi_{\hat{y}_s,y}$, where $\Phi_{\hat{y}_s,y_s} = 1$ for $\hat{y}_s = y_s$ and $\Phi_{\hat{y}_s,y_s} = 0$ otherwise. In the label smoothing scheme, a modified label distribution is considered:

$$D'(\hat{y}_s|x) = (1-\lambda)\Phi_{\hat{y}_s,y_s} + \lambda p(|\mathcal{S}|)\,, \tag{8}$$

where λ represents smooth factor, and $p(\cdot)$ is a prior label distribution and is usually assumed to be uniform distributed over the $|\mathcal{S}|$ classes, which means that with probability λ, the label distribution is uniform. Though other kinds of distributions have been discussed recently [55], we follow this assumption and set $p(|\mathcal{S}|) = 1/|\mathcal{S}|$. For the smoothing parameter λ, we set λ to 0.1 in this study. Further details on the label smoothing method can be found in [56], [57].

D. Note and Instrument Extraction

The output values $\mathbf{Y}_s^{\text{on}}[k,n]$ and $\mathbf{Y}_s^{\text{act}}[k,n]$ represent the likelihood of onset and activation of instrument class s, respectively. $\mathbf{Y}_s^{\text{on}}$ determines the onset time of a note, and $\mathbf{Y}_s^{\text{act}}$ determine the duration of a note. Their values are between 0 and 1. To quantify the transcription results, a threshold value is required to binarize the output values. The simplest way is to set a constant threshold and clip the output values to zero or one. This way is however ineffective in multi-instrument transcription, as the distribution of the output values could vary with the instrument classes. Extra processing is therefore required to determine the threshold values adaptively. This process contains four steps: global normalization, instrument selection, local normalization, and note inference. Details are described as follows.

1) Global Normalization: All the output values from all the channels are normalized together by z-scoring, such that the mean of the output is zero and its standard deviation is one.

2) Instrument Selection: This process follows the *all-or-none* principle: the instrument classes appearing in the prediction result are selected by a global threshold θ^{ins}. We define the *confidence value* v_s of an instrument class s as the standard deviation of the elements in $\mathbf{Y}_s^{\text{on}}$ (denoted as σ_s^{on}) plus the standard deviation of the elements in $\mathbf{Y}_s^{\text{act}}$ (denoted as σ_s^{act}) over all elements in that channel, resulting in $v_s := \sigma_s^{\text{on}} + \sigma_s^{\text{act}}$. A low value of v_s implies that $\mathbf{Y}_s^{\text{on}}$ and $\mathbf{Y}_s^{\text{act}}$ approximate all-zero prediction, and this means that the instrument class s might not exist. On the other hand, a high value of v_s indicates that s is used in that music piece. The set of selected instrument $\mathcal{S}_p$ is therefore those classes which v_s values are greater than a threshold θ^{ins}. Those classes with v_s values lower than θ^{ins} are considered absent from that music piece. The value of θ^{ins} is fine-tuned from the validation set.

3) Local Normalization: The z-score normalization process is again applied, but this time it is applied in a channel-wise manner for the output of every selected instrument class $\mathbf{Y}_{s'}^{\text{on}}$ and $\mathbf{Y}_{s'}^{\text{act}}$, $s' \in \mathcal{S}_p$. After this normalization process, we filter out the values smaller than the thresholds θ_s^{on} and θ_s^{act} for the onset and activation channels, respectively. The threshold values are fine-tuned from the validation set.

4) Note Inference: After the normalization and thresholding processes, the resulting $\mathbf{Y}_{s'}^{\text{on}}[k,n]$ and $\mathbf{Y}_{s'}^{\text{act}}[k,n]$ are then used for onset and note duration inference. A note onset position $[k^{\text{on}}, n^{\text{on}}]$ is detected at the position that $\mathbf{Y}_{s'}^{\text{on}}[k,n]$ is at a local maximum, and the minimum distance between two consecutive peaks is set to be $\eta = 50$ ms; this can be done by finding the maximum over a sliding window with a length of $2\eta = 100$ ms. When an onset event is detected, it triggers a mechanism to find its corresponding offset event in $\mathbf{Y}_{s'}^{\text{act}}$. This offset event is at $[k^{\text{on}}, n^{\text{on}} + \delta]$, where the note duration δ is determined by the smallest value that introduces a silence interval ξ longer than 60 ms started from $n^{\text{on}} + \delta$.

V. EXPERIMENTS

A. Settings

We compare different models working in conjunction with the label smoothing strategy discussed in Section IV. More specifically, we consider the following three settings:

- Using the ASPP to connect the encoder and decoder. Label smoothing is not applied in the training process. Since this model is a fully convolutional network, such a setting is denoted by **Conv** hereafter.
- Using the ASPP to connect the encoder and decoder. Label smoothing is applied in the training process. This setting is then referred to as **Conv-LS**.
- Using the self-attention mechanism to connect the encoder and decoder. Label smoothing is applied in the training process. This setting is designated as **Attn-LS**. In this case, the ASPP layers are not used.

There are two disparate ways to execute instrument-agnostic transcription using the proposed model. The first is to train the model directly with single-instrument outputs (e.g., $|\mathcal{S}| = 1$), and the second is to train the model with multi-instrument outputs, but in the end sum all the results over different channels together. In this paper, we consider the latter one. More specifically, we do compare the performance of the three models on MPE, NT, instrument-informed MPS/NS, and instrument-agnostic MPS/NS (see Section II), but all the three models presented *are trained only for the multi-instrument NS task*. That is, the results of MPE and NT are all derived from the results of the NS task. The MPE results are obtained by summing the outputs over all the channels, and performing normalization and thresholding in the same way as that previously described in Section IV-D for the summed channel. For NT, the onset and

pitch activation channels are summed individually into two channels. Normalization and thresholding are applied independently on the two channels. We do not train the models specifically for MPE and NT because of two reasons: 1) the effectiveness of using the proposed model on MPE has been demonstrated in [15], and 2) there is good reason reporting the *degenerated* (obviously underestimated) performance of MPE and NT so as to demonstrate the generalization power of the proposed model.

B. Datasets

We perform instrument-agnostic transcription on both the single- and multi-instrument datasets, and instrument-informed and instrument-agnostic transcription on multi-instrument datasets. For generality, the chosen multi-instrument datasets also contain a few single-instrument clips. The details of the dataset are introduced as follows.

The single-instrument dataset used in the experiments is a subset of the MAPS dataset [58]. This subset contains 60 piano solo recordings (ENSTDkCl and ENSTDkAm) and is often used as the benchmark in piano solo transcription [58]. Following the state-of-the-art piano transcription methods, we trained our models on the MAESTRO dataset [11], an external dataset containing 1,184 real piano performance recordings collected from International Piano-e-Competition, with a total length of 172.3 hours. As this dataset is single-instrument, only the MPE and NT results are reported.

Three multi-instrument datasets are used. The first is the MusicNet [59] dataset, which contains 330 pieces of solo and ensemble music. These music pieces are all real-world performance, and the ground-truth is generated by audio-to-score alignment. This dataset contains 11 classes of instruments, namely piano (pn), violin (vn), viola (va), cello (vc), flute (fl), horn (hn), bassoon (bn), clarinet (cl), harpsichord (hpd), contrabass (db), and oboe (ob). We follow the partition of train and test sets used in [59]: 320 pieces are for training and the remaining 10 pieces are for testing. As mentioned above, the number of different instrument samples is highly imbalanced. Table III shows the portion of the total length of different instrument types. The length is computed by accumulating each note length, rather than only considering the length of the appearance of each instrument in the music piece. Since the model is supposed to predict multiple notes at the same time in AMT, thereby for this fact, the notes should be counted separately. As can be seen in Table III, the piano recordings constitute a far larger portion than the others in the dataset, whereas contrabass gives almost no contribution to the length. Note that the training set contains all the 11 instrument classes while the test set only contains 7 classes among them, see Table IV. For the purpose of validating thresholds, we randomly pick up 40 pieces from the training set as the validation set. The second dataset is an extension of the Su dataset, which has been taken into the MIREX MF0 campaign since 2015. The original Su dataset contains three piano solos, three string quartets, two piano quintets, and two violin sonatas, amounting to 10 pieces. The ground-truth labels were annotated by human experts with a "co-performance" process described in [4] and were made in a multi-track format. In this paper, we introduce the extended Su dataset, which contains 30 Western classical music recordings with multi-track labels annotated with the same process.[4] The dataset contains excerpts from four symphonies, eight piano quintets, nine string quartets, five violin sonatas, and four woodwind quintets, covering the 10 classes of the instrument in the MusicNet dataset. The total length of the clips is 852 seconds, and the total length of the notes is 3124.68 seconds. We evaluate the MPE and NT tasks on the original Su (denoted as Su-10 hereafter) dataset and evaluate the MPS and NS tasks on the extended Su (denoted as Ext-Su hereafter) dataset.

TABLE IV
INFORMATION OF INSTRUMENT CLASSES IN THE MUSICNET TEST SET

File name	Number of instrument classes / details	Length (secs)
1759	1 (pn)	530
1819	3 (2 hn, 2 bn, 2 cl)	580.2
2106	3 (2 vn, va, vc)	640.6
2191	1 (vn)	97
2298	1 (vc)	150.4
2303	1 (pn)	220
2382	3 (2 vn, va, vc)	243.4
2416	3 (2 hn, 2 bn, 2 cl)	267.7
2556	1 (pn)	436.8
2628	2 (pn, vn)	356.6

The third dataset used is the Multi-Modal Music Performance (URMP) dataset [60]. The dataset contains music, ground-truth MIDI, and video clips of 44 ensemble performances. We take the music recordings and the ground-truth for our experiments. We remove all the clips which contain the instrument classes unseen in the MusicNet dataset and take the remaining 22 clips for the experiment. We denote this remaining set as URMP-22. Note that the URMP-22 dataset contains only 7 classes of the instrument, and some of the majority classes (e.g., flute, with 15.37% of total length) in URMP-22 are the minority in MusicNet (1.62% only). Detailed statistics of these datasets are shown in Table III. We use the models trained on the MusicNet training set and test on the Ext-Su and URMP datasets. That is, the latter two datasets are evaluated as the external data, which have different portions of instrument classes as shown in Table III.

C. Training

The input of the model is the CFP representation introduced in Section IV. Each input segment has a dimensionality of $352 \times 128 \times 2$, where 128 is the number of time frames, 352 is the total number of pitch values (i.e., 48 bins per octave), and 2 is the number of channels. The ground-truth activation channel is a piano roll representation with the same size and resolution. The ground truth pitches are placed assuming ideal tuning. To facilitate the training process, each single-frame onset label is extended to three frames with fading-out boundaries: for an onset event at the time step n_i, the frames at $n_i - 1$, n_i, and $n_i + 1$ are all labeled as one, while the frames at $n_i \pm 2$, $n_i \pm 3$, and $n_i \pm 4$ are labeled as 1/2, 1/3, and 1/4, respectively. The

[4]To include more multi-instrument recording, in the extended Su dataset, we removed the three piano solos from the original Su dataset.

default hop size is 20 ms. The batch size is set to 8, while 3,000 steps for each epoch. We use Adam optimizer to fine-tune the model with the initial learning rate set to 0.001. The maximum number of training epochs is set to 20, with an early-stopping mechanism after the validation accuracy reaches a maximum for 6 epochs. The model is trained on two RTX-2080 GPU cards, an i9-9820X CPU with 20 cores, and 128 GB RAM on an Ubuntu-18.04 machine. Typically, it would take around 12 hours to finish the training process for a model. All the experiment codes and checkpoints can be found on: https://github.com/BreezeWhite/Music-Transcription-with-Semantic-Segmentation.

D. Evaluation Metrics

To our knowledge, evaluation of multi-instrument AMT has not been systematically discussed. Previous MPS methods were evaluated by the highest frame-level accuracy values of all possible permutations of streams [8], [27]. This evaluation method is, however, not applicable to the scenarios where transcribing a specific class of instruments is needed. Therefore, we consider a more rigorous method, where a frame- or note-level prediction result is considered correct (i.e., a true positive) only when all its attributes (i.e., pitch and instrument class for frame-level predictions; pitch, onset, and instrument class for note-level prediction) are all correctly predicted. Here, the term "correctly predicted" means that 1) a pitch is within a half semitone of a ground-truth pitch, 2) its onset is within a ± 50 ms range of the ground-truth note onset, and 3) its instrument class is the same as the ground-truth instrument class. It should be noted that using only 1) is equivalent to the conventional MF0 evaluation, and using 1) and 2) are equivalent to the case of 'onset-only' NT in the MIREX MF0 campaign.[5] Note also that in the MPS and NS cases, a predicted note with correct onset time and pitch would still be identified as a false prediction if its predicted instrument class is incorrect.

The precision, recall, and F-score values are the metrics for all the evaluations. These metrics are computed by counting the number of true positive (TP), false positive (FP), and false negative (FN) over all frames/notes in the test data: $\mathrm{P} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, $\mathrm{R} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$, and $\mathrm{F1} = 2\mathrm{PR}/(\mathrm{P}+\mathrm{R})$. We use the `mir_eval` library to compute these metrics [61].[6] Since the library takes 10 ms as the default time resolution, we apply cubic spline interpolation on the transcription results to change the time resolution from 20 ms to 10 ms. The results are obtained by fine-tuning the output thresholds θ^{on}, θ^{act}, and θ^{ins} on a validation set. The threshold values resulting in the highest F1-scores are adopted. For all the note-level transcription tasks, we report average overlap ratio (AOR). Given t_p^{on} and t_p^{off} are the predicted onset and offset time, and t_g^{on} and t_g^{off} the ground-truth onset and offset time, respectively. The overlap ratio is defined as

$$\text{overlap ratio} := \frac{\min\{t_g^{\mathrm{off}}, t_p^{\mathrm{off}}\} - \max\{t_g^{\mathrm{on}}, t_p^{\mathrm{on}}\}}{\max\{t_g^{\mathrm{off}}, t_p^{\mathrm{off}}\} - \min\{t_g^{\mathrm{on}}, t_p^{\mathrm{on}}\}}\,. \tag{9}$$

[5][Online]. Available: https://www.music-ir.org/mirex/wiki/2020:Multiple_Fundamental_Frequency_Estimation_%26_Tracking

[6][Online]. Available: https://craffel.github.io/mir_eval/

The AOR value of a clip is the average over all the TP notes in that clip. The final results to be reported are therefore the average of the precision, recall, F1-score, and AOR values of all clips in a dataset. For the instrument-wise evaluation, we first compute the precision, recall, F1-score, and AOR for each instrument class in each clip, then report the average of these values over all the clips having that instrument class.

We consider two more metrics that reveal how well a model transcribes instrument information. First, we consider the *standard deviation of F1-scores* (σ_{F1}) over all the instrument classes $\mathcal{S}$ existing in the test set:

$$\sigma_{\mathrm{F1}} := \sqrt{\frac{1}{|\mathcal{S}|-1}\sum_{s\in\mathcal{S}}(\mathrm{F}_s - \bar{\mathrm{F}})^2}\,, \tag{10}$$

where F_s is the F1-score of the instrument class s and $\bar{\mathrm{F}}$ is the mean value of F1-scores over all instrument classes. A large σ_{F1} value implies that the model is biased towards some specific instruments, and a small σ_{F1} implies that the model is fair. Second, the *instrument accuracy* (A_{ins}) is the accuracy of the model in predicting the classes of instruments existing in a music piece. This metric is to measure how well the model performs in the instrument selection process (i.e., select the active output channels) as previously described in Section IV-D. For a music piece with instrument classes $\mathcal{S}_g$ and the predicted instrument classes are $\mathcal{S}_p$, the instrument accuracy is

$$A_{\mathrm{ins}} := \frac{|\mathcal{S}_p \cap \mathcal{S}_g|}{|\mathcal{S}_p \cup \mathcal{S}_g|}\,. \tag{11}$$

VI. Experiment Results

A. Instrument-Agnostic Transcription: MPE and NT

We first evaluate the AMT scenarios which ignore instrument information. Table V lists the MPE and NT results of three settings (i.e., Conv, Conv-LS, and Attn-LS) on a piano solo dataset (i.e., MAPS) and two multi-instrument datasets (i.e., MusicNet and Su-10). Three baseline methods are listed at the bottom of Table V for comparison. First, [15] can be regarded as a simplified version of Conv; it was trained for the MPE task only and was trained on the training set of *Configuration II* in the MAPS dataset [62], rather than on the external MAESTRO dataset. Second, [10] is the Onsets and Frames method, the state-of-the-art method for piano transcription. The third baseline method [25] is a novel CNN-based AMT method operating on raw audio signals. The F1-score of [25] on Su-10 is directly computed from the precision and recall reported in the paper. Note again that the models for single-instrument piano solo AMT and multi-instrument AMT are trained on different datasets (cf. Section V-B).

Table V shows that for the MAPS dataset, the three proposed models have similar F1-scores on MPE. However, on NT, Conv-LS, and the Attn-LS outperform Conv by 10.49% and 8.45%, respectively. This indicates that label smoothing does bring benefit to note-level transcription, probably because onset labels are sparser than pitch activation labels. Improvement using label smoothing can also be observed in multi-instrument datasets. For MusicNet, using label smoothing boosts the MPE performance

TABLE V
RESULTS FOR MULTI-PITCH ESTIMATION (MPE) AND NOTE TRACKING NT

Model	MAPS						MusicNet						Su-10					
	MPE			NT			MPE			NT			MPE			NT		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Conv	71.95	72.82	71.58	70.40	77.43	71.12	55.81	78.74	64.19	55.13	54.38	54.40	60.46	73.89	64.30	51.41	39.16	42.60
Conv-LS	74.11	72.36	72.61	85.03	78.84	81.61	67.50	74.95	70.41	66.66	62.54	64.28	67.55	71.79	**68.99**	52.33	42.17	45.74
Attn-LS	72.88	73.39	72.50	80.10	79.59	79.57	68.26	78.62	72.47	**69.27**	**64.61**	**66.59**	64.72	69.48	65.88	**55.93**	**44.31**	**48.68**
[15]	87.48	**86.29**	**86.73**	–			**69.34**	**79.29**	**73.70**	–			56.43	**77.60**	65.16	–		
[10]	**92.86**	78.46	84.91	**87.46**	**85.58**	**86.44**	–			–			–			–		
[25]	–			–			68.71	77.30	72.75	–			**70.10**	54.60	61.39	–		

TABLE VI
CROSS-DATASET EVALUATION RESULTS. BOLD VALUES REPRESENT THE BEST MODEL FOR EACH DATASET

Model	Data	MPS						NS								A_{ins}
		instrument-informed			instrument-agnostic			instrument-informed				instrument-agnostic				
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	AOR	Prec	Rec	F1	AOR	
Conv	MusicNet	57.27	**73.39**	62.48	40.31	47.52	42.07	53.07	57.03	52.89	40.75	39.51	38.55	37.98	26.88	67.50
	Ext-Su	34.05	**57.65**	40.63	26.86	**42.84**	31.40	18.48	26.84	19.06	41.65	16.01	**19.69**	15.82	29.67	65.46
	URMP-22	42.07	**78.62**	52.91	21.06	35.41	25.91	10.06	16.07	11.69	60.72	5.84	8.20	6.52	28.45	45.61
Conv-LS	MusicNet	64.14	67.18	64.79	48.36	49.06	48.00	59.56	55.51	57.66	**47.03**	45.29	42.93	43.43	33.20	71.67
	Ext-Su	**41.63**	51.49	**44.53**	32.60	38.62	34.63	20.51	**27.03**	20.41	**44.62**	16.91	19.47	16.31	**31.76**	63.85
	URMP-22	**51.07**	66.34	**56.82**	31.17	**36.66**	**33.43**	**14.40**	**20.10**	**16.22**	64.86	**9.04**	**11.40**	**9.81**	35.52	51.21
Attn-LS	MusicNet	**67.11**	70.85	**68.24**	**67.11**	**70.85**	**68.24**	**60.04**	**57.85**	**57.87**	44.22	**60.04**	**57.85**	**57.87**	**44.22**	**100.0**
	Ext-Su	41.47	47.36	43.15	**34.09**	40.13	**36.27**	**25.26**	23.45	**22.46**	39.52	**21.26**	19.17	**19.26**	30.75	**65.73**
	URMP-22	47.10	58.86	51.65	**31.39**	34.97	32.85	12.83	14.27	12.80	**65.79**	7.33	6.99	6.90	**39.73**	**57.42**

by 6.21% and the NT performance by 9.88%, respectively. For Su-10, using label smoothing promotes the MPE performance by 4.69% and the NT performance by 3.14%, respectively. Attn-LS performs better on MusicNet but worse on the other two datasets. It should be noted again that the results presented above are the case of onset-only NT. Our experiment shows that for Attn-LS on MAPS, the onset-offset F1-score is 30.31%, a value much lower than the onset-only F1-score at 79.57% and exhibits the challenge in onset-offset NT.

As looking into the results of the baseline method, it shows that in MPE, our previous methods [15] outperforms [10] on MAPS and [25] on the MusicNet test set. This indicates the effectiveness of Conv, since [15] is also a model based on ASPP and U-net. Besides, Conv-LS outperforms all the other models on the Su-10 dataset. As to the NT task, to the best of our knowledge which is the first time having evaluated on note-level, [10] outperforms others on MAPS, and Attn-LS outperforms others on MusicNet and Su-10. In summary, our proposed settings outperform the baseline methods on the two multi-instrument datasets, and in these evaluations, using ASPP is better on MPE, and using self-attention is better on NT. On the other hand, our settings are less effective on the single-instrument MAPS dataset. [15] achieves high MPE performance as it overfits the MAPS training set. It then becomes less effective when trained on the MAESTRO dataset. Another reason [10] outperforms all is that it is designed only for piano transcription rather than a general-purpose transcription system. A loss function to model the note attack and decay of piano is used in [10].

B. MPS and NS: Cross-Dataset Evaluation

Table VI lists the average precision, recall, F1-score, instrument accuracy, and AOR for instrument-informed MPS and NS, and instrument-agnostic MPS and NS. The model trained on the MusicNet training set is evaluated on the MusicNet test set, Ext-Su, and URMP-22. The evaluation of the latter two datasets can be regarded as cross-dataset evaluation for generalizability testing.

Similar to the results of single-instrument AMT, label smoothing consistently improves the F1-scores and AOR for all the tasks. A major finding in Table VI is that the self-attention mechanism consistently improves the instrument accuracy (A_{ins}) over the other two convolution-based models. Attn-LS even achieves 100% of A_{ins} on the internal test set; this means that the model successfully recognizes the 32 individual instrument labels in the 10 test pieces of the MusicNet test set (see Table IV). This also implies that the instrument-informed and instrument-agnostic scenarios become equivalent and therefore have the same P, R, F1 and AOR. On the contrary, for other cases whose instrument accuracies are mostly lower than 70%, there is a huge gap (i.e., differences of 4–27% in F1-score, differences of 8–32% in AOR) between instrument-informed transcription and instrument-agnostic transcription. Also note that in the NS task, the gap between instrument-informed transcription and instrument-agnostic transcription is smaller than the one in the MPS task, probably because the major bottleneck in the NS task is onset detection rather than instrument recognition.

Table VI also shows that Attn-LS does not improve MPE/ NT performance over Conv-LS but leads to great improvement in instrument accuracy. A possible reason is that in comparison to dilated convolution, self-attention has a wider receptive field. Such a wide receptive field can guide the model to learn the global information, such as timbre, pitch ranges, and arrangement of different instruments. On the other hand, convolution focuses better on local information such as note onsets. In addition, there is also no significant relationship between instrument

TABLE VII
INSTRUMENT-WISE EVALUATION RESULTS ON THE MUSICNET TEST SET

Model	Class	Frame-level MPS						Note-level NS							
		Instrument-informed			Instrument-agnostic			Instrument-informed				Instrument-agnostic			
		Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	AOR	Prec	Rec	F1	AOR
Conv-LS	pn	73.85	58.60	65.29	49.23	39.06	43.53	**76.93**	54.96	68.24	**48.16**	48.20	39.28	42.83	32.10
	vn	62.28	76.75	68.50	41.52	51.16	45.67	60.84	**63.46**	**60.84**	**43.17**	37.22	38.78	37.20	28.78
	va	36.20	52.78	42.90	18.10	26.39	21.45	21.34	**34.75**	26.34	**43.53**	12.73	20.75	15.72	21.76
	vc	56.16	71.32	62.71	26.81	32.48	29.37	**59.58**	**62.54**	**59.13**	**41.70**	27.85	28.93	27.52	19.72
	hn	**57.89**	61.25	59.09	0.00	0.00	–	**11.56**	36.45	**17.52**	**54.53**	0.00	0.00	–	–
	bn	56.62	58.47	57.51	0.00	0.00	–	28.01	32.11	29.91	**55.54**	0.00	0.00	–	–
	cl	58.75	68.83	63.39	0.00	0.00	–	**49.28**	**55.92**	**52.32**	51.80	0.00	0.00	–	–
	σ_{F1}	7.41			18.91			17.74				17.12			
Attn-LS	pn	**77.13**	**64.35**	**70.11**	**77.13**	**64.35**	**70.11**	74.64	**64.69**	**68.93**	45.07	**74.64**	**64.69**	**68.93**	**45.07**
	vn	**65.05**	**79.26**	**71.31**	**65.05**	**79.26**	**71.31**	**61.88**	60.12	60.54	40.95	**61.88**	**60.12**	**60.54**	**40.95**
	va	**41.51**	**60.73**	**49.25**	**41.51**	**60.73**	**49.25**	**28.94**	31.98	**30.07**	37.54	**28.94**	**31.98**	**30.07**	**37.54**
	vc	**60.88**	**71.44**	**65.67**	**60.88**	**71.44**	**65.67**	58.67	44.76	50.35	38.06	**58.67**	**44.76**	**50.35**	**38.06**
	hn	57.38	**69.14**	**62.38**	**57.38**	**69.14**	**62.38**	10.78	**38.11**	16.79	48.57	**10.78**	**38.11**	**16.79**	**48.57**
	bn	**63.91**	**66.53**	**65.19**	**63.91**	**66.53**	**65.19**	**36.63**	**45.63**	**40.60**	52.82	**36.63**	**45.63**	**40.60**	**52.82**
	cl	**63.85**	**68.84**	**66.24**	**63.85**	**68.84**	**66.24**	47.88	55.18	50.98	**52.37**	**47.88**	**55.18**	**50.98**	**52.37**
	σ_{F1}	**6.45**			**6.45**			**16.00**				**16.00**			

accuracy and AOR in the NS tasks. Therefore, the Conv-LS model is still competitive not only in frame-level transcription but also in note-level transcription, due to its effectiveness in note duration estimation. The behaviors of URMP-22 are different from MusicNet and Ext-Su. This is probably because of the dataset mismatch: as the music clips in the URMP dataset are mostly arranged for pose estimation in the video, the note lengths are in general long enough for better pose estimation quality of slow motion.

C. Comparison Over Different Instruments

To further discuss the performance of individual instruments, we focus on comparing Conv-LS and Attn-LS on the MusicNet test set, the only dataset where our model achieves an instrument accuracy of 100%. Table VII lists the results of the seven instrument classes in the MusicNet test set. Unsurprisingly, recognition of the instruments of the majority classes (i.e., piano and violin) achieves high performance in comparison to other classes, but the instrument class with the lowest F1 is not the most rarely seen in the dataset. This will be discussed in Section VI-D. The performance gap between the instrument-informed and instrument-agnostic scenarios exists for all classes. Even, for the NS tasks, Conv-LS fails to recognize the three minority instrument classes (i.e., horn, bassoon, and clarinet); they are missed during the instrument selection process described in Section IV-D2. The notes of these three instruments can be transcribed only when the instrument class information is informed. Such a gap is narrowed down by the Attn-LS model – with the self-attention mechanism, all the instrument channels in the test set are successfully selected. This again reveals the importance of instrument recognition in instrument-level transcription.

The standard deviation of the F1-scores (σ_{F1}) reveals the fairness a model performs in transcribing different instruments. It can be observed that Attn-LS is a model fairer than Conv-LS, as it has lower σ_{F1} values for all the four multi-instrument AMT scenarios. This can also be seen by comparing Conv-LS and Attn-LS on instrument-informed NS; their F1-scores are similar (57.66% and 57.87%, respectively; see Table VI), but Attn-LS has a smaller σ_{F1} value, where the F1-score of bassoon is improved the most (from 29.9% to 40.1%).

TABLE VIII
CONFUSION MATRIX

ground truth \ predicted as (%)	pn	vn	va	vc	hn	bn	cl	unk
pn	72.6	0.1	0.0	0.0	0.0	0.0	0.0	27.3
vn	1.3	60.1	4.1	0.2	0.0	0.0	0.0	34.3
va	0.0	5.0	28.4	12.0	0.0	0.0	0.0	54.5
vc	0.0	0.2	0.4	64.1	0.0	0.0	0.0	35.3
hn	0.0	0.0	0.0	0.0	10.5	13.7	25.3	50.6
bn	0.0	0.0	0.0	0.0	12.4	36.0	3.7	48.0
cl	0.0	0.0	0.0	0.0	12.5	3.9	45.7	38.0

D. An Analysis of the Confusion Matrix

Discussing the confusion of instrument classes in multi-instrument AMT is not a trivial task. In multi-instrument recognition, a falsely detected example always contributes to confusion between two different instrument classes, but in AMT, the relationship between false detection and instrument class confusion is complicated as there are multiple reasons (e.g., a discrepancy in pitch value or onset time) that result in a false detection. Our approach is to discuss the confusion among instrument classes *only* ; for example, if a predicted music note has wrong pitch, onset as well as instrument class, it should be excluded from the discussion of confusion matrix as it introduces other types of error. For every instrument class $s \in \mathcal{S}$, we specify the following three types of predicted notes:

- Type-A: the predicted true positive examples on s.
- Type-B: the predicted false examples which are on $\hat{s} \in \mathcal{S} \backslash s$, but could become true positive if changing their instrument class to s, with their pitches and onsets fixed.
- Type-C: other predicted false examples on s.

We only discuss the confusion among instrument classes contributed by type-B examples, while type-C examples are listed only for reference. Table VIII shows the confusion matrix of the seven instrument classes in the MusicNet test set for the instrument-agnostic NS task with Attn-LS. Rows of Table VIII

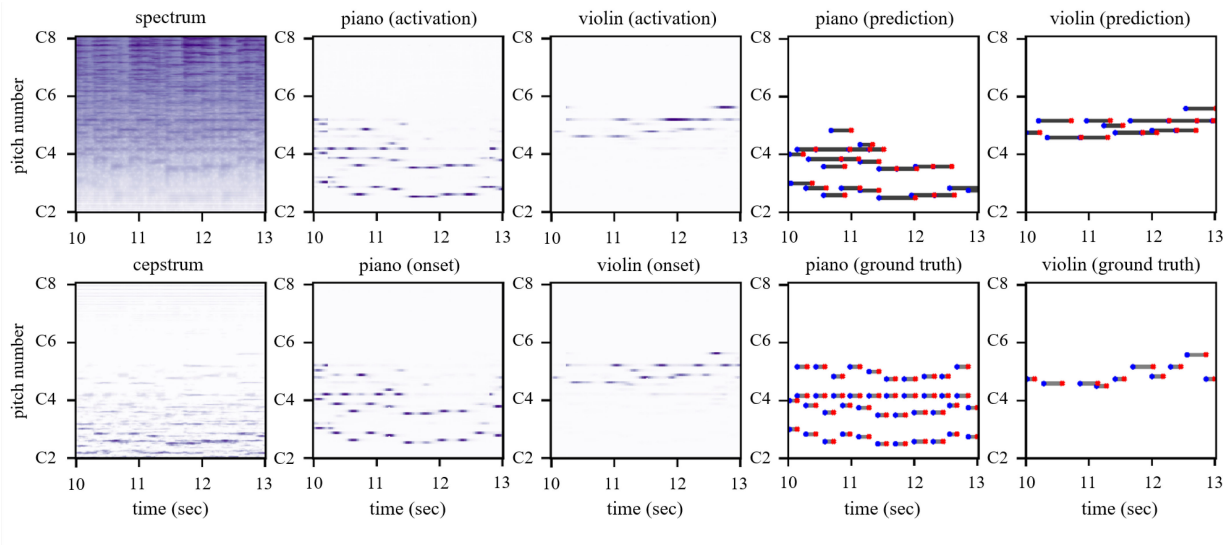

Fig. 5. Illustration of the input data representations, the output of Attn-LS, final output, and ground truth of a segment (10–13 seconds) clipped from '2628.wav' (Beethoven's *Violin Sonata No. 10 in G major*, movement 3, *Scherzo: Allegro - Trio*) in the MusicNet test set. Top row, from left to right: spectrum ($\mathbf{Z}_f$), output activation channel of the piano ($\mathbf{Y}^{\text{act}}_{\text{pn}}$), output activation channel of the violin ($\mathbf{Y}^{\text{act}}_{\text{vn}}$), predicted notes of the piano, predicted notes of the violin. Bottom row, from left to right: cepstrum ($\mathbf{Z}_q$), output onset channel of the piano ($\mathbf{Y}^{\text{on}}_{\text{pn}}$), output onset channel of the violin ($\mathbf{Y}^{\text{on}}_{\text{vn}}$), ground-truth notes of the piano, ground-truth notes of the violin. Dark gray lines: predicted note activation. Light gray lines: ground-truth note activation. Blue dots: note onsets. Red dots: note offsets.

represent the ground-truth instrument classes and columns represent the predicted classes. Type-C notes (denoted by 'unk,' which means 'unknown') are listed in the rightmost column for reference. The percentage of each class is calculated over the sum of type-A, type-B, and type-C notes. For example, the first row is interpreted as: for all notes related to piano based on the above conditions, 72.6% are true positive, 0.1% are predicted as violin but should be predicted as piano, and the remaining 27.3% are predicted as piano but are false because of miscellaneous reasons.

Table VIII reveals how the F1-score of viola is the lowest among all instrument classes though it is not in the extreme minority. First, a certain portion of viola notes are misclassified as the other two similar string instruments, violin and cello (5.0% and 12.0% with respect to 28% true positive notes). Second, 54.5% of the notes on the viola channel are type-C, and this is the highest among all instrument classes. Confusion of instrument classes can also be seen in horn, bassoon, and clarinet. Horn has the greatest portion of type-B notes (13.7% predicted as bassoon, and 25.3% predicted as clarinet) and type-C notes (50.6%) among the three wind instruments. Another way to explain such a phenomenon is to notice that both viola and horn typically serve as the inner parts in ensemble music; these parts are likely to be overlapped to or interwoven with other instruments. To summarize, we identify three factors which affect the performance of individual instruments in multi-instrument AMT: 1) the signal-level similarity to other instruments, 2) the semantic-level relation to other instruments, and 3) the amount of training data.

E. Illustration

Fig. 5 shows the input data representations, the output of Attn-LS, final output, and ground truth of a segment containing piano and violin in the MusicNet test set; see the caption for details. The spectrum and generalized cepstrum are two features canceling each other out: one suppresses harmonics and the other suppresses sub-harmonics. The output channels of the neural network model show the likelihood of onset and activation for the two instruments. It can be seen that the model learns the relationship between onset and activation: onset events appear earlier than activation events. However, the distribution of onset prediction is not very local, probably because the onset labels are not assigned at a single time step, but rather in an interval during training. Comparing the final output note events after post-processing to the ground-truth notes, we observe that the model effectively captures the lowest voice of piano and the predominant melody performed by the violin. On the other hand, the highest voice of the piano is not recognized, where some of the notes in this voice are recognized as the violin. The voice repeating the C4 note of the piano is also not recognized. Moreover, the model constantly overestimates the note duration. These observations clearly indicate the difficulties in recognizing the inner parts of polyphonic music and offset events of notes.

VII. Discussions

In this section, we discuss the critical issues that are found important in the development of a multi-instrument AMT system but are out of the scope of this paper.

The first issue is the perceptual and functional evaluation methodology of multi-instrument AMT. Though a multi-instrument AMT system may open diverse application scenarios, it is unclear whether the traditional metrics such as F1-score and AOR can effectively reveal the performance of the system on the target application. It is worth mentioning that the precision-recall pairs reported in Tables V–VII are highly unbalanced and vary over different settings. An interesting trend is that when fine-tuning the F1-score on the validation set,

note-level transcription tends to have high precision and low recall, while frame-level transcription tends to have high recall and low precision. Such minor difference is however critical in some application scenarios. For example, for the listening experience of sonified transcription results, high precision is preferred as it might eliminate the unwanted notes that annoy the listeners, while for editing experience of transcription results, high recall is sometimes preferred, as deleting unwanted notes might be easier than adding undetected notes for an editor. How and to what extent these can be verified and quantified would require further subjective tests [63] and user studies [64], rather than just comparing the precision and recall values. Another implication is that the transcription of low-level semantics (e.g., frame-level) is still of its own merit in some real-world applications, depending on how it can be effectively combined with high-level transcription results and how the results are evaluated perceptually and functionally.

The second issue is over-fitting. Experimental results have clearly show that [15] overfits the MAPS dataset in comparison to fitting the MAESTRO dataset, and the models proposed in this work also overfit the MusicNet dataset in comparison to fitting the Ext-Su and URMP-22 datasets. Comparing Attn-LS and [15] (cf. Table V), the issue of over-fitting in [15] is less severe in the multi-instrument data (i.e., MusicNet and Su-10) than in the single-instrument data (i.e., MAPS). This confirms the assumption of multi-task learning mentioned in Section I: training on multiple sets of labels could reduce over-fitting on single labels. However, such an assumption does not help well in the multi-label prediction (i.e., MPS and NS) tasks, as severe over-fitting is still observed in cross-dataset evaluation. This might be attributed to the discrepancy of instrument distributions among different datasets (see Table III) and test samples (see Table IV), which contributes to additional dimensions to the mismatch between different datasets. In such cases, using an independent multi-instrument recognition model such as [13] could be a solution to avoid this issue.

The last-but-not-least issue is the quality of training data. In fact, the datasets used in this work are mostly annotated in a semi-automatic way and was not manually checked (and it is impossible to do this). We spot several errors which might introduce misevaluation in onset detection (e.g., ground truth shifted by 50 ms). Since manual labeling is hard, automatically generated training data using the A/S model [65] or weakly-supervised learning [66], [67] should be topics of central concern in the future work of multi-instrument AMT.

VIII. Conclusion

We have explicitly described the generalized problem scenarios of multi-instrument AMT, and describe the relationship among them. Our proposed framework is technically effective on three levels: first, it benchmarks new problem scenarios of specific multi-instrument AMT tasks, such as the instrument-informed and instrument-agnostic note streaming tasks; second, it outperforms the baseline methods on frame-level MPS; and third, its degenerated version is still competitive to other single-instrument AMT methods without any retraining of the model. According to the results of the experiments, we suggest that the CV-based instance segmentation solutions such as image-to-image translation networks with label-balancing focal loss and label smoothing are highly suited to multi-instrument AMT. On the other hand, our investigation shows that multi-instrument AMT is still a challenging problem by far, especially on the issues of over-fitting, quality of annotation, and evaluation methodology on perceptual and functional aspects. Solving these issues is a critical step to leverage the AMT technology into real-world applications.

References

[1] 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.

[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Challenges and future directions," *J. Intell. Inf. Syst.*, vol. 41, pp. 407–434, Jul. 2013.

[3] B. Emmanouil, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: Breaking the glass ceiling," in *Proc. 13th Int. Soc. Music Inf. Retrieval Conf.*, 2012, pp. 379–384.

[4] Li Su and Yi-H. Yang, "Escaping from the abyss of manual annotation: New methodology of building polyphonic datasets for automatic music transcription," in *Proc. Int. Symp. Comput. Music Multidisciplinary Res. (CMMR)*, 2015, pp. 309–321.

[5] E. Benetos and S. Dixon, "Polyphonic music transcription using note onset and offset detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 37–40.

[6] S. Chang and K. Lee, "A pairwise approach to simultaneous onset/offset detection for singing voice using correntropy," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 629–633.

[7] L. Gao, Li Su, Yi-Hsuan 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.*, 2017, pp. 291–295.

[8] G. Grindlay and D. P. Ellis, "Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1159–1169, Oct. 2011.

[9] Y. Zhang and Q. Yang, "An overview of multi-task learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 30–43, Jan. 2018.

[10] C. Hawthorne *et al.*, "Onsets and frames: Dual-objective piano transcription," in *Proc. 19th Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 50–57.

[11] C. Hawthorne *et al.*, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *Proc. 7th Int. Conf. Learn. Representations*, 2019, pp. 1–10.

[12] C.-Z. A. Huang *et al.*, "Music transformer: Generating music with long-term structure," in *Proc. 7th Int. Conf. Learn. Representations*, 2019, pp. 1–11.

[13] Y.-N. Hung and Y.-H. Yang, "Frame-level instrument recognition by timbre and pitch," in *Proc. 19th Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 135–142.

[14] Y.-N. Hung, Yi-An Chen, and Yi-Hsuan Yang, "Multitask learning for frame-level instrument recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 381–385.

[15] Yu-Te Wu, B. Chen, and Li Su, "Polyphonic music transcription with semantic segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 166–170.

[16] C.-Yi Kuan, L. Su, Y.-H. Chin, and J.-C. Wang, "Multi-pitch streaming of interwoven streams," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 311–315.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Berlin, Germany: Springer, 2015, pp. 234–241.

[18] A. Jansson, E. J. Humphrey, N. Montecchio, R. M. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep U-net convolutional networks," in *Proc. 18th Int. Soc. Music Inf. Retrieval Conf.*, 2017, pp. 745–751.

[19] W.-T. Lu and Li Su, "Vocal melody extraction with semantic segmentation and audio-symbolic domain transfer learning," in *Proc. 19th Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 521–528.

[20] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for F0 estimation in polyphonic music," in *Proc. 18th Int. Soc. Music Inf. Retrieval Conf.*, 2017, pp. 63–70.

[21] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Comput. Vis. - ECCV - 13th Eur. Conf., Proc., Part V*, 2014, pp. 740–755.

[22] N. Parmar *et al.*, "Image transformer," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 4052–4061.

[23] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 8, pp. 2121–2133, Dec. 2010.

[24] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Music genre database and musical instrument sound database," in *Proc. 4th Int. Conf. Music Inf. Retrieval, Proc.*, 2003, pp. 229–230.

[25] J. Thickstun, Z. Harchaoui, D. P. Foster, and S. M. Kakade, "Invariances and data augmentation for supervised music transcription," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 2241–2245.

[26] E. Manilow, G. Wichern, P. Seetharaman, and J. L. Roux, "Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2019, pp. 45–49.

[27] Z. Duan, J. Han, and B. Pardo, "Multi-pitch streaming of harmonic sound mixtures," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 1, pp. 138–150, Jan. 2014.

[28] V. Arora and L. Behera, "Musical source clustering and identification in polyphonic audio," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 6, pp. 1003–1012, Jun. 2014.

[29] V. Arora and L. Behera, "Multiple F0 estimation and source clustering of polyphonic music audio using plca and HMRFs," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 2, pp. 278–287, Feb. 2015.

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

[31] E. Benetos and S. Dixon, "A shift-invariant latent variable model for automatic music transcription," *Comput. Music J.*, vol. 36, pp. 81–94, 2012.

[32] B. Fuentes, R. Badeau, and G. Richard, "Harmonic adaptive latent component analysis of audio and application to music transcription," *IEEE Trans. Audio, Speech, Lang Process.*, vol. 21, no. 9, pp. 1854–1866, Sep. 2013.

[33] E. Vincent and X. Rodet, "Music Transcription with ISA and HMM," in *Proc. Independent Component Anal. Blind Signal Separation, 5th Int. Conf.*, 2004, pp. 1197–1204.

[34] H. Kameoka, T. Nishimoto, and S. Sagayama, "A multipitch analyzer based on harmonic temporal structured clustering," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 982–994, Mar. 2007.

[35] W.-C. Chang, A. WY Su, C. Yeh, A. Roebel, and X. Rodet, "Multiple-F0 tracking based on a high-order HMM model," in *Proc. 11th Int. Conf. Digit. Audio Effects*, 2008, p. 1.

[36] H. Kirchhoff, S. Dixon, and A. Klapuri, "Multiple instrument tracking based on reconstruction error, pitch continuity and instrument activity," in *Proc. Int. Symp. Comput. Music Multidisciplinary Res.*, 2013, pp. 894–903.

[37] E. Benetos and S. Dixon, "Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model," *J. Acoust. Soc. Amer.*, vol. 133, pp. 1727–1741, 2013.

[38] E. Benetos, S. Ewert, and T. Weyde, "Automatic transcription of pitched and unpitched sounds from polyphonic music," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 3107–3111.

[39] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "MedleyDB: A multitrack dataset for annotation-intensive MIR research," in *Proc. 15th Int. Soc. Music Inf. Retrieval Conf.*, 2014, pp. 155–160.

[40] E. Humphrey, S. Durand, and B. McFee, "Openmic-2018: An open dataset for multiple instrument recognition," in *Proc. 19th Int. Soc. for Music Inf. Retrieval Conf.*, 2018, pp. 438–444.

[41] S. Gururani and A. Lerch, "Mixing secrets: A multi-track dataset for instrument recognition in polyphonic music," in *Proc. 18th Int. Soc. Music Inf. Retrieval Conf. (Late Breaking and Demo Paper)*, 2017, pp. 1–2.

[42] J. J. Bosch, J. Janer, F. Fuhrmann, and P. Herrera, "A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals," in *Proc. 13th Int. Soc. Music Inf. Retrieval Conf.*, 2012, pp. 559–564.

[43] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 1, pp. 208–221, Jan. 2017.

[44] S. Gururani, C. Summers, and A. Lerch, "Instrument activity detection in polyphonic music using deep neural networks," in *Proc. 19th Int. Soc. Music Inf. Retrieval Conf.*, 2018, pp. 569–576.

[45] R. Shankar, "Instrument Identification in Polyphonic Music," Ph.D. dissertation, New York University, New York, NY, USA, 2018.

[46] Yu-Te Wu, B. Chen, and Li Su, "Automatic music transcription leveraging generalized cepstral features and deep learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 401–405.

[47] Li Su and Y.-H. Yang, "Combining spectral and temporal representations for multipitch estimation of polyphonic music," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 10, pp. 1600–1612, Oct. 2015.

[48] T. Kobayashi and S. Imai, "Spectral analysis using generalized cepstrum," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 6, pp. 1235–1238, Oct. 1984.

[49] S. Li, T.-Y. Chuang, and Yi-H. Yang, "Exploiting frequency, periodicity and harmonicity using advanced time-frequency concentration techniques for multipitch estimation of choir and symphony," in *Proc. 17th Int. Soc. Music Inf. Retrieval Conf.*, 2016, pp. 393–399.

[50] G. Peeters, "Music pitch representation by periodicity measures based on combined temporal and spectral representations," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2006, pp. 53–56.

[51] C.-Y. Yu and Li Su, "Multi-layered cepstrum for instantaneous frequency estimation," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, 2018, pp. 276–280.

[52] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*, Accessed: Oct. 18, 2020.

[53] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. 30: Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[54] T.-Y. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2999–3007.

[55] M. Goibert and E. Dohmatob, "Adversarial robustness via adversarial label-smoothing," 2019, *arXiv:1906.11567*, Accessed: Oct. 18, 2020.

[56] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?," in *Proc. Adv. Neural Inf. Process. Syst. 32: Annu. Conf. Neural Inf. Process. Syst.*, 2019, pp. 4696–4705.

[57] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

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

[59] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning features of music from scratch," in *Proc. 5th Int. Conf. Learn. Representations*, 2017, pp. 1–10.

[60] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 522–535, Feb. 2019.

[61] C. Raffel *et al.*, "Mir_eval: A transparent implementation of common MIR metrics," in *Proc. 15th Int. Soc. Music Inf. Retrieval Conf.*, 2014, pp. 367–372.

[62] 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. Retrieval Conf.*, 2016, pp. 475–481.

[63] A. Cogliati, D. Temperley, and Z. Duan, "Transcribing human piano performances into music notation," in *Proc. 17th Int. Soc. Music Inf. Retrieval Conf.*, 2016, pp. 758–764.

[64] A. Holzapfel and E. Benetos, "Automatic music transcription and ethnomusicology: A user study," in *Proc. 20th Int. Soc. Music Inf. Retrieval Conf.*, 2019, pp. 678–684.

[65] J. Salamon, R. M. Bittner, J. Bonada, J. J. Bosch, E. Gómez, and J. P. Bello, "An analysis/synthesis framework for automatic F0 annotation of multitrack datasets," in *Proc. 18th Int. Soc. Music Inf. Retrieval Conf.*, 2017, pp. 71–78.

[66] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirović, "SPICE: Self-supervised pitch estimation," *IEEE/ACM Trans. Audio, Speech, Lang. Proc.*, vol. 28, pp. 1118–1128, Mar. 2020.

[67] R. Nishikimi, E. Nakamura, S. Fukayama, M. Goto, and K. Yoshii, "Automatic singing transcription based on encoder-decoder recurrent neural networks with a weakly-supervised attention mechanism," in *IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 161–165.

Yu-Te Wu received the B.S. and M.S. degrees in computer science and information engineering from National Taiwan Normal University, Taipei, Taiwan, in 2018 and 2020, respectively. He has been a Research Student with the Institute of Information Science, Academia Sinica, Taipei, Taiwan, since 2017. He is currently a Research Assistant with Music and Culture Technology Lab, Institute of Information Science, Academia Sinica. His research interests include automatic music transcription and deep learning.

Berlin Chen (Member, IEEE) received the B.S. and M.S. degrees in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1994 and 1996, respectively, and the Ph.D. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 2001. He was with the Institute of Information Science, Academia Sinica, Taipei, from 1996 to 2001, and then with the Graduate Institute of Communication Engineering, National Taiwan University, from 2001 to 2002. In 2002, he joined the Graduate Institute of Computer Science and Information Engineering, National Taiwan Normal University, Taipei, Taiwan. He is currently a Professor with the Department of Computer Science and Information Engineering, National Taiwan Normal University. His research interests generally lie in the areas of speech recognition, automatic summarization, information retrieval, and music processing. He is the author or coauthor of more than 200 academic publications.

Li Su (Member, IEEE) was born in Kaohsiung City, Taiwan. He received the B.S. degrees (double) in electronic engineering and mathematics from National Taiwan University, Taipei, Taiwan, in 2008, and the Ph.D. degree in communication engineering from National Taiwan University, Taipei, Taiwan, in 2012. Since then, he had served as a Postdoctoral Research Fellow with the Center of Information and Technology Innovation, Academia Sinica. Since 2017, he has served as an Assistant Research Fellow with the Institute of Information Science, Academia Sinica, and hosted the Music and Cultural Technology Lab. He also lectures with Taiwan AI Academy, National Tsing Hua University, Taiwan International Graduate Program (TIGP), and the Data Science Degree Program of National Taiwan University. He has served as a Program Committee Member of the International Society of Music Information Retrieval (ISMIR) since 2014. His research interests include audio signal processing, automatic recognition and analysis of musical signals, automatic generation of multimedia contents, and interactive virtual musician systems. He has authored or coauthored more than 50 papers in refereed journals and international conferences of music information retrieval and multimedia. He was an awardee of the Best Paper Award in the ISMIR conference in 2019.