Adaptive Scattering Transforms for Playing Technique Recognition

Changhong Wang, Emmanouil Benetos, Vincent Lostanlen, Elaine Chew

Adaptive Scattering Transforms for Playing Technique Recognition

Changhong Wang, *Student Member, IEEE,* Emmanouil Benetos, *Senior Member, IEEE*
Vincent Lostanlen and Elaine Chew

***Abstract*—Playing techniques contain distinctive information about musical expressivity and interpretation. Yet, current research in music signal analysis suffers from a scarcity of computational models for playing techniques, especially in the context of live performance. To address this problem, our paper develops a general framework for playing technique recognition. We propose the adaptive scattering transform, which refers to any scattering transform that includes a stage of data-driven dimensionality reduction over at least one of its wavelet variables, for representing playing techniques. Two adaptive scattering features are presented: frequency-adaptive scattering and direction-adaptive scattering. We analyse seven playing techniques: vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando. To evaluate the proposed methodology, we create a new dataset containing full-length Chinese bamboo flute performances (CBF-dataset) with expert playing technique annotations. Once trained on the proposed scattering representations, a support vector classifier achieves state-of-the-art results. We provide explanatory visualisations of scattering coefficients for each technique and verify the system over three additional datasets with various instrumental and vocal techniques: VPset, SOL, and VocalSet.**

***Index Terms*—Music signal analysis, music performance analysis, scattering transform.**

I. Introduction

Performance analysis plays a crucial role in music information retrieval (MIR) and presents valuable information for applications such as genre classification, performance style recognition, and performer identification. A typical example of expressive music performance is the application of playing techniques, such as vibratos and tremolos. The modelling and detection of playing techniques find potential applications in the automatic transcription of musical ornaments [1], realistic music generation [2], computer-aided music pedagogy [3], instrument classification [4], [5], and performance analysis [6].

In this article, we propose a general framework based on the scattering transform [7] for playing technique recognition in music signals. The approach is motivated by the observation, when displaying playing techniques in the time–frequency domain, that each technique has a distinctive spectro-temporal pattern. Due to the physical characteristics of certain instruments, some playing techniques are instrument-specific. We focus on seven commonly-used playing techniques in music signals: vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando. Fig. 1 shows the examples of these playing techniques on the Chinese bamboo flute (CBF). The four playing techniques in Fig. 1 (a)—vibrato, tremolo, trill, and flutter-tongue—are periodic modulations that elaborate on stable notes and are temporally symmetric. The modulation patterns in their harmonic partials move in parallel. The difference between them exists in the rate, frequency depth, and shape of the modulations. We refer to these playing techniques as *pitch modulation-based techniques* (PMTs).

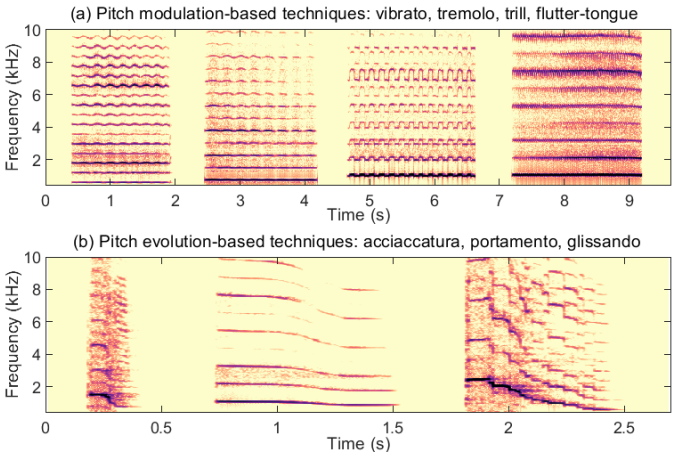

Fig. 1. Spectrograms of commonly-used playing techniques in music signals: (a) pitch modulation-based techniques (PMTs) and (b) pitch evolution-based techniques (PETs).

However, as with playing techniques containing monotonic pitch changes, finding an appropriate representation that captures discriminative information is a challenging task. Fig. 1 (b) shows three examples from this group of playing techniques: acciaccatura, portamento, and glissando. For the CBF, these playing techniques are known as 垛音 (duoyin), 滑音 (huayin), and 历音 (liyin), respectively. Acciaccatura includes a sharp attack and strong air flow on the first note followed by a rapid transition to the second note, and is a characteristic CBF playing technique. Portamento is a continuous slide between two notes. Glissando is a slide across a series of discrete tones. We call this group of playing techniques *pitch evolution-based techniques* (PETs), because they contain monotonic pitch changes over time and are temporally asymmetric.

Patterns of regularity within each playing technique family, PMTs or PETs, motivate us to build a general-purpose model for playing technique recognition. The representation should be stable to local time-shifts, time-warps, and frequency-transpositions. We find that the *scattering transform* [7], a signal representation with many successful applications [8]–[10], provides such invariance properties with mathematical

C. Wang and E. Benetos are with the Centre for Digital Music, School of Electronic Engineering and Computer Science, Queen May University of London, UK (emails: changhong.wang; emmanouil.benetos@qmul.ac.uk). V. Lostanlen is with CNRS–UMR6004/LS2N, Nantes, France (email: vincent.lostanlen@ls2n.fr). E. Chew is with CNRS–UMR9912/STMS (IRCAM), Paris, France (email: elaine.chew@ircam.fr).

guarantees. Specifically within the scattering framework, the time–frequency scattering [11] applies frequency scattering along the log-frequency axis. This operation offers frequency-transposition invariance and captures spectral regularities, besides the local invariance to translation and stability to deformation of the standard scattering transform.

This paper builds upon two conference papers, [12] and [13], and extends them in three directions: a broader literature review on computational playing technique analysis; mathematical definitions of the adaptive scattering transforms; and an evaluation of the proposed method on existing datasets with a variety of instrumental and vocal techniques. Overall, this paper presents contributions from three aspects:

- **Representation**: we propose a new branch of the scattering transform, the *adaptive scattering*, for representing playing techniques. We define the adaptive scattering as any scattering transform that includes a stage of data-driven dimensionality reduction over at least one of its wavelet variables. Two adaptive scattering representations, the frequency-adaptive scattering and the direction-adaptive joint time–frequency scattering (dJTFS), are introduced and are mathematically defined. Both representations are locally invariant to time-shifts, time-warps, frequency-transpositions, and time-reversal of the signal. The frequency-adaptive scattering [12] represents PMTs by calculating the second-order transform adaptively around the frequency band with maximum acoustic energy. In contrast, the dJTFS [13] captures PET patterns by extracting the joint time–frequency scattering via a pooling operation over its direction variable.
- **Dataset**: we publicly release a new dataset for computational analysis of playing techniques in a performance context. The dataset, called CBFdataset, is the first dataset on the Chinese bamboo flute, comprising full-length performances and playing technique annotations by musical experts. The data collection process takes into account the diversity of performers, flute types, pieces, and styles.
- **Application**: we develop a supervised learning system for detecting and classifying playing techniques, which is robust to frequency-transpositions, variations in instruments, performers, and regional musical styles. Using the proposed representations as input, we evaluate the system on different datasets with a variety of playing techniques. We provide a formal interpretation of the role of each component in the proposed scattering transform feature extractor, confirmed by explanatory visualisations.

The paper is organised as follows: Sections II and III review related work and introduce the fundamentals of the scattering transform, respectively. Sections IV and V describe the characteristics of PMTs and PETs, respectively, and demonstrate how the proposed representations capture these characteristics. Section VI presents the recognition system. Evaluation and recognition results are provided in Sections VII and VIII, respectively. Section IX discusses the strengths, weaknesses, and possible applications of the system, followed by the conclusions in Section X.

II. Related Work

Due to the annotation-intensive nature and scarcity of playing techniques in real-world performances, prior computational research on playing techniques was typically instrument- or technique-specific, or focused on playing techniques recorded in highly controlled environments. We summarise existing research from three fronts below, and provide a complete list of playing techniques analysed and the corresponding methodologies applied in the supplementary material [14].

Instruments: Prior research has focused mainly on Western instruments. Guitar playing techniques were most frequently explored [15]–[21]. These techniques were commonly categorised by the active hand: expression-style (left-hand) and plucking-style (right-hand). Piano technique recognition only included trills [22] and pedalling techniques [23]. Playing technique analysis on other Western instruments covered violin [24], [25], drums [26], [27], cello [28], Irish flute [29] and the highland pipe [30]. Non-Western instruments studied included erhu [6], guqin [31], ney [32], and the CBF [33]. Due to their similarity with instrumental playing techniques, we also include vocal techniques [34], [35] for completeness.

Playing techniques: While a small number of playing techniques are shared across instruments, such as vibrato, tremolo, and trill, many others are instrument-specific. For example, pedalling techniques are part of piano playing while airflow techniques are not. We thus narrow the scope to playing techniques that are relevant to multiple types of instruments: vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando.

Methodologies: Early research on playing technique recognition often fed a large set of features, such as the fundamental frequency (F0), mel-frequency cepstral coefficients (MFCCs), and spectral flux, to machine learning classifiers [20], [21]. Other methodologies focused on specific playing techniques to explicitly incorporate prior knowledge. The filter diagonalisation method (FDM), which efficiently extracts high resolution spectral information for short time signals, was first applied to vibrato detection in erhu performances [6]. It was based on the F0 estimated by pYIN [36], an error-prone stage prior to detection for some instruments. The auditory temporal modulations (ATM) [37] and modulation power spectrum (MPS) [38] are representations which capture temporal and spectro-temporal modulation information, respectively, in audio signals. Typical applications of ATM and MPS include genre classification [37] and instrument recognition [38] but both representations have not yet been used for playing technique recognition. Hidden Markov models (HMMs) were used in [33] for detecting CBF glissando and in [6] to recognise erhu portamento. There is not yet any general framework that explicitly recognises multiple types of playing techniques in real-world music performances.

III. Scattering Transform

In this section, we introduce the scattering transform and provide an overview of different scattering operators. Proposed in [7], the scattering transform has the structure of a convolutional neural network (CNN): both comprise a cascade of convolutions, nonlinearities, and pooling operations. The

difference is that the filters of the scattering transform are not learnt but defined as wavelets.

Fig. 2 displays the cascading of scattering operations with the example of a musical trill. Let $\boldsymbol{x}(t)$ be an audio waveform and $\boldsymbol{\psi}_{\lambda_k}(t)$ with $k \in \mathbb{N}$ the wavelet filterbank at the kth-order scattering decomposition. $t \in \mathbb{R}$ is the time variable and $\lambda_k \in \mathbb{R}$ is the log-frequency variable of $\boldsymbol{\psi}_{\lambda_k}(t)$. Here, an "order" of the scattering transform is analogous to a "layer" in terms of CNNs. Take the first order for instance: by convolving $\boldsymbol{x}(t)$ with each wavelet in $\boldsymbol{\psi}_{\lambda_1}(t)$ and applying complex modulus, we obtain the first-order wavelet modulus transform $\mathbf{U}_1\boldsymbol{x}(t, \lambda_1)$, also known as scalogram. Note that $\mathbf{U}_1\boldsymbol{x}(t, \lambda_1)$ is stable to small deformations but not translation-invariant. The scattering transform aims at an invariance property up to some time structure T by average-pooling, i.e. applying to each frequency band in $\mathbf{U}_1\boldsymbol{x}(t, \lambda_1)$ a lowpass filter $\boldsymbol{\phi}_T(t)$ of a cutoff frequency T^{-1}, which results in the first-order scattering transform $\mathbf{S}_1\boldsymbol{x}(t, \lambda_1)$. Cascading the operations of wavelet convolutions with $\boldsymbol{\psi}_{\lambda_k}(t)$ and complex modulus generates a "scattering network", after which the lowpass filtering of the kth-order wavelet modulus transform $\mathbf{U}_k\boldsymbol{x}(t, \lambda_k)$ by $\boldsymbol{\phi}_T(t)$ yields the kth-order scattering transform $\mathbf{S}_k\boldsymbol{x}(t, \lambda_k)$. For completeness, we also extract the zeroth-order scattering transform $\mathbf{S}_0\boldsymbol{x}(t)$ by convolving $\boldsymbol{x}(t)$ with $\boldsymbol{\phi}_T(t)$.

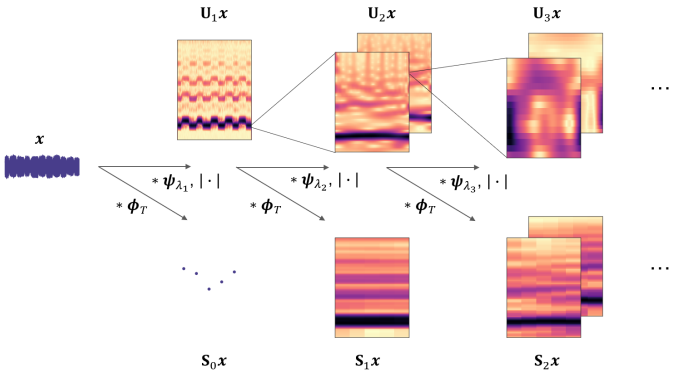

Fig. 2. Diagram of the scattering transform for a trill example. Convolving waveform $\boldsymbol{x}$ with wavelet filterbank $\boldsymbol{\psi}_{\lambda_1}$ and taking complex modulus yields the first-order wavelet modulus transform $\mathbf{U}_1\boldsymbol{x}$. Averaging $\mathbf{U}_1\boldsymbol{x}$ by lowpass filter $\boldsymbol{\phi}_T$ results in the first-order scattering transform $\mathbf{S}_1\boldsymbol{x}$. Cascading these operations, i.e. convolving with $\boldsymbol{\psi}_{\lambda_k}$ ($k \in \mathbb{N}$), taking complex modulus, and averaging by $\boldsymbol{\phi}_T$, generates the kth-order wavelet modulus transform $\mathbf{U}_k\boldsymbol{x}$ and scattering transform $\mathbf{S}_k\boldsymbol{x}$, forming a "scattering network".

Previous studies demonstrated empirically that, for T below 1.5 s, the first- and second-order scattering transform absorb the majority of the signal energy [39]; thus this paper focuses on the scattering transform at these two orders only. For simplicity, we denote the log-frequency variables of the wavelet filterbanks at the first and second order as λ and v_t, replacing λ_1 and λ_2 above. The corresponding wavelet filterbanks are then $\boldsymbol{\psi}_\lambda(t)$ and $\boldsymbol{\psi}_{v_\mathrm{t}}(t)$. $\boldsymbol{\psi}_\lambda(t)$ is obtained by dilation of a "mother wavelet" $\boldsymbol{\psi}(t)$ with a scaling factor $2^{-\lambda}$, yielding:

$$\boldsymbol{\psi}_\lambda(t) = 2^\lambda \boldsymbol{\psi}\big(2^\lambda t\big), \tag{1}$$

and likewise at the second order, $\boldsymbol{\psi}_{v_\mathrm{t}}(t)$ is generated by replacing λ with v_t in Eq. (1). We also use the notation $\mathbf{X}(t, \lambda)$ as a shorthand for the scalogram of the waveform $\boldsymbol{x}(t)$:

$$\mathbf{X}(t, \lambda) = \mathbf{U}_1\boldsymbol{x}(t, \lambda) = \big|\boldsymbol{x} * \boldsymbol{\psi}_\lambda\big|(t). \tag{2}$$

After averaging $\mathbf{X}(t, \lambda)$ along the time axis by a lowpass filter $\boldsymbol{\phi}_T(t)$, we obtain the *first-order scattering transform* [7]:

$$\mathbf{S}_1\boldsymbol{x}(t, \lambda) = \Big(\big|\boldsymbol{x} * \boldsymbol{\psi}_\lambda\big| * \boldsymbol{\phi}_T\Big)(t), \tag{3}$$

which is locally invariant to time-shifting and time-warping.

Similarly, we decompose each frequency band of $\mathbf{X}(t, \lambda)$ by another wavelet filterbank $\boldsymbol{\psi}_{v_\mathrm{t}}(t)$. We denote the log-frequency variable of this filterbank by v_t, where the subscript t signifies that it captures the temporal variation of the scalogram. After taking complex modulus and local averaging, we then obtain the *second-order scattering transform* [7]:

$$\mathbf{S}_2\boldsymbol{x}(t, \lambda, v_\mathrm{t}) = \Big(\big|\mathbf{X} \overset{t}{*} \boldsymbol{\psi}_{v_\mathrm{t}}\big| \overset{t}{*} \boldsymbol{\phi}_T\Big)(t, \lambda), \tag{4}$$

where the symbol $\overset{t}{*}$ denotes a one-dimensional (1-D) convolution over the time variable t. When applied to the two-dimensional (2-D) scalogram $\mathbf{X}(t, \lambda)$, this 1-D convolution is implicitly broadcast over the variable λ.

To capture only the temporal variation regardless of the absolute energy of the waveform, we normalise the second-order coefficients $\mathbf{S}_2\boldsymbol{x}(t, \lambda, v_\mathrm{t})$ over the first-order coefficients $\mathbf{S}_1\boldsymbol{x}(t, \lambda)$. Motivated by auditory perception, the logarithm is applied to the normalised coefficients [39]. The *log-normalised second-order scattering transform* is expressed as [39]:

$$\widetilde{\mathbf{S}}_2\boldsymbol{x}(t, \lambda, v_\mathrm{t}) = \log_2\left(\frac{\mathbf{S}_2\boldsymbol{x}(t, \lambda, v_\mathrm{t})}{\mathbf{S}_1\boldsymbol{x}(t, \lambda) + \varepsilon}\right), \tag{5}$$

where $\varepsilon > 0$ is a small additive that avoids division by zero.

Note that the above convolutions are carried out in the time domain only. Thus, we call $\mathbf{S}_1\boldsymbol{x}(t, \lambda)$ and $\mathbf{S}_2\boldsymbol{x}(t, \lambda, v_\mathrm{t})$ the first- and second-order *time scattering* (or *standard scattering*) coefficients, respectively. Similar to CNNs which may have horizontal and vertical filters [40], one may apply wavelet convolutions along the frequency axis of a given time–frequency representation. The different ways of applying wavelet convolutions form different scattering operators, as shown in Fig. 3. Each operator captures a specific signal pattern, thus making the scattering transform a flexible framework for different music signal analysis tasks.

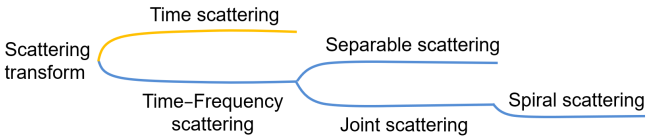

Fig. 3. Relationship between different operators in the scattering transform framework.

The time scattering captures the long-term temporal structure of the signal. The *separable scattering* [41] and *joint scattering* [11] are different instances of the time–frequency scattering that apply wavelet convolutions along both the time and the log-frequency axes. Besides these two convolution operations, the *spiral scattering* [42] adds a wavelet convolution across octaves to capture harmonic variations. In this

paper, we propose the adaptive scattering, which is a new branch of the scattering transform, apart from the time and the time–frequency scattering shown in Fig. 3. In the following sections, we introduce two instances of the adaptive scattering: the frequency-adaptive scattering and the direction-adaptive scattering. Hereafter, we use Morlet wavelets throughout the whole scattering framework for wavelet convolutions. This is because Morlet wavelets have an exactly null average while reaching a quasi-optimal tradeoff in time–frequency localisation [43]. Our source code is based on the ScatNet toolbox[1] and is publicly available for reproducibility at `c4dm.eecs.qmul.ac.uk/CBFdataset.html`.

IV. Frequency-adaptive Scattering for Pitch Modulation-based Techniques

A. Characteristics of Pitch Modulation-based Techniques

Although PMTs all result in some periodic modulations in the time–frequency domain, each type of PMT has distinct characteristics, as listed in Table I. The extent and shape characteristics are based on music theory and the rate information is summarised from the CBFdataset (see Section VII-A1). Flutter-tongue has a much higher modulation rate as compared to the other three modulations. For the other three techniques with similar modulation rates, the discriminative information lies in the modulation extent and shape of the modulation unit. The *modulation unit* refers to the unit pattern that repeats periodically within the modulation. It can be either an amplitude modulation (AM), a frequency modulation (FM), or a spectro-temporal modulation. This can be intuitively observed from the spectrograms given in Fig. 4. Trills are note-level modulations, for which the frequency variations are at least one semitone. This extent of modulation is much larger than that of vibratos and tremolos. The shape of the modulation unit for trills is more square-like than vibratos' sinusoidal form. The difference between vibrato and tremolo is that vibratos are FMs, while tremolos are AMs. We show later how this discriminative information is captured by the proposed frequency-adaptive scattering representations in Section IV-C.

TABLE I
Characteristics of Pitch Modulation-based Techniques

Type	Rate (Hz)	Extent	Shape
Flutter-tongue	25-50	$<$ 1 semitone	Sawtooth-like
Vibrato	3-10	$<$ 1 semitone	Sinusoidal (FM)
Tremolo	3-8	$\approx$ 0 semitone	Sinusoidal (AM)
Trill	3-10	Note level	Square-like

B. Frequency-adaptive Scattering

Due to the harmonic nature of PMTs, one harmonic partial sufficiently captures all the characteristic information: rate, extent, and shape. Therefore, we propose the *frequency-adaptive scattering* for representing PMTs. Instead of decomposing all frequency bands of the scalogram, the frequency-adaptive scattering calculates the second-order transform adaptively around the dominant frequency band, the band with maximum acoustic energy. This provides representations that are highly-compact and are invariant to large frequency-transpositions as compared to the standard scattering described in Section III.

[1]https://www.di.ens.fr/data/software/scatnet

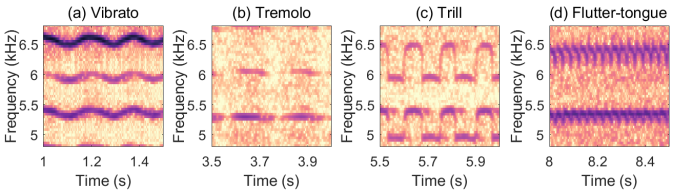

Fig. 4. Visual comparison of PMT characteristics for (a) vibrato, (b) tremolo, (c) trill, and (d) flutter-tongue, in the time–frequency domain (partially enlarged from Fig. 1 (a)).

From the first-order scattering coefficients $\mathbf{S}_1\boldsymbol{x}(t,\lambda)$ shown in Fig. 5 (b), we extract the frame-wise index of the frequency band with maximum acoustic energy:

$$\lambda_{\max}(t) = \arg\max_{\lambda} \Big(\mathbf{S}_1\boldsymbol{x}(t,\lambda)\Big). \tag{6}$$

This forms the *dominant band trajectory*, a 1-D time series. PMTs are spectro-temporal patterns normally spread over several frequency bands. To extract information of the full modulation pattern, we introduce an L-band tolerance symmetrically centred at the dominant band trajectory. L is the total number of frequency bands decomposed. We then define the *decomposition trajectory* as:

$$\Lambda(t) = \left\{\lambda_{\max}(t) + l \,\middle|\, -\frac{L}{2} \leq l \leq \frac{L}{2}\right\}. \tag{7}$$

We locate the L-band decomposition trajectory of the scalogram (Fig. 5 (a)) by expressing its log-frequency axis in local coordinates with respect to the dominant band trajectory:

$$\mathbf{X}_{\Lambda}(t,l) = \mathbf{X}\big(t,\Lambda\big). \tag{8}$$

Then, we define the *frequency-adaptive time scattering* (AdaTS) by convolving $\mathbf{X}_{\Lambda}(t,l)$ with all wavelets in $\boldsymbol{\psi}_{v_\mathrm{t}}(t)$, applying complex modulus, and averaging locally with $\boldsymbol{\phi}_T(t)$:

$$\mathbf{S}_2^{\mathrm{AdaTS}}\boldsymbol{x}(t,l,v_\mathrm{t}) = \Big(\big|\mathbf{X}_{\Lambda} \overset{t}{*} \boldsymbol{\psi}_{v_\mathrm{t}}\big| \overset{t}{*} \boldsymbol{\phi}_T\Big)(t,l). \tag{9}$$

In the equation above, $\mathbf{S}_2^{\mathrm{AdaTS}}\boldsymbol{x}(t,l,v_\mathrm{t})$ is a three-dimensional (3-D) representation along t, l, and v_t. On the flip side, its number of log-frequency bins l is equal to L, i.e. much less than the number of log-frequency bins λ for the second-order time scattering $\mathbf{S}_2\boldsymbol{x}(t,\lambda,v_\mathrm{t})$. We then log-normalise the AdaTS via Eq. (5) and obtain $\widetilde{\mathbf{S}}_2^{\mathrm{AdaTS}}\boldsymbol{x}(t,l,v_\mathrm{t})$. Fig. 5 (c) shows the log-normalised AdaTS decomposed from the dominant band trajectory of Fig. 5 (a).

Besides the difference on the fundamental modulation rate, the AdaTS of PMTs exhibits different spectral characteristics along the modulation rate axis, as observed from Fig. 5 (c). Tremolo has only the fundamental modulation rate while trill and vibrato carry upper harmonic partials. Trill has a richer harmonic structure than vibrato. These characteristics may provide additional information for the recognition of PMTs. Therefore, we propose to apply frequency scattering along

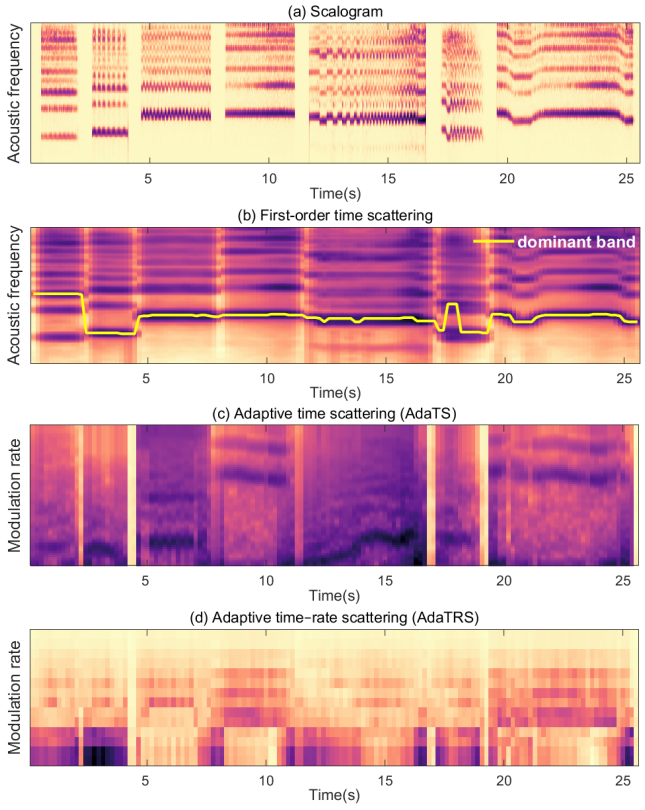

Fig. 5. Extracting the frequency-adaptive scattering representations for PMTs: vibrato, tremolo, trill, flutter-tongue, variable rate trill, variable extent trill, and variable pitch flutter-tongue. (a) scalogram; (b) dominant band trajectory in the first-order time scattering; (c) adaptive time scattering (AdaTS) obtained by localising and decomposing the scalogram trajectory; (d) adaptive time–rate scattering (AdaTRS) obtained by applying a spectral filterbank. The AdaTS+AdaTRS is the frame-wise concatenation of (c) and (d).

the modulation rate axis of $\widetilde{\mathbf{S}}_2^{\mathrm{AdaTS}}\boldsymbol{x}(t,l,v_\mathrm{t})$ with a wavelet filterbank $\boldsymbol{\psi}_{v_\mathrm{f}}(v_\mathrm{t})$ and define the *frequency-adaptive time–rate scattering* (AdaTRS) as:

$$\mathbf{S}_2^{\mathrm{AdaTRS}}\boldsymbol{x}(t,l,v_\mathrm{t},v_\mathrm{f}) = \left(\left|\widetilde{\mathbf{S}}_2^{\mathrm{AdaTS}}\boldsymbol{x} \overset{v_\mathrm{t}}{*} \boldsymbol{\psi}_{v_\mathrm{f}}\right| \overset{v_\mathrm{t}}{*} \boldsymbol{\phi}_F\right)(t,l,v_\mathrm{t}), \tag{10}$$

where $\boldsymbol{\phi}_F(v_\mathrm{t})$ is a lowpass filter. Frequency scattering has a similar form as time scattering where the former generally applies a wavelet filterbank along the acoustic frequency axis, i.e. $\boldsymbol{\psi}_{v_\mathrm{f}}(\lambda)$, such as the time–frequency scattering operators shown in Fig. 3. In such cases, it generates representations that are invariant to frequency-transpositions and captures modulation information along the log-frequency axis of the scalogram. In this paper, the proposed AdaTS itself is frequency-transposition invariant due to the adaptive operation. Applying frequency scattering on top of the AdaTS is to capture its characteristics along the modulation rate axis. Fig. 5 (d) displays the AdaTRS obtained from (c). We define *AdaTS+AdaTRS* as the concatenation of AdaTS and AdaTRS:

$$\mathbf{S}_2^{\mathrm{AdaTS+AdaTRS}}\boldsymbol{x}(t,l,v_\mathrm{t},v_\mathrm{f}) = \left\{\widetilde{\mathbf{S}}_2^{\mathrm{AdaTS}}\boldsymbol{x}(t,l,v_\mathrm{t}), \mathbf{S}_2^{\mathrm{AdaTRS}}\boldsymbol{x}(t,l,v_\mathrm{t},v_\mathrm{f})\right\}, \tag{11}$$

which is the input representation to our recognition system for PMTs in Section VI.

C. Scattering for Pitch Modulation-based Techniques

From the analysis in Section IV-A, the core information for PMT recognition lies in the modulation rate, extent, and shape. Fig. 5 shows (a) the scalogram, (b) the first-order time scattering, (c) the AdaTS, and (d) the AdaTRS representations of a series of PMT examples in the CBFdataset (see Section VII-A1). The first four are modulations based on stable pitches or constant parameters: vibrato, tremolo, trill, and flutter-tongue. The last three are cases with time-varying parameters: trills with variable rate and with variable extent, and flutter-tongue with time-varying pitch.

Fig. 5 (c) is the AdaTS decomposed only from the dominant band trajectory. Flutter-tongue is the most discriminative one with the highest modulation rate. Dominant band decomposition also captures trills because of their large modulation extent. This can be interpreted by filters with a bandwidth larger than one semitone, which blurs other subtle modulations. To specifically detect vibratos and tremolos, we use frequency bands less than one semitone and concatenate the decompositions of multiple bands. Ideally, the AdaTS of tremolo should display only the fundamental modulation rate with no upper harmonics since tremolo is an AM. This is verified by the second example in Fig. 5 (c). However, vibratos are FMs with modulations spread over neighbouring frequency bands. Decomposing neighbouring frequency bands above or below the dominant band provides additional information to distinguish vibratos from tremolos. All this discriminative information can be visualised from the fundamental modulation rate and the richness of the harmonics of the AdaTS in Fig. 5 (c). Although the last example is flutter-tongue with time-varying pitch, its modulation rate is relatively stable. The trills with variable rate and extent are also captured. To capture the spectral structure of the AdaTS (Fig. 5 (c)), we apply frequency scattering along the modulation rate axis and obtain the AdaTRS (Fig. 5 (d)), which provides extra information for the discrimination between PMTs.

V. Direction-adaptive joint time–frequency scattering for Pitch Evolution-based Techniques

A. Characteristics of Pitch Evolution-based Techniques

Similarly to Section IV-A, we analyse characteristics of PETs and calculate statistical information from the CBFdataset (see Section VII-A1), as shown in Table II. Each playing technique has a specific duration range: 0.1–0.4 s for acciaccatura, 0.2–1.2 s for portamento, and 0.2–1.1 s for glissando. For temporal variations, although all three playing techniques contain monotonic pitch changes over time, portamento exhibits smooth pitch changes while the pitch changes within acciaccatura and glissando are both at the note level. Acciaccatura contains only one note change, while glissando spans a series of note changes. For spectral variations, acciaccatura has a noisy attack while glissando and portamento exhibit clear harmonic structures. The possible directions of their pitch changes are different: acciaccatura in CBF playing only occurs downwards, while the other two techniques can exhibit both upward and downward directions.

TABLE II
CHARACTERISTICS OF PITCH EVOLUTION-BASED TECHNIQUES

Characteristics	Acciaccatura	Portamento	Glissando
Duration (s)	0.1-0.4	0.2-1.2	0.2-1.1
Temporal variation	One note change	Smooth pitch changes	Consecutive note changes
Spectral variation	Noisy attack	Harmonic	Harmonic
Pitch direction	↘	↗ or ↘	↗ or ↘

B. Direction-adaptive Joint Time–Frequency Scattering

Different from separable scattering [41], which calculates time and frequency scattering in separate steps, the *joint scattering* [11] applies them jointly. The interaction of the two types of wavelet convolutions captures spectro-temporal variations in the time–frequency domain. Motivated by the recognition task for PETs, we interpret the definition of the joint scattering in [11] from a new perspective. Rather than formulating a 2-D mother wavelet, we consider the temporal and spectral wavelet convolutions in a sequential manner. This is more precise in terms of the computations performed and provides explicit information of what has been captured at each step.

Following the notations in Section III, we denote by $\psi(t)$ and $\psi(\lambda)$ the mother wavelets along the time and the log-frequency axes, respectively; $\psi_{v_\mathrm{t}}(t)$ and $\psi_{v_\mathrm{f}}(\lambda)$ are the corresponding wavelet filterbanks dilated from the mother wavelets. We introduce an orientation variable $\theta = \pm 1$ to reflect the oscillation direction (up or down) of the spectro-temporal pattern. Specifically, $\theta = -1$ flips the centre frequency of wavelet $\psi_{v_\mathrm{f}}(\lambda)$ from 2^{v_f} to -2^{v_f}. The resulting temporal and spectral wavelet filterbanks are respectively:

$$\begin{aligned} \psi_{v_\mathrm{t}}(t) &= 2^{v_\mathrm{t}}\psi(2^{v_\mathrm{t}}t) \quad \text{and} \\ \psi_{v_\mathrm{f},\theta}(\lambda) &= 2^{v_\mathrm{f}}\psi(\theta 2^{v_\mathrm{f}}\lambda). \end{aligned} \tag{12}$$

The joint wavelet transform of $\mathbf{X}(t,\lambda)$ computes convolutions of the form:

$$\left((\mathbf{X} \overset{t}{*} \psi_{v_\mathrm{t}}) \overset{\lambda}{*} \psi_{v_\mathrm{f},\theta}\right)(t,\lambda) = \left(\mathbf{X} \overset{t,\lambda}{*} (\psi_{v_\mathrm{t}} \otimes \psi_{v_\mathrm{f},\theta})\right)(t,\lambda), \tag{13}$$

where the operator $\otimes$ denotes the outer product between two 1-D wavelets, returning a 2-D wavelet; and the symbol $\overset{t,\lambda}{*}$ denotes a 2-D convolution over both the time variable t and the log-frequency variable λ. In practice, we implement the joint time–frequency convolution via the left-hand side of the equation above, that is, by a sequence of two 1-D convolutions. This two-step factorised procedure is more efficient than the one-step 2-D convolution, described on the right-hand side. However, the right-hand side of Eq. (13) is useful for the theoretical understanding of joint scattering as involving a joint convolutional operator in the time–frequency domain. Indeed, we may view the outer product between the temporal wavelet $\psi_{v_\mathrm{t}}(t)$ and the spectral wavelet $\psi_{v_\mathrm{f},\theta}(\lambda)$ as the factorisation of a joint time–frequency wavelet,

$$\Psi_{v_\mathrm{t},v_\mathrm{f},\theta}(t,\lambda) = \psi_{v_\mathrm{t}}(t)\psi_{v_\mathrm{f},\theta}(\lambda), \tag{14}$$

which captures the local spectro-temporal modulations of $\mathbf{X}(t,\lambda)$ around time t and log-frequency λ in terms of the temporal variability v_t, the spectral variability v_f, and the orientation θ.

For a specific recognition task at hand, we typically focus on a spectro-temporal pattern smaller than a "time–frequency box" restricted by some time scale T in samples and frequency interval F in octaves. To ensure local time-shifting invariance, time-warping stability, frequency-transposition invariance, and frequency-warping stability, we take the modulus of the output of Eq. (13) and average it by a 2-D lowpass filter $\Phi_{T,F}(t,\lambda)$. Following [11], we define the *joint time–frequency scattering* (JTFS) of $\mathbf{X}(t,\lambda)$ according to Eqs. (13) and (14) as:

$$\mathbf{S}_2^{\mathrm{JTFS}}\boldsymbol{x}(t,\lambda,v_\mathrm{t},v_\mathrm{f},\theta) = \left(\left|\mathbf{X} \overset{t}{*} \psi_{v_\mathrm{t}} \overset{\lambda}{*} \psi_{v_\mathrm{f},\theta}\right| \overset{t,\lambda}{*} \Phi_{T,F}\right)(t,\lambda). \tag{15}$$

Fig. 6 shows the calculation process of the JTFS for a glissando. Convolving (a) the scalogram $\mathbf{X}(t,\lambda)$ with $\psi_{v_\mathrm{t}}(t)$, we obtain (b) the temporal wavelet transform, which mainly captures the temporal variations of each frequency band. To capture correlations across frequency bands, we apply wavelet convolution with $\psi_{v_\mathrm{f},\theta}(\lambda)$ along the log-frequency axis and obtain (c) the joint wavelet transform. Taking complex modulus of (c) and averaging the resulting coefficients yield (d) the JTFS. According to Eq. (15), for each "time–frequency" box around (t,λ), we obtain a 3-D tensor indexed by $(v_\mathrm{t},v_\mathrm{f},\theta)$. As shown in Fig. 6 (d), this tensor captures the joint activation of temporal and spectral variations as well as its direction.

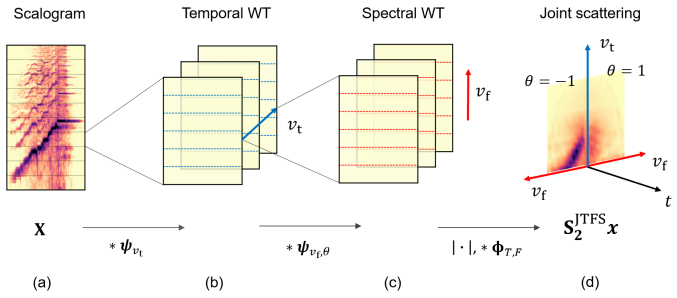

Fig. 6. Calculating the joint time–frequency scattering (JTFS) for a glissando: (a) scalogram; (b) temporal wavelet transform (temporal WT) by convolving with temporal filterbank ψ_{v_t}; (c) spectral wavelet transform (spectral WT) by applying spectral filterbank $\psi_{v_\mathrm{f},\theta}$; (d) the JTFS, result of modulus operation and averaging with 2-D lowpass filter $\Phi_{T,F}$.

C. Scattering for Pitch Evolution-based Techniques

As discussed in Section V-A, each PET exhibits one direction of pitch change, while according to Eq. (15), we obtain information for both directions. For recognising only the type of PETs, we modify the JTFS into the *direction-adaptive joint time–frequency scattering* (dJTFS), which introduces a pooling operation over the direction variable of the JTFS. It can be either max-pooling or average-pooling. We define the former case as *dJTFS-max*, extracting only the JTFS corresponding to $\theta_{\max}$:

$$\mathbf{S}_2^{\mathrm{dJTFS-max}}\boldsymbol{x}(t,\lambda,v_\mathrm{t},v_\mathrm{f}) = \mathbf{S}_2^{\mathrm{JTFS}}\boldsymbol{x}(t,\lambda,v_\mathrm{t},v_\mathrm{f},\theta_{\max}). \tag{16}$$

where $\theta_{\max}$ is the direction with maximum spectro-temporal modulation energy:

$$\theta_{\max}(t) = \arg\max_{\theta=\pm 1} \sum_{\lambda, v_{\mathrm{t}}, v_{\mathrm{f}}} \mathbf{S}_2^{\mathrm{JTFS}} \boldsymbol{x}(t, \lambda, v_{\mathrm{t}}, v_{\mathrm{f}}, \theta). \tag{17}$$

In the latter case, we average the JTFS over both directions, i.e. $\theta = 1$ and $\theta = -1$, and define the resulting representation as *dJTFS-avg*:

$$\mathbf{S}_2^{\mathrm{dJTFS-avg}} \boldsymbol{x}(t, \lambda, v_{\mathrm{t}}, v_{\mathrm{f}}) = \frac{1}{2} \sum_{\theta=1,-1} \mathbf{S}_2^{\mathrm{JTFS}} \boldsymbol{x}(t, \lambda, v_{\mathrm{t}}, v_{\mathrm{f}}, \theta). \tag{18}$$

We compare the performance of dJTFS-max and dJTFS-avg on PET recognition in Section VIII-A1.

Fig. 7 shows the JTFS of acciaccatura, portamento, and glissando: (a) is the spectrogram; (b), (c), and (d) are the 2-D joint activations for each type of PET. As observed, although both acciaccatura and glissando have high-energy regions in the JTFS, their energy distributions along the variation scales are different. From (b) and (d), noisy attacks show as diffused energy in the JTFS, and the time and frequency regularity of glissando results in clear slopes. Extracting the JTFS coefficients via max-pooling or average-pooling over its direction variable reduces the dimensionality by half without losing useful information.

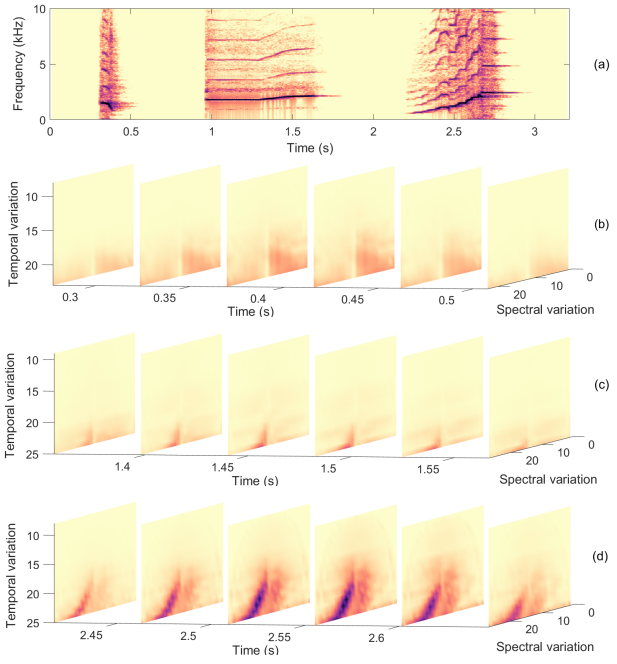

Fig. 7. Joint activation of temporal and spectral variations for PETs. (a) Spectrogram showing acciaccatura, portamento, and glissando; (b), (c), and (d) are the corresponding JTFS plots for each case.

VI. Playing Technique Recognition

To develop a general framework for recognising playing techniques, we investigate two classification schemes: (1) A recognition system with a set of binary classifiers, each detecting one type of playing technique. Each classifier takes as input the proposed frequency-adaptive scattering or dJTFS coefficients with hyperparameters set according to the characteristics of each technique. (2) A recognition system with a multiclass classifier, using as input the concatenation of the AdaTS+AdaTRS and dJTFS-avg, and detecting all playing techniques simultaneously. The binary classification scheme exhibits lower feature dimension and is capable of detecting co-articulations, such as the combination of tremolo and trill, or glissando co-articulated with flutter-tongue. The multiclass one provides a confusion matrix between techniques. In this section, we discuss experimental settings for the binary classifiers; we introduce the multiclass classifier in Section VII-D.

A. Adaptive Scattering Features

Table III gives the hyperparameters of the proposed scattering representations, the frequency-adaptive scattering and the dJTFS, which capture the discriminative information for PMT and PET recognition. For detecting PMTs, we calculate and compare all three frequency-adaptive scattering operators: the AdaTS, AdaTRS, and AdaTS+AdaTRS. The averaging scale T (in samples) is useful for discriminating modulations with large differences on the modulation rate, for example, for distinguishing flutter-tongue from other low-rate PMTs. Averaging scales covering at least four unit patterns are recommended for reliable estimation of the modulation rate. According to the rate range of PMTs (see Table I), we use $T = 2^{13}$ (186 ms at a sampling rate $F_s = 44.1$ kHz) for flutter-tongue, and $T = 2^{15}$ (743 ms) for other three techniques. The range M (in Hz) of the modulation rate narrows the frequency-adaptive scattering to the part that contains core information of the playing technique. Setting an interval for M, the frequency-adaptive scattering extracts only the coefficients corresponding to this range. An interval larger than the modulation rate provides some harmonics in the modulation representation. For example, we set $M = [0, 150]$ Hz for flutter-tongue, and $M = [0, 100]$ Hz for the other three PMTs. $Q_1^{(\mathrm{t})}$ is the number of filters per octave of the temporal filterbank in the first-order time scattering. Here, we use $Q_1^{(\mathrm{t})} = 16$ to support subtly-modulated vibratos and tremolos, of which the modulation extent is less than one semitone. $Q_1^{(\mathrm{t})} = 12$ is applied to trill due to its note-level nature. Since the most distinct feature of flutter-tongue is the modulation rate, we set a small $Q_1^{(\mathrm{t})} = 4$ for computation saving. L is the number of frequency bands centred at the dominant band in the scalogram. For all modulations, we use $L = 7$ according to experimental results. All frequency-adaptive scattering representations operate with $Q_2^{(\mathrm{t})} = 1$ and $Q_2^{(\mathrm{t})} = 4$ filters per octave for flutter-tongue and the other three techniques, respectively. Besides the hyperparameters above, the AdaTRS uses frequency scattering with $Q_1^{(\mathrm{f})} = 1$ filters per octave and an averaging scale corresponding to the entire modulation rate axis of the AdaTS.

For recognising PETs, we encode the characteristics of each playing technique into the dJTFS representations by setting appropriate transform parameters in a similar way. The averaging scale T carries duration information via setting T approximately equivalent to the mean duration of each playing technique. According to the duration range of PETs

TABLE III
HYPERPARAMETERS OF THE PROPOSED SCATTERING REPRESENTATIONS THAT CAPTURE DISCRIMINATIVE INFORMATION FOR PMTS AND PETS

Frequency-adaptive scattering for PMTs			Direction-adaptive joint time–frequency scattering for PETs		
Hyperparameter	Notation	Characteristics	Hyperparameter	Notation	Characteristics
Averaging scale	T	Modulation rate	Averaging scale	T	Duration
Temporal filters per octave	$Q_1^{(\mathrm{t})}$	Modulation extent	Temporal filters per octave	$Q_1^{(\mathrm{t})}$	Pitch change
Spectral filters per octave	$Q_1^{(\mathrm{f})}$	Spectral structure	Spectral filters per octave	$Q_1^{(\mathrm{f})}$	Spectral structure
Number of bands decomposed	L	Modulation shape	Orientation variable	θ	Direction of pitch change
Feature dimension reduction	M	Modulation rate range	Feature dimension reduction	M	Temporal variation range

in Table II, we use $T = 2^{13}$ (186 ms) for acciaccatura, and $T = 2^{14}$ (372 ms) for portamento and glissando. Here, $Q_1^{(\mathrm{t})}$ is useful for distinguishing note changes from smooth pitch changes. For acciaccatura and glissando, we set $Q_1^{(\mathrm{t})} = 12$ due to their note-change property. To capture the smooth pitch evolution within portamento, $Q_1^{(\mathrm{t})} > 12$ is required, and we set $Q_1^{(\mathrm{t})} = 16$ for this case. We use $Q_2^{(\mathrm{t})} = 2$ due to the less oscillatory nature of audio signals at this order of decomposition. One may observe from Fig. 7 (a) the different harmonic structures between the selected PETs. This timbral information can be captured by applying frequency scattering with $Q_1^{(\mathrm{f})}$ filters per octave. Here we use $Q_1^{(\mathrm{f})} = 2$ filters per octave and average the coefficients along the whole log-frequency axis. We then obtain the dJTFS of PETs for each time frame according to Eq. (16) or Eq. (18). Similar to the frequency-adaptive scattering, we also use a range $M = [0, 50]$ Hz to extract meaningful temporal modulation information. The evolutionary nature of PETs suggests the importance of temporal context. Here we calculate the mean and standard deviation of 5 frames centred at the current frame to represent contextual information. All scattering features used in this paper are log-normalised coefficients by Eq. (5).

B. Recognition System

We use support vector machines (SVMs) [44] with Gaussian kernels as classifiers throughout the paper due to their good generalisability based on a limited amount of training data [45]. The SVM hyperparameters to be optimised are the error penalty parameter C and the width of the Gaussian kernel γ. We use consistent parameter grids of $2^{\{3:1:8\}}$ and $2^{\{-12:1:-7\}}$ for C and γ, respectively, during training and select the best hyperparameters for testing. In the recognition process, the CBFdataset (see Section VII-A1) is split into training and test sets according to an 8:2 ratio by performers (performers are randomly initialised). We create 5 splits in the same way, with no performer overlap between the test sets across splits and between the training-test sets in each split. Within each split, we run a 3-fold cross-validation, sampling on the training set such that each fold includes approximately the same ratio of positive and negative class instances for a given playing technique. This is to avoid the case where there is no instance or too few instances of a given playing technique class in the validation set if we further split the training set based on performer identity.

The classifiers take as input the frequency-adaptive scattering or the dJTFS features and output frame-wise predictions of playing technique type. All features are z-score normalised. As introduced in Section III, the scattering coefficients are the results of convolving the wavelet modulus transform with a lowpass filter. The original frame size of the scattering coefficients equals the averaging scale T, ranging from 186 to 743 ms for the techniques discussed in Section VI-A. To compensate for the low temporal resolution resulting from the large averaging scales, we use an oversampling parameter α [39] which introduces overlaps to neighbouring averaging windows. The frame size h is then inversely log-proportional to α by $h = T/(F_s \times 2^{\alpha})$. We set $\alpha = 2$ consistently for all classifiers, which yields frame sizes for flutter-tongue, trill, vibrato, tremolo, acciaccatura, portamento, and glissando of 46, 186, 186, 186, 46, 93, and 93, respectively (all in ms). Besides the proposed scattering operators, we also investigate the performance of two existing scattering representations, i.e. the standard scattering for PMT detection and the JTFS for PET recognition. We list the frame sizes and dimensionalities of the scattering representations for each type of playing technique in the supplementary material [14].

VII. EVALUATION

A. Datasets

Most existing datasets for playing technique analysis include only techniques recorded in isolation, without considering the variations of techniques in real-world performances. We release publicly a new dataset of Chinese bamboo flute performances (CBFdataset) for analysing playing techniques recorded in context. To further verify the methodology, we test the proposed system on three existing datasets with a variety of playing techniques: vibrato/portamento dataset (VPset) [6], Studio On Line (SOL) dataset [5], and vocal technique dataset (VocalSet) [35]. We call these three datasets the *additional datasets*. The types of playing techniques and number of samples in each dataset are summarised in Fig. 10.

1) CBFdataset: The CBFdataset comprises monophonic Chinese bamboo flute performances and expert annotations of seven playing techniques: vibrato, tremolo, trill, flutter-tongue, acciaccatura, portamento, and glissando. The performances were recorded by 10 professional CBF performers from the China Conservatory of Music. All data was recorded in acoustically treated environments of professional recording studios using a Zoom H6 recorder with its stock microphones, in xy stereo configuration, at 44.1kHz/24-bits. Each performer played both isolated playing techniques covering all notes on the CBF and two full-length pieces selected

from *Busy Delivering Harvest* «扬鞭催马运粮忙», *Jolly Meeting* «喜相逢», *Morning* «早晨», and *Flying Partridge* «鹧鸪飞». Performers were grouped by flute type (C and G, the most representative types for Southern and Northern styles, respectively) with each performer used their own flute. The dataset was originally published as two subsets, CBF-periDB [12] and CBF-petsDB [13]; in this paper, we use the complete CBFdataset for all experiments. All recordings and playing technique annotations in the CBFdataset can be downloaded from `c4dm.eecs.qmul.ac.uk/CBFdataset.html`.

2) Additional datasets: To provide additional evidence on the generalisability of the proposed framework for recognising playing techniques, we examine the existing datasets and focus on three datasets below based on their diversity of playing techniques, performers, or instruments.

VPset: Proposed in [6], the vibrato[2]/portamento[3] dataset (VPset) includes two separate subsets. The vibrato subset comprises 4 full-length pieces played on the Chinese instrument erhu and Western instrument violin, 64 short excerpts of solo instrument playing, and vibrato annotations. The duration of this subset is 25 minutes. Besides having the same erhu and violin recordings as the vibrato subset, the portamento subset also includes recordings of Beijing opera singing and portamento annotations; the total duration is 55 minutes. It is not applicable to concatenate the two subsets into one since there are no vibrato (portamento) annotations for Beijing opera singing (solo instrument playing) in the portamento (vibrato) subset. For simplicity, we hereafter denote these two subsets as the VPset. When it comes specifically to vibrato (portamento) detection, we refer to the *vibrato (portamento) subset*.

SOL[4]: Studio On Line [5] (version 0.9HL) is a multi-instrument dataset, comprising 12 categories of instruments playing isolated tones. It covers 140 types of playing techniques; total duration is 27.1 hours. To focus on commonly-used playing techniques, we consider only playing techniques with over 100 excerpts. Non-techniques like crescendos and decrescendos are beyond the scope of this paper. The list of playing techniques is shown in Fig. 10 (c); audio with the considered data total 9.8 hours. For the playing technique labels, we follow the original annotations except for five labels resulting from merging similar patterns: sul-tasto/ponticello, pizzicato, glissando, trill, and flatterzunge. For example, we merge the labels trill-major-second-up and trill-minor-second-up into one label, trill. Note that glissando in the SOL dataset corresponds to portamento in the CBFdataset, both consisting of smooth pitch changes.

VocalSet[5]: a singing voice dataset [35]. It has recordings of 10.1 hours of 20 professional singers (11 male, 9 female) performing 17 different vocal techniques. To make the results comparable to that obtained in [35], we focus on the same ten techniques: straight, vibrato, belt, lip trill, breathy, vocal fry, trillo, inhaled, trill, and spoken, as shown in Fig. 10 (d). The number of trills and spoken techniques in this dataset are below 100, with 95 and 20 examples, respectively.

[2]https://github.com/skx300/vibrato_dataset
[3]https://github.com/skx300/portamento_dataset
[4]https://forum.ircam.fr/projects/detil/orchids/
[5]https://zenodo.org/record/1193957

We order playing techniques in each dataset by number of samples and group them into PMTs and PETs, as shown by the top subfigures of Fig. 10. Here PMTs and PETs are not limited to the CBF playing techniques discussed in Sections IV-A and V-A; they include similar acoustic patterns that follow the definitions of PMTs and PETs.

B. Metrics

Playing techniques are typically music events with certain durations. Due to the heterogeneous structure of the datasets, three ways of evaluation are considered for performance comparison: frame-based, event-based, and clip-based evaluation. We use precision $\mathcal{P} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$, recall $\mathcal{R} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, and F-measure $\mathcal{F} = \frac{2\mathcal{P}\mathcal{R}}{\mathcal{P}+\mathcal{R}}$ as the metrics for each evaluation method, where $\mathrm{TP}, \mathrm{FP}, \mathrm{FN}$ are true positives, false positives, and false negatives, respectively [46].

Frame-based: Labels assigned by the classifier are compared to the ground truth in a frame-wise manner. The frame sizes are different from technique to technique. When evaluating for a specific technique over different methods, we resample the detection result to the same frame sizes that we use for CBF technique evaluation. The CBFdataset, VPset, and VocalSet are evaluated in this way.

Event-based: The CBFdataset includes mainly full-length performances. To investigate the recognition result at the event level, we merge frame labels into events and evaluate each type of playing technique based on the onset and duration of its instances in the test set. Frame labels are merged into events according to the *mir_eval* [47] Python library. Considering the duration range of each playing technique, the events are postprocessed by minimum duration pruning and gap filling. We fill the gaps between neighbouring events when the gaps are shorter than the shortest event in the training set; and prune the events that have durations smaller than the minimum duration event in the training set. The minimum duration is automatically calculated subject to the technique, dataset, and training-test split during recognition. Onsets of events are also evaluated using *mir_eval* [47], which computes a maximum match between reference and estimated onsets, subject to a window constraint. An event is considered to be detected only when its onset falls within a 200 ms window of the ground truth and its duration is at least 50% of the ground truth.

Clip-based: This method of evaluation is considered for the SOL dataset, which comprises short audio clips with one technique per clip. To each clip, one label is assigned based on the majority vote of its frame labels.

C. Baselines

There does not yet exist any general framework for detecting all seven types of playing techniques although methods for detecting specific playing techniques can be found in the literature, such as FDM for vibrato detection [6], and HMMs for portamento [6] and glissando [33] recognition. Therefore, we compare the proposed system with these methods for vibrato, portamento, and glissando detection in the CBFdataset for the binary classification scheme (see Section VIII-A1). All detection systems take as input the frame-wise F0 estimated

TABLE IV
FRAME-BASED BINARY CLASSIFICATION RESULTS FOR THE SEVEN CBF PLAYING TECHNIQUES USING THE PROPOSED ADAPTIVE SCATTERING TRANSFORMS, EXISTING SCATTERING REPRESENTATIONS, AND BASELINE METHODS. ALL NUMBERS ARE F-MEASURE SCORES (%).

PMT Recognition						PET Recognition				
PMTs	AdaTS	AdaTRS	AdaTS+AdaTRS	Standard	FDM	PETs	dJTFS-max	dJTFS-avg	JTFS	HMM
Flutter-tongue	88.8	80.6	89.2	**91.6**	NA	Acciaccatura	70.4	**74.8**	73.0	NA
Trill	89.8	83.6	**90.4**	85.6	NA	Portamento	**66.4**	66.0	63.6	30.0
Vibrato	71.2	60.6	**72.0**	65.0	67.7	Glissando	81.4	86.4	**86.8**	12.7
Tremolo	42.6	30.0	42.2	**48.8**	NA					
Average	73.1	63.7	**73.5**	72.8	NA	Average	72.7	**75.7**	74.5	NA

TABLE V
EVENT-BASED BINARY CLASSIFICATION RESULTS FOR THE SEVEN CBF PLAYING TECHNIQUES USING THE PROPOSED ADAPTIVE SCATTERING TRANSFORMS, EXISTING SCATTERING REPRESENTATIONS, AND BASELINE METHODS.

PMT Recognition						PET Recognition				
PMTs	AdaTS	AdaTRS	AdaTS+AdaTRS	Standard	FDM	PETs	dJTFS-max	dJTFS-avg	JTFS	HMM
Flutter-tongue	72.0	54.3	71.3	**84.5**	NA	Acciaccatura	74.1	**78.2**	76.1	NA
Trill	71.5	56.2	**73.3**	42.7	NA	Portamento	**65.7**	65.6	63.5	22.4
Vibrato	50.1	35.9	50.1	32.0	**58.9**	Glissando	75.6	83.1	**84.6**	14.8
Tremolo	25.2	16.7	**25.9**	25.1	NA					
Average	54.7	40.8	**55.2**	46.1	NA	Average	71.8	**75.6**	74.7	NA

by pYIN [36]. The FDM feature is then computed and fed into a naive Bayes classifier for vibrato detection. Portamento and glissando recognition are both based on HMMs. For fair comparisons, we resample the detection results into the same frame sizes as that used for CBF vibrato, portamento, and glissando evaluation (see Section VI-B). Different hyperparameter ranges are experimented for FDM and HMMs based on the characteristics of these three techniques. The best frame-based F-measures obtained for vibrato, portamento, and glissando detection in the CBFdataset are 67.7%, 30.0%, and 12.7%; while the event-based ones are 58.9%, 22.4%, and 14.8%.

To detect all seven playing techniques simultaneously via the multiclass classification scheme (see Section VIII-A2), we compare the proposed representations with commonly used features such as MFCCs and MPS for the CBFdataset. Macro F-measures obtained using MFCCs and MPS are 35.9% and 52.0%, respectively. Fig. 10 displays the F-measure for recognising each playing technique in the CBFdataset based on these two features. "Other" refers to frames that are none of the discussed seven playing techniques. The supplementary material [14] compares the performance of MFCCs and MPS on the CBFdataset in terms of confusion matrices.

For the additional datasets, we compare the proposed system with FDM for vibrato detection and with HMMs for portamento recognition in the VPset [6]; and with CNNs for detecting vocal techniques in the VocalSet [35]. This is because these methods were originally used for detecting playing techniques in the corresponding datasets. Frame-based F-measures for vibrato and portamento recognition are 77.7% and 50.6% for the VPset. CNNs were used in [35] for vocal technique classification with a frame size of 3 s. A macro F-measure of 65.2% for the 10 techniques were reported. For the SOL dataset, we also compare the proposed representations with MFCCs and MPS. Macro F-measures for detecting the 17 playing techniques in the SOL dataset using MFCCs and MPS are 27.1% and 26.6%, respectively. The bottom subfigures of Fig. 10 display frame-based F-measures for recognising each type of playing technique in these three additional datasets.

D. Experimental Settings

In this section, we build a recognition system with a multiclass classifier for all the datasets in Section VII-A. The binary classifiers discussed in the previous section detect co-articulations. However, cases of co-articulation form only a small portion of the CBFdataset (details can be found in the supplementary material [14]) and do not exist in other datasets. In the multiclass classification for CBF techniques, we discard all co-articulation samples. This not only enables us to generate a confusion matrix between techniques, but also to provide comparable results across datasets.

For the CBFdataset, the system takes the concatenation of the proposed AdaTS+AdaTRS and dJTFS-avg features as input. This is based on a comparison of the recognition results on the CBFdataset using the AdaTS+AdaTRS only, the dJTFS-avg only, and the concatenation of AdaTS+AdaTRS and dJTFS-avg as input to the classifier, respectively, where the concatenated feature yields the best result. We provide details of this comparison in the supplementary material [14]. We set $T = 2^{15}$, $Q_1^{(\mathrm{t})} = 16$, $Q_2^{(\mathrm{t})} = 4$, $Q_1^{(\mathrm{f})} = 1$, $\alpha = 2$, and $M = [0, 100]$ Hz for calculating the AdaTS+AdaTRS. The dJTFS-avg is calculated via $T = 2^{14}$, $Q_1^{(\mathrm{t})} = 16$, $Q_2^{(\mathrm{t})} = 2$, $Q_1^{(\mathrm{f})} = 2$, $\alpha = 2$, and $M = [0, 50]$ Hz. Due to the different averaging scales, $T = 2^{15}$ for the AdaTS+AdaTRS and $T = 2^{14}$ for the dJTFS-avg, we duplicate the AdaTS+AdaTRS coefficients before concatenation to have the same number of frames as the dJTFS-avg, with a frame size of 93 ms. The feature dimension is 613, higher than those in the binary classification. The data split follows that in the binary classification.

For the additional datasets, we conduct binary classification for the VPset and multiclass classification for the SOL dataset and VocalSet. This is because vibrato and portamento techniques are from two separate subsets in the VPset, as described in Section VII-A2. All experiments for the additional datasets use the same settings as the CBF binary or multiclass classification, i.e. hyperparameters of the scattering transform and hyperparameter grids of SVMs. However, the ways of splitting data vary according to dataset. Rather than cross-validating within the same performance for the VPset in [6], we take into account the performer identity. We use one performer's playing for testing and the remaining recordings for training, and repeat this for all four performers. The final result is the average of frame-based F-measures over all performances. After removing silence from the recordings, a random 8:2 training-test split ratio is used for the SOL dataset due to a lack of performer information. For the VocalSet, we keep the data splits as in the original work [35]. Silence is also removed before scattering feature extraction. All samples from 15 singers are placed in the training set and the remaining 5 singers in the test set. The frame sizes and feature dimensionalities on all additional datasets (except the VocalSet) are the same as those on the CBFdataset.

VIII. Results

A. CBFdataset

1) Binary classification: Tables IV and V show the frame- and event-based binary classification results for the four PMTs and the three PETs in the CBFdataset using the proposed adaptive scattering transforms, existing scattering representations (standard scattering and JTFS), and baseline methods (FDM and HMM). We compare these results from three fronts: the performance of different scattering representations, the recognition results on different playing techniques, and the comparison between the proposed scattering representations and the baseline methods.

In both frame- and event-based evaluation, the AdaTS+AdaTRS and the dJTFS-avg achieve the best overall performance for PMT and PET recognition, respectively, measured by the average F-measure over the playing techniques in each group (see the last row of Tables IV and V). For example, in the frame-based evaluation, the AdaTS+AdaTRS yields an average F-measure over the four PMTs of 73.5% against 72.8% from the standard scattering; and the dJTFS-avg returns an average F-measure over the three PETs of 75.7% versus 74.5% from the JTFS. Similar trends take place in the event-based evaluation. Besides the performance difference, the proposed representations have much lower dimensionalities than the standard scattering and the JTFS (see [14]). Narrowing the scope with the three frequency-adaptive scattering representations—AdaTS, AdaTRS, and AdaTS+AdaTRS—we notice that the AdaTS+AdaTRS slightly outperforms the other two. The average F-measure improves 0.4% and 0.5% in the frame- and event-based evaluation, respectively, as compared to the AdaTS only. With regard to the two direction-adaptive scattering representations, i.e. dJTFS-max and dJTFS-avg, the latter achieves frame- and event-based average F-measure of 3.0% and 3.8% higher than that of the former.

Comparing the recognition performance of different methods for specific playing techniques, we observe that although the proposed system does not achieve the best results for all techniques, it yields considerably higher F-measures for some techniques. Among the four PMTs, the AdaTS+AdaTRS generates frame-based F-measure improvement of 4.8% and 7.0% for trill and vibrato detection, and event-based F-measure increase of 30.6%, 18.1%, 0.8% for trill, vibrato, and tremolo detection, respectively, as compared to the standard scattering. For flutter-tongue and tremolo detection, the AdaTS+AdaTRS yields frame-based F-measures that are 2.4% and 6.6% lower than the standard scattering; and returns an event-based F-measure of 13.2% less than the standard scattering for flutter-tongue recognition. The poorer performance of the AdaTS+AdaTRS over the standard scattering for these two techniques may be attributed to the timbre or energy variations within the technique. As shown in Fig. 1 (a), there exists upper harmonic fading within the tremolo technique and energy variation within the flutter-tongue technique. The timbral variation in the former may not be captured when we extract only the frequency bands around the dominant band for calculating the second-order scattering coefficients and the energy variation in the latter may introduce instabilities to the dominant band trajectory.

Switching to the recognition results on the three PETs, we observe that the dJTFS-max outperforms the dJTFS-avg and the JTFS for portamento recognition. It yields frame- and event-based F-measures of 0.4% and 0.1% higher than that of the dJTFS-avg; and generates frame- and event-based F-measure improvement of 2.8% and 2.2% as compared to the JTFS. However, it underperforms both the dJTFS-avg and the JTFS for acciaccatura and glissando recognition. For example, the dJTFS-max returns frame-based F-measures of 4.4% and 5.0%, and event-based F-measures of 4.1% and 7.5%, respectively, lower than those of the dJTFS-avg. One possible reason for the better performance of the dJTFS-avg and the JTFS over the dJTFS-max on these two techniques may be the instability of the dJTFS-max to noisy pitch changes within the technique. Take the glissando technique in Fig. 7 (a) for instance: although the direction of the glissando is upward, downward note changes exist inside the technique, e.g. the note change at around 2.5 s. For such cases, the dJTFS-max which extracts the direction with maximum spectral-temporal modulation energy may oscillate between upward and downward directions within the playing technique. In contrast, the portamento technique comprises smooth pitch changes where a direction change within the playing technique is less likely to happen. The dJTF-avg detects PETs regardless of their directions and meanwhile mitigates this instability.

For the comparison between the scattering representations and the baselines, the AdaTS+AdaTRS achieves frame-based F-measures of 4.3% higher and event-based F-measures of 8.8% lower than the FDM for detecting vibratos. The dJTFS-avg considerably outperforms the HMMs for portamento and glissando recognition, with frame-based F-measures improving 36.0% and 73.7%, and with event-based F-measures increasing 43.2% and 68.3%, respectively. Apart from the performance comparison of different methods, we also investigate

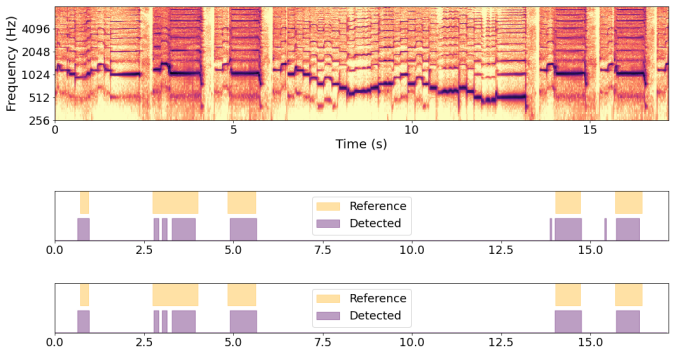

Fig. 8. Flutter-tongue detection result for an excerpt in the performance of *Morning* by Player 9. Top: log-frequency spectrogram; middle: comparison between the ground truth and frame-based classification output (frame-based $\mathcal{P}$=96%, $\mathcal{R}$=85%, $\mathcal{F}$=90%); bottom: comparison between the ground truth and obtained events after gap filling and minimum duration pruning (frame-based $\mathcal{P}$=99%, $\mathcal{R}$=85%, $\mathcal{F}$=91%; event-based $\mathcal{P}$=62%, $\mathcal{R}$=68%, $\mathcal{F}$=65%). The $\mathcal{P}$, $\mathcal{R}$, $\mathcal{F}$ values above are the results on this example.

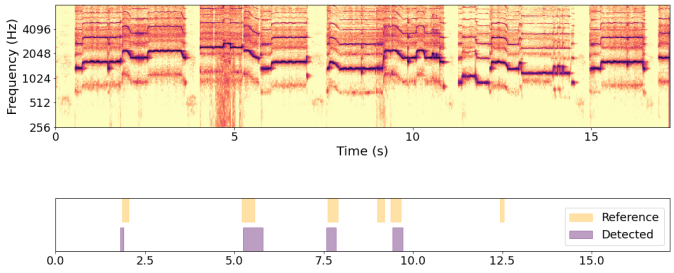

Fig. 9. Portamento detection result for an excerpt in Player 3's performance of *Busy Delivering Harvest*. Top: log-frequency spectrogram; bottom: comparison between the ground truth (upper half) and frame-based classification output (lower half). For this example, frame-based $\mathcal{P}$=85%, $\mathcal{R}$=52%, $\mathcal{F}$=65%.

the influence of gap filling and minimum duration pruning on the event-based evaluation by visualising the detection result. Fig. 8 top shows the log-frequency spectrogram of an excerpt in the CBFdataset; the middle and bottom subfigures display the detected flutter-tongue events before and after postprocessing, compared with the ground truth. Although the two gaps at around 3 s are not filled due to their long durations, the frame-based F-measure for this excerpt increases 1% after pruning the events at around 14 s and 16 s.

Cross checking the detection results with the original audio, we summarise the typical errors into three types below.

- Co-articulation: Fig. 9 shows an example portamento detection result in the CBFdataset compared to the ground truth. The false negative at around 9 s is an instance of portamento and flutter-tongue co-articulation. In such cases, portamento is no longer smooth but modulated with small ripples, making it hard to detect, even with 16 filters per octave in the first-order scattering transform.
- Rapid pitch change: This can be observed from the flutter-tongue recognition result in Fig. 8. The stable pitch regions are correctly detected with a precision of 96%. False negatives mostly happen during rapid note changes such as the gaps around 3 s.
- Techniques exhibit similar spectro-temporal patterns to non-techniques: for example, short portamento and note change. The false negative at 12.5 s in Fig. 9 is an instance of an undetected short protamento.

2) Multiclass classification: Fig. 10 (a) shows the frame-based F-measures of multiclass classification on the CBF-dataset using the proposed adaptive scattering transforms, the MFCCs baseline, and the MPS baseline. The macro F-measures over the techniques (including "other" cases) obtained using these three representations are 79.9%, 35.9%, and 52.0%, respectively. Fig. 11 (a) displays the number of frames detected for each class where "other" cases form the majority of the CBFdataset. This is because playing techniques are occasional events in real-world performances. Additionally, among the seven playing techniques, the number of samples in each class is highly unbalanced. We thus normalise the detection result over the number of class instances, as shown in Fig. 11 (b). The confusion between vibrato and tremolo is expected since frequency variations are commonly accompanied by amplitude modulations and vice versa, and there is no clear definition boundary between these two techniques.

B. Additional Datasets

Fig. 10 (b), (c), and (d) display the playing technique recognition results for the VPset, SOL dataset, and VocalSet, respectively. Note that these results are based on the same scattering transform hyperparameters (see Section VII-D) that we use for the CBFdataset. Parameter tuning for each dataset could potentially improve the recognition results. In the VPset, the proposed method yields an F-measure of 12.5% lower than that of the FDM for vibrato detection and considerably underperforms the HMMs for portamento detection. The most frequent errors found in portamento detection are note changes being detected as portamentos, which is consistent with the detection errors for CBF portamentos. For the SOL dataset, our proposed scattering representation achieves a macro F-measure of 75.5%, improving by 48.4% and 48.9% as compared to that using the MFCCs and MPS, respectively. For the VocalSet, our recognition system yields more stable F-measures across playing techniques than CNNs while generating a lower macro F-measure, i.e. 62.2% against 65.2% from the CNNs. CNNs failed to recognise any of the 20 spoken techniques. We provide confusion matrices on the SOL dataset and VocalSet in the supplementary material [14].

IX. Discussion and Perspectives

A. Discussion

Creating a consistent yet flexible taxonomy for playing techniques, either for instrumental or vocal techniques, is an under-explored area of music research. Playing techniques are musical patterns, which vary over instruments, regions, and performers. The same technique may exist under a different name in the context of another instrument or genre. For example, portamento in the VPset corresponds to glissando in the SOL dataset. The definition of playing techniques may also overlap depending on the player or singer performing it, e.g. trill and vibrato in the VocalSet [35]. Another observation is that, in the context of a music piece, playing techniques exhibit

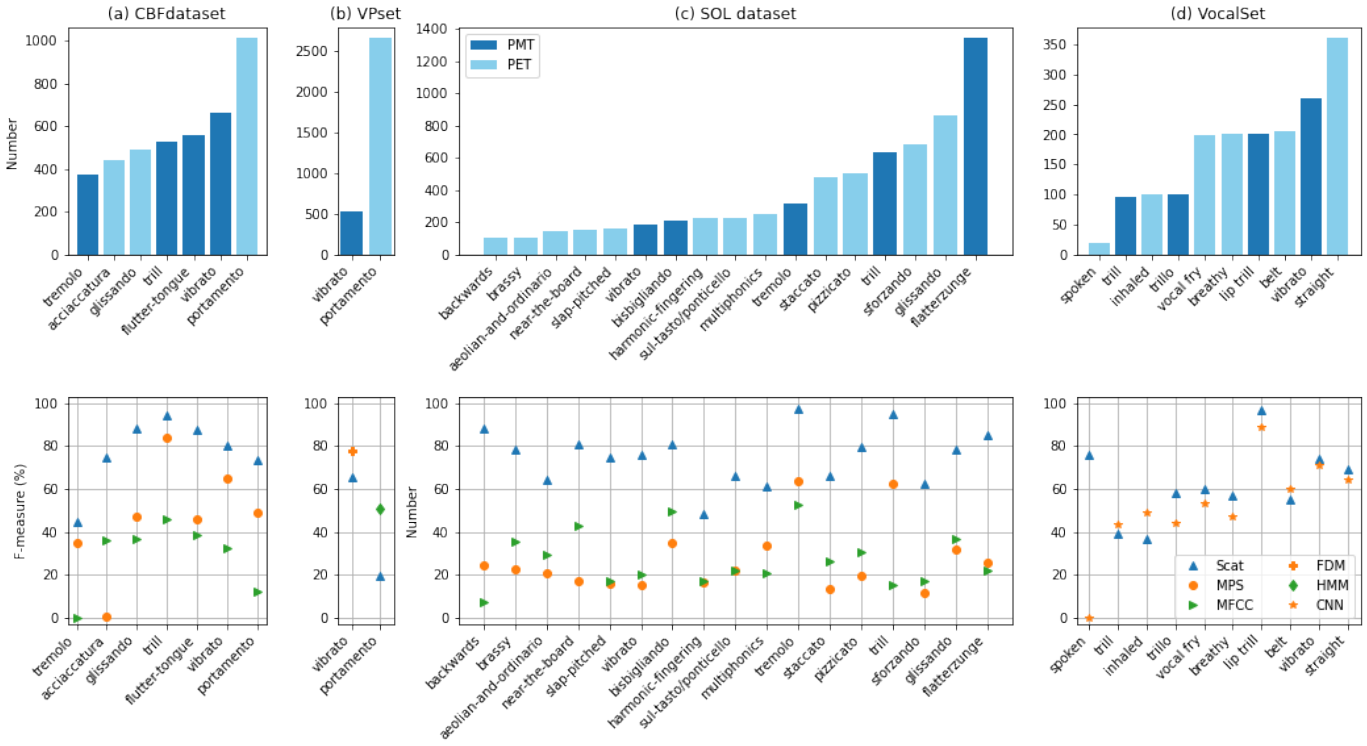

Fig. 10. Multiclass classification results on each dataset (binary classification on VPset). Top: the list of playing techniques and number of samples in each dataset. The techniques are grouped into pitch modulation-based techniques (PMTs, in blue) and pitch evolution-based techniques (PETs, in light blue). Bottom: the recognition results by multiclass classifiers for each dataset. The blue triangles are the F-measures obtained by the proposed scattering (Scat) representation while others are that of the baselines: modulation power spectrum (MPS), mel-frequency cepstral coefficients (MFCC), filter diagonalisation method (FDM), hidden Markov models (HMM), and convolutional neural networks (CNN).

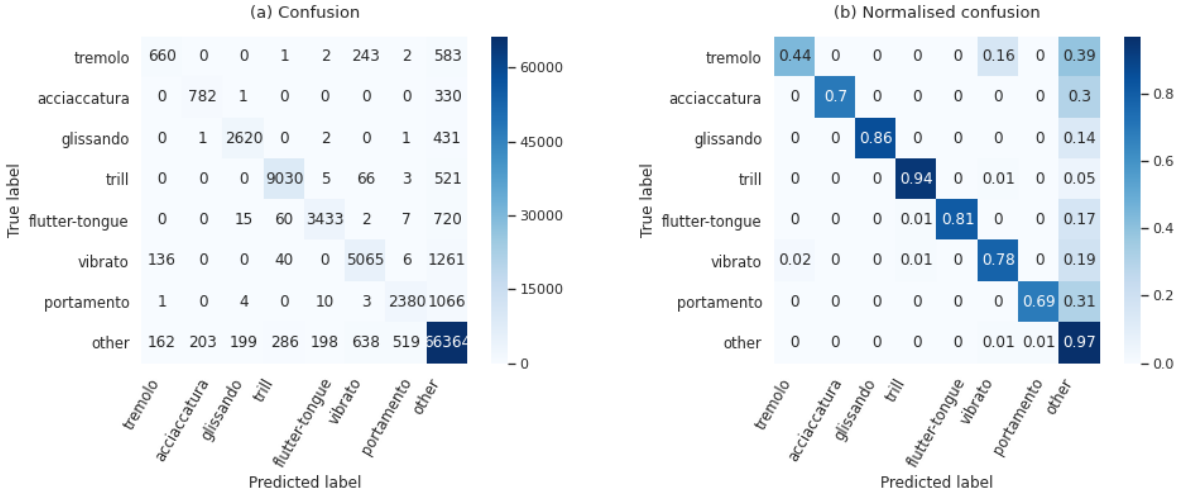

Fig. 11. Confusion matrices obtained for multiclass classification of the seven CBF techniques. (a): confusion matrix with number of frames detected; (b) normalised confusion matrix with values in (a) divided by the number of samples per technique.

considerable variations as compared to when they are played in isolation. Ecologically valid data requires the collection of full-length performances. We demonstrate the difference between isolated and performed techniques in the supplementary material [14] using glissando as an example. Isolated glissandos exhibit consecutive note changes with approximately equal durations, while performed ones possess variations on note duration, number of notes, and co-articulation. This is also confirmed by the performance differences of the recognition system on the SOL dataset (with isolated techniques) and other datasets (containing full pieces or long passages). The experiments in this paper all operate on the original datasets without any data augmentation. However, no evidence is found that the detection result relies highly on the number of samples (see Fig. 10). The techniques with many more samples do not achieve better results, e.g. the portamento technique in the CBFdataset and the straight technique in the VocalSet.

B. Limitations

For recognising PMTs, we calculate the frequency-adaptive scattering representations around the dominant frequency band, the band with maximum acoustic energy. This may not be robust when the dominant frequency band is noisy or not stable, e.g. the octave jumps at around 18 s in Fig. 5 (b). To suppress this effect, we could improve the system by smoothing the dominant band trajectory or by limiting the trajectory to the tonal range of the instrument. We could also consider other potential decomposition trajectories such as extracted fundamental frequency [36] or predominant melody [48]. Due to the lack of datasets with both polyphonic recordings and playing technique annotations, we have only evaluated the methodology on monophonic music. However, the entire pipeline is potentially applicable to polyphonic cases preprocessed by a source separation technique [49] or using a multi-pitch detection and instrument recognition method that could assign a pitch to a specific instrument [50]. Since co-articulations form a small portion of the CBFdataset, we conduct single-label multiclass classification by discarding the samples having more than one label. In practice, a user may expect a recognition system to detect all playing technique components in cases of co-articulation. In this case, a multilabel classifier would be the most appropriate choice.

C. Future Directions

We summarise the potential directions for future research into two groups: improvement of the methodology itself, and application of the methodology to either computational music analysis or music generation. Trainable scattering [51] is one example in the first group. We could tune the hyperparameters of the scattering transform and the classifier jointly for each type of playing technique, or for a specific instrument or genre. One may also apply recurrent classifiers such as long short-term memory units [9] to account for temporal context. The scattering operators discussed in this paper, the frequency-adaptive scattering and the dJTFS, capture variations along the time, modulation rate, and acoustic frequency axes, with the aim to detect PMTs and PETs. The flexibility of the scattering transform suggests that another direction would be to expand the framework by developing new operators or adding other existing operators to make the system as general-purpose as possible. For example, spiral scattering [42] is such an instance which captures variations across harmonics. This may provide useful information for recognising playing techniques characterised by harmonic variations, such as multiphonics.

The local invariance of the adaptive scattering transforms to time-shifts, time-warps, and frequency-transpositions may also be attractive to other music signal analysis tasks, such as music structure analysis, genre recognition, instrument recognition, and music transcription. Motivated by the observation in Fig. 5 (c) that the second-order scattering transform carries information on the modulation rate, we can use the scattering transform as a tool for playing technique modelling. Fig. 12 shows an example of modelling the modulation rate of a trill played on G6-A6. A clear harmonic partial appears between 5 and 8 Hz, which indicates the range of the modulation rate.

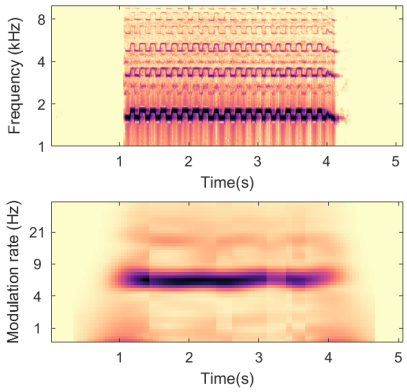

Fig. 12. Example of a trill modelling played on G6-A6. Top: log spectrogram; bottom: adaptive time scattering feature before log-normalisation.

Playing technique recognition and modelling can greatly help music synthesis systems generate realistic sounds that account for acoustic variations due to the exercise of a variety of instrumental or vocal techniques. A music style transformer [52] or note ornamentor is also possible since playing techniques carry important information regarding musical styles. Remodelling a straight note based on a playing technique or articulation of a professional player, or synthesising playing techniques that go beyond real instrument limitations present other attractive directions for further exploration, for example creating a flutter-tongue effect for piano.

X. Conclusions

In this paper, we have proposed a general framework based on the scattering transform for representing playing techniques in music signals. Two scattering operators, the frequency-adaptive scattering and the direction-adaptive joint time–frequency scattering, are introduced and we publicly release a new real-world dataset for playing technique analysis in context. Using the proposed representations as input, we evaluate the system over different datasets encompassing a variety of vocal and instrumental techniques and obtain promising results. We conclude that the scattering transform offers a versatile and compact representation for analysing playing techniques in performed music, and opens up new avenues for computational research in music signal analysis.

Acknowledgement

C. Wang is funded by the China Scholarship Council (CSC). V. Lostanlen is partially supported by the TrAcS grant from Atlanstic2020. E. Chew is supported by the European Union's Horizon 2020 research and innovation program (Grant no.788960) under the ERC ADG project COSMOS. We thank Meinard Müller and Joakim Andén for their valuable advice.

References

[1] M. Gainza and E. Coyle, "Automating ornamentation transcription," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, Honolulu, Hawaii, USA, 2007, pp. I–69.

[2] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *arXiv:1609.03499*, 2016.

[3] Y. Han and K. Lee, "Hierarchical approach to detect common mistakes of beginner flute players," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2014, pp. 77–82.

[4] G. E. Hall, H. Ezzaidi, M. Bahoura, and C. Volat, "Classification of pizzicato and sustained articulations," in *Proc. Eur. Signal Process. Conf.*, Marrakech, Morocco, Sep. 2013, pp. 1–5.

[5] V. Lostanlen, J. Andén, and M. Lagrange, "Extended playing techniques: the next milestone in musical instrument recognition," in *Proc. Int. Conf. Digital Libraries Musicology*, Paris, France, Sep. 2018, pp. 1–10.

[6] L. Yang, "Computational modelling and analysis of vibrato and portamento in expressive music performance," Ph.D. dissertation, Queen Mary Univ. of London, London, UK, 2017.

[7] S. Mallat, "Group invariant scattering," *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.

[8] R. Leonarduzzi, G. Rochette, J.-P. Bouchaud, and S. Mallat, "Maximum-entropy scattering models for financial time series," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Barcelona, Spain, May 2019, pp. 5496–5500.

[9] P. A. Warrick, V. Lostanlen, and M. N. Homsi, "Hybrid scattering-LSTM networks for automated detection of sleep arousals," *J. Physiological Measurement*, vol. 40, no. 7, p. 074001, 2019.

[10] M. Andreux and S. Mallat, "Music generation and transformation with moment matching-scattering inverse networks." in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Paris, France, Sep. 2018, pp. 327–333.

[11] J. Andén, V. Lostanlen, and S. Mallat, "Joint time–frequency scattering," *IEEE Trans. Signal Process.*, vol. 67, no. 14, pp. 3704–3718, Jul. 2019.

[12] C. Wang, E. Benetos, V. Lostanlen, and E. Chew, "Adaptive time–frequency scattering for periodic modulation recognition in music signals," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Delft, The Netherlands, Nov. 2019, pp. 809–815.

[13] C. Wang, V. Lostanlen, E.Benetos, and E. Chew, "Playing technique recognition by joint time–frequency scattering," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Barcelona, Spain, May 2020, pp. 881–885.

[14] C. Wang, E. Benetos, V. Lostanlen, and E. Chew, "Supplementary material for the article: Adaptive scattering transforms for playing technique recognition," 2021, [Online]. Available: https://changhongw.github.io/publications/TASLP_supplementary_material.pdf.

[15] S. Giraldo and R. Ramírez, "Performance to score sequence matching for automatic ornament detection in jazz music," in *Proc. Int. Conf. New Music Concepts*, Treviso, Italy, Mar. 2015.

[16] T. H. Ozaslan and J. L. Arcos, "Legato and glissando identification in classical guitar," in *Proc. Sound Music Comput. Conf*, Barcelona, Spain, Oct. 2010, pp. 457–463.

[17] L. Reboursière, O. Lähdeoja, T. Drugman, S. Dupont, C. Picard-Limpens, and N. Riche, "Left and right-hand guitar playing techniques detection." in *Proc. Int. Conf. New Interfaces Musical Expression*, Arbor, Michigan, USA, May 2012.

[18] L. Su, L. F. Yu, and Y. H. Yang, "Sparse cepstral, phase codes for guitar playing technique classification." in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2014, pp. 9–14.

[19] Y. P. Chen, L. Su, and Y. H. Yang, "Electric guitar playing technique detection in real-world recording based on F0 sequence pattern recognition," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2015, pp. 708–714.

[20] J. Abeßer, H. Lukashevich, and G. Schuller, "Feature-based extraction of plucking and expression styles of the electric bass guitar," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 2290–2293.

[21] J. Abeßer and G. Schuller, "Instrument-centered music transcription of solo bass guitar recordings," *IEEE Trans. Audio, Speech, Lang. Process*, vol. 25, no. 9, pp. 1741–1750, 2017.

[22] J. C. Brown and P. Smaragdis, "Independent component analysis for automatic note extraction from musical trills," *J. Acoust. Soc. Am.*, vol. 115, no. 5, pp. 2295–2306, 2004.

[23] B. Liang, G. Fazekas, A. McPherson, and M. Sandler, "Piano pedaller: a measurement system for classification and visualisation of piano pedalling techniques," in *Proc. Int. Conf. New Interfaces Musical Expression*, Aalborg, Denmark, 2017, pp. 325–329.

[24] J. Charles, "Playing technique and violin timbre: Detecting bad playing," Ph.D. dissertation, Technological Univ. Dublin, Dublin, Ireland, 2010.

[25] L. Su, H. M. Lin, and Y. H. Yang, "Sparse modeling of magnitude and phase-derived spectra for playing technique classification," *IEEE Trans. Audio, Speech, Lang. Process*, vol. 22, no. 12, pp. 2122–2132, 2014.

[26] C. W. Wu and A. Lerch, "On drum playing technique detection in polyphonic mixtures." in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, NY, USA, Aug. 2016, pp. 218–224.

[27] P. Herrera, A. Yeterian, and F. Gouyon, "Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques," in *Int. Conf. Music Artificial Intelligence*, Edinburgh, UK, Sep. 2002, pp. 69–80.

[28] J. F. Ducher and P. Esling, "Folded CQT RCNN for real-time recognition of instrument playing techniques," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Delft, The Netherlands, Nov. 2019, pp. 708–714.

[29] P. Jancovic, M. Köküer, and W. Baptiste, "Automatic transcription of ornamented Irish traditional flute music using hidden Markov models," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Malaga, Spain, Oct. 2015, pp. 756–762.

[30] D. Menzies and A. McPherson, "Highland piping ornament recognition using dynamic time warping," in *Proc. Int. Conf. New Interfaces Musical Expression*, Rouge, LA, USA, May 2015, pp. 50–53.

[31] Y. F. Huang, J. I. Liang, I. C. Wei, and L. Su, "Joint analysis of mode and playing technique in guqin performance with machine learning," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Oct. 2020, pp. 85–92.

[32] T. H. Özaslan, X. Serra, and J. L. Arcos, "Characterization of embellishments in ney performances of makam music in Turkey," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Porto, Portugal, Oct. 2012, pp. 13–18.

[33] C. Wang, E. Benetos, X. Meng, and E. Chew, "HMM-based glissando detection for recordings of Chinese bamboo flute," in *Proc. Sound Music Comput. Conf*, Malaga, Spain, May 2019, pp. 545–550.

[34] A. Neocleous, G. Azzopardi, C. N. Schizas, and N. Petkov, "Filter-based approach for ornamentation detection and recognition in singing folk music," in *Int. Conf. Computer Analysis Images Patterns*, 2015, pp. 558–569.

[35] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "Vocalset: A singing voice dataset," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Paris, France, Sep. 2018, pp. 468–474.

[36] M. Mauch and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Florence, Italy, May 2014, pp. 659–663.

[37] B. L. Sturm and P. Noorzad, "On automatic music genre recognition by sparse representation classification using auditory temporal modulations," in *Proc. Int. Symposium Computer Music Modeling and Retrieval*, 2012, pp. 379–394.

[38] E. Thoret, P. Depalle, and S. McAdams, "Perceptually salient regions of the modulation power spectrum for musical instrument identification," *Frontiers in psychology*, vol. 8, no. 587, 2017.

[39] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4114–4128, 2014.

[40] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT Press, 2016.

[41] C. Baugé, M. Lagrange, J. Andén, and S. Mallat, "Representing environmental sounds using the separable scattering transform," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8667–8671.

[42] V. Lostanlen and S. Mallat, "Wavelet scattering on the pitch spiral," in *Proc. Int. Conf. Digital Audio Effects*, Trondheim, Norway, Nov. 2015, pp. 429–432.

[43] S. Mallat, *A wavelet tour of signal processing. Third Edition: The sparse way*. Academic Press, 2008.

[44] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009.

[45] F. Albu and D. Martinez, "The application of support vector machines with Gaussian kernels for overcoming co-channel interference," in *Proc. IEEE Signal Process. Soc. Workshop*, 1999, pp. 49–57.

[46] M. Müller, *Fundamentals of music processing*: *Audio, analysis, algorithms, applications*. Springer, 2015.

[47] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir_eval: A transparent implementation of common mir metrics," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Oct. 2014, pp. 367–372.

[48] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Trans. Audio, Speech, Lang. Process*, vol. 20, no. 6, pp. 1759–1770, 2012.

[49] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-Unmix - A reference implementation for music source separation," *Journal of Open Source Software*, vol. 4, no. 41, p. 1667, 2019.

[50] D. Giannoulis and A. Klapuri, "Musical instrument recognition in polyphonic audio using missing feature approach," *IEEE Trans. Audio, Speech, Lang. Process*, vol. 21, no. 9, pp. 1805–1817, 2013.

[51] F. Cotter and N. Kingsbury, "A learnable scatternet: Locally invariant convolutional layers," in *IEEE Int. Conf. Image Process.*, Sep. 2019, pp. 350–354.

[52] S. Dai, Z. Zhang, and G. G. Xia, "Music style transfer: A position paper," *arXiv preprint arXiv:1803.06841*, 2018.