Hierarchical Classification for Instrument Activity Detection in Orchestral Music Recordings

Michael Krause and Meinard Müller, *Fellow, IEEE*

***Abstract*—Instrument activity detection is a fundamental task in music information retrieval, serving as a basis for many applications, such as music recommendation, music tagging, or remixing. Most published works on this task cover popular music and music for smaller ensembles. In this article, we embrace orchestral and opera music recordings as a rarely considered scenario for automated instrument activity detection. Orchestral music is particularly challenging since it consists of intricate polyphonic and polytimbral sound mixtures where multiple instruments are playing simultaneously. Orchestral instruments can naturally be arranged in hierarchical taxonomies, according to instrument families. As the main contribution of this article, we show that a hierarchical classification approach can be used to detect instrument activity in our scenario, even if only few fine-grained, instrument-level annotations are available. We further consider additional loss terms for improving the hierarchical consistency of predictions. For our experiments, we collect a dataset containing 14 hours of orchestral music recordings with aligned instrument activity annotations. Finally, we perform an analysis of the behavior of our proposed approach with regard to potential confounding errors.**

***Index Terms*—Hierarchical classification, instrument activity detection, orchestral music, music processing, music information retrieval.**

I. Introduction

INSTRUMENT recognition is a long-studied task in the field of music information retrieval (MIR), which aims at identifying the musical instruments that are playing in an audio excerpt. It is a difficult task, since many instruments produce sounds in overlapping pitch ranges and may exhibit similar timbral characteristics, especially those from the same instrument family. Furthermore, in real music recordings, multiple instruments may be active simultaneously (also called polyphonic instrument recognition). The task is closely related to instrument activity detection (IAD), where the aim is to identify the active instruments in a frame-wise fashion, over the course of an entire music recording. IAD can be useful to inform music recommendation or auto tagging systems, as well as aid in music editing and remixing.

Manuscript received 10 February 2023; revised 21 June 2023; accepted 21 June 2023. Date of publication 3 July 2023; date of current version 12 July 2023. This work was supported by German Research Foundation under Grants DFG MU 2686/7-2 and MU 2686/11-2. The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Zafar Rafii. *(Corresponding author: Michael Krause.)*

The authors are with the International Audio Laboratories Erlangen, Friedrich-Alexander-Universitat Erlangen-Nurnberg, 91058 Erlangen, Germany (e-mail: michael.krause@audiolabs-erlangen.de; meinard.mueller@audiolabs-erlangen.de).

Digital Object Identifier 10.1109/TASLP.2023.3291506

In this article, we are concerned with IAD in complex orchestral and opera recordings. Orchestral music in general constitutes a very challenging scenario for instrument detection, as many individual instruments are playing simultaneously, creating a highly complex sound mixture. In addition to the high degree of polyphony, orchestral music is also polytimbral, i.e., sounds from various instrument groups merge to create a single texture of sound. To approach this scenario, we utilize the hierarchical relationships that exist between orchestral instruments and instrument families, as illustrated in Fig. 1. In this context, we show that hierarchical classification improves results of a deep neural network for IAD, especially when fine-grained, instrument-level annotations are unavailable, but coarse, family-level annotations exist. Moreover, we investigate the consistency of predictions across hierarchy levels and demonstrate how additional training losses can promote consistent predictions, while preserving detection quality.

In contrast to popular music settings, public datasets with instrument activity annotations for orchestral or opera music rarely exist. For our experiments, we thus collect a dataset based on a combination of existing multi-track datasets and semi-automatically annotated commercial recordings. In total, our dataset consists of 14 hours of real-life orchestral recordings and covers 18 different classes. We make the instruments annotations for these recordings publicly available.[1]

A common pitfall of music classification systems is over-reliance on confounding factors in training and test data, which may lead to poor generalization ability. A system for genre classification may, for example, make decisions based on inaudible artifacts rather than musical content [1]. Such confounding effects may also arise for our IAD system by, e.g., affecting predictions for classes that are often active simultaneously (such as brass and woodwinds). To explore the impact of such effects on our system, we perform an analysis of model predictions with regard to classes that composers often use in conjunction.

We now summarize the main contributions of this article. First, we introduce a challenging new setting for IAD, for which no standard datasets exist. Second, we show how one can improve detection results and reduce the need for instrument-level annotations by exploiting the hierarchical class structure of our scenario. Third, we show how the consistency of predictions made by our model can be improved through additional loss terms. Fourth, we perform an analysis of our model's behavior and uncover confounding effects for certain instrument classes.

[1][Online]. Available: https://www.audiolabs-erlangen.de/resources/MIR/2023-TASLP-HierarchicalInstrumentClass

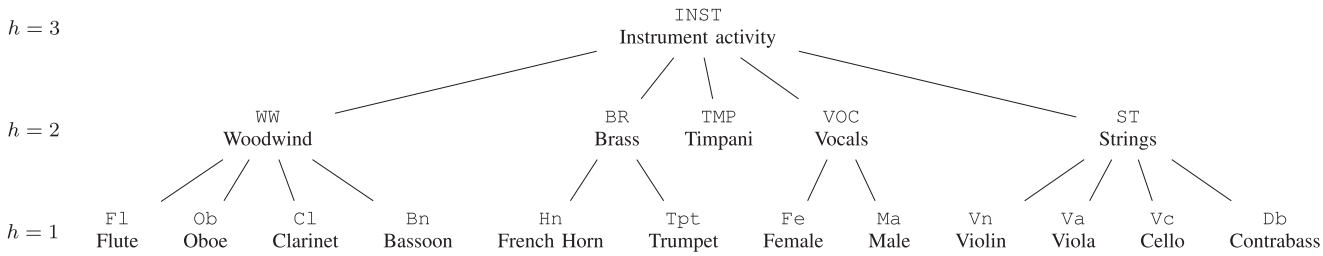

Fig. 1. Class hierarchy as used in this paper. The number of hierarchy levels is $H = 3$. Level $h = 2$ corresponds to instrument families, while level $h = 1$ corresponds to fine-grained instrument classes.

In a previous work [2], we explored hierarchical classification for detecting singing activity, singer gender and voice type in orchestral recordings. In terms of the hierarchical classification techniques used, this article is an extended version of [2]. Here, we go beyond [2] by considering an instrument detection scenario, involving a different and larger class hierarchy compared to [2]. Furthermore, we present additional technical details, use different datasets and models, and provide extensive additional experiments and analyses.

The remainder of the article is organized as follows: Section II discusses related work on instrument recognition, hierarchical classification, and analysis of orchestra recordings. In Section III, we formalize our problem statement, outline our main classification approach, and describe evaluation measures used. Sections IV and V cover our dataset and model architecture, respectively. Section VI contains the main experimental results. In Section VII, we describe and evaluate losses for improving the consistency of predictions. In Section VIII, we analyze our model with regard to confounding effects among instrument classes. Finally, Section IX concludes the article with an outlook on possible future work.

II. Related Work

Our article draws upon related work from several fields, including the vast field of music instrument classification. In our review of these fields, we focus on key references relevant to the present paper and relate our contributions to the state of the art.

A. Instrument Detection

Early work on automatic musical instrument classification dealt with recordings of isolated note events and used classical machine learning techniques [3], [4], [5]. Other works considered real music recordings, but restricted themselves to a single instrument playing [6], [7], [8], a scenario called monophonic instrument recognition. In contrast, polyphonic instrument recognition attempts to recognize instruments within mixtures where several instruments are playing simultaneously. This has been approached with classical machine learning techniques [8], [9], [10], [11], [12], [13], [14] and, more recently, with deep learning [15], [16], [17], [18], [19], [20].

Works on instrument recognition can also be categorized according to whether only the predominant instrument in a mixture (e.g., [13], [15]) or all active instruments (e.g., [16]) are to be recognized. Furthermore, some works classify activity for an entire audio excerpt lasting several seconds (e.g., [13], [14], [15], [18], [19], [20], [21]) whereas more fine-grained approaches yield predictions on a frame-level (e.g., [16], [17]). Such frame-level outputs can be used to obtain instrument predictions for every time step in a music recording—also called instrument activity detection (IAD). The scenario considered in this article is polyphonic IAD and considers all instruments playing. In contrast to prior work on this scenario, which usually examines popular music or works for small ensembles, we consider complex orchestral and opera music.

B. Hierarchical Classification for Audio

Some previous works have used hierarchical class structures for classification of audio data. For example, the authors in [22] propose a specialized network architecture for classifying bird calls, based on bird taxonomies. They perform classification on an excerpt- and not on a frame-level. In [23], a network for sound event detection is iteratively pretrained on successive hierarchy levels. Some papers [24], [25] employ tree hierarchies for audio representation learning (but not for classification). These works also usually do not take into account audio inputs where several classes may be active at the same time.

Fewer works use hierarchical structures for music audio classification. In [26], hierarchical classification is used in the context of singing transcription. Essid et al. [7] use hierarchies for instrument classification, but their scenario involves only synthetic audio data and their system does not yield predictions on a frame-level (which are necessary for IAD). In our previous work [2], which we extend here, we explored hierarchical classification for singing detection (see also Section I).

Hierarchical structures have also been exploited for other tasks in MIR, including musical instrument separation [27]. Garcia et al. [28] use instrument hierarchies for few-shot detection, where the model requires examples of the target class at test time (see also [29]). In contrast, we use a classification approach with a fixed set of classes, since the instruments in orchestral recordings are known a-priori. Nolasco and Stowell [30] employ instrument hierarchies for audio representation learning (not for classification).

Incorporating hierarchical structures in machine classifiers has also been a topic in the wider machine learning literature, see, e.g., [31], [32], [33], [34], [35]. These works typically evaluate their proposed methods on small and artificial datasets.

In contrast, we perform hierarchical classification on a dataset of real orchestra and opera recordings.

C. Orchestra and Opera in MIR

Only few works in MIR have focused on opera and orchestral music. Among these are works on singing detection [2], [36], [37], emotion identification [38], predominant melody estimation [39], as well as source separation informed by multichannel recordings [40] or by score information [41]. Taenzer et al. [42] performed instrument family classification on classical music recordings, but they only considered monotimbral pieces (i.e., works for ensembles consisting of one family only). To the best of our knowledge, there rarely is work in MIR on IAD in real-world, polyphonic and polytimbral recordings of orchestra and opera.

III. Hierarchical Instrument Detection

In this article, we aim to utilize the hierarchical relationships among orchestral instruments. To this end, we now formalize the hierarchical class model, classification approach, and evaluation measures used throughout this work. In this section, we follow [2], where we introduced the concepts and notation.

A. Hierarchical Class Model

We write $\mathbf{C}$ for the set of all classes in our detection problem. We can partition these classes across a total of H hierarchy levels, where $\mathbf{C}^h \subseteq \mathbf{C}$ are the sets of classes at hierarchy level $h \in \{1, \dots, H\}$. Thus,

$$\mathbf{C} = \dot{\bigcup}_{h \in \{1,\dots,H\}} \mathbf{C}^h. \tag{1}$$

In our setting, we consider 18 different classes, corresponding to the nodes of the tree illustrated in Fig. 1. We use $H = 3$ hierarchy levels, with the lowest level $h = 1$ corresponding to fine-grained instrument classes, level $h = 2$ containing coarse instrument families, and the highest level $h = 3$ signifying any kind of instrument activity (as opposed to silence or noise). Thus, $\mathbf{C}^1 = \{\texttt{Fl}, \texttt{Ob}, \texttt{Cl}, \dots\}$, $\mathbf{C}^2 = \{\texttt{WW}, \texttt{BR}, \dots\}$, and $\mathbf{C}^3 = \{\texttt{INST}\}$. It should be noted that the hierarchy could also be constructed in alternative ways. Our choice here is motivated by practical considerations, i.e., simplicity and the availability of sufficient data for each class $c \in \mathbf{C}$. Generally, constructing appropriate instrument hierarchies can be a challenging problem, especially for electronic or non-standard instruments [43], [44].

B. Classification Approach

To approach IAD using a deep network, we pose the problem as a frame-wise, multi-label classification task. Thus, several instrument classes may be active simultaneously and we want to produce predictions for every frame in a music recording. Formally, we model both reference annotations and predictions as families of subsets of frames. We write $\mathcal{I}$ for the set of all audio frames in our test recordings. Now, our instrument annotations are given as families $(\mathcal{I}_c^{\text{Ref}})_{c \in \mathbf{C}}$ of subsets $\mathcal{I}_c^{\text{Ref}} \subseteq \mathcal{I}$, with $\mathcal{I}_c^{\text{Ref}}$ containing all frames where class $c \in \mathbf{C}$ is active. Note that the sets $\mathcal{I}_c^{\text{Ref}}$ are generally not disjoint, e.g., in cases where instruments are playing at the same time. Frames with silence or noise are not contained in any $\mathcal{I}_c^{\text{Ref}}$.

In a similar fashion, we model the estimates made by our IAD system as families of sets $(\mathcal{I}_c^{\text{Est}})_{c \in \mathbf{C}}$. These outputs are obtained from a deep network that takes an audio excerpt as input and yields predictions for the center frame of that excerpt. These estimates are values in $[0, 1]$ for every class $c \in \mathbf{C}$, which are subsequently thresholded to obtain the sets $\mathcal{I}_c^{\text{Est}}$. Since this approach jointly considers all hierarchy levels $1 \leq h \leq H$, we will refer to it as **HC** (hierarchical classification).[2] More details on the network architecture used are provided in Section V.

C. Evaluation Measures

With our formulation above, we can define some classical metrics used for evaluating detection systems, such as frame-wise precision, recall, and F-measure for each class $c \in \mathbf{C}$:

$$P_c = \frac{|\mathcal{I}_c^{\text{Ref}} \cap \mathcal{I}_c^{\text{Est}}|}{|\mathcal{I}_c^{\text{Est}}|}, R_c = \frac{|\mathcal{I}_c^{\text{Ref}} \cap \mathcal{I}_c^{\text{Est}}|}{|\mathcal{I}_c^{\text{Ref}}|}, F_c = \frac{2 \cdot P_c \cdot R_c}{P_c + R_c}. \tag{2}$$

Intuitively, P_c is the fraction of frames that are correctly predicted as belonging to class c, R_c refers to the fraction of ground truth frames of class c that are correctly identified, and F_c is an average of the two. Note that the three measures may be overly optimistic if c is a very common class that is active in most frames (as is the case, e.g., for $c = \texttt{INST}$). In this case, it is important to additionally consider the specificity for class c, i.e., the recall of non-active frames, defined as

$$S_c = \frac{|(\mathcal{I} \setminus \mathcal{I}_c^{\text{Ref}}) \cap (\mathcal{I} \setminus \mathcal{I}_c^{\text{Est}})|}{|\mathcal{I} \setminus \mathcal{I}_c^{\text{Ref}}|}, \tag{3}$$

where $\setminus$ denotes the set difference operator.

A classification approach that is aware of class hierarchies should not produce predictions that are hierarchically inconsistent. For example, a frame i may be classified as trumpet ($\texttt{Tpt}$), so $i \in \mathcal{I}_{\texttt{Tpt}}^{\text{Est}}$, but at the same time may not be classified as brass ($\texttt{BR}$), so $i \notin \mathcal{I}_{\texttt{BR}}^{\text{Est}}$. We will refer to this as a bottom-up inconsistency. Such an inconsistency makes the output of a detection system difficult to interpret, since it is unclear whether the frame was erroneously classified as trumpet or whether it does indeed contain brass, but the system failed to identify $\texttt{BR}$. Similarly, we may have a frame $i \in \mathcal{I}_{\texttt{BR}}^{\text{Est}}$ that is classified as brass but at the same time is neither classified as horn nor trumpet, thus $i \notin \mathcal{I}_{\texttt{Hn}}^{\text{Est}} \cup \mathcal{I}_{\texttt{Tpt}}^{\text{Est}}$. We call this a top-down inconsistency. Fig. 2 gives an illustration of these inconsistencies. Ideally, a detection system produces no inconsistencies, making its output straightforward to interpret.

We now define additional measures that capture different kinds of inconsistencies. For a class $c \in \mathbf{C}$, we write $c\uparrow$ for the parent of c in the hierarchy (e.g., $\texttt{Fl}\uparrow = \texttt{WW}$). Similarly, $c\downarrow$ contains the set of children of c (e.g., $\texttt{BR}\downarrow = \{\texttt{Hn}, \texttt{Tpt}\}$). Additionally, for a subset $\mathbf{C}' \subseteq \mathbf{C}$, we use the notation $\mathcal{I}_{\mathbf{C}'} = \cup_{c \in \mathbf{C}'} \mathcal{I}_c$. Now, for any $h > 1$ and $c \in \mathbf{C}^h$, we define measures

[2] This approach is called Strategy $D^{0,0}$ in [2].

Fig. 2. Illustration of (a) bottom-up and (b) top-down inconsistencies. Filled and empty circles correspond to classes predicted as active or inactive, respectively. Red circles indicate inconsistencies.

of bottom-up consistency, top-down consistency, and overall consistency as:

$$\gamma_c^{\uparrow} = \frac{|\mathcal{I}_c^{\text{Est}} \cap \mathcal{I}_{c\downarrow}^{\text{Est}}|}{|\mathcal{I}_{c\downarrow}^{\text{Est}}|}, \gamma_c^{\downarrow} = \frac{|\mathcal{I}_c^{\text{Est}} \cap \mathcal{I}_{c\downarrow}^{\text{Est}}|}{|\mathcal{I}_c^{\text{Est}}|}, \gamma_c = \frac{|\mathcal{I}_c^{\text{Est}} \cap \mathcal{I}_{c\downarrow}^{\text{Est}}|}{|\mathcal{I}_c^{\text{Est}} \cup \mathcal{I}_{c\downarrow}^{\text{Est}}|}. \tag{4}$$

All measures have values in the range $[0, 1]$. Intuitively, if there are no bottom-up inconsistencies, then $\gamma_c^{\uparrow} = 1$. Likewise, if there are no top-down inconsistencies, then $\gamma_c^{\downarrow} = 1$. Finally, $\gamma_c = 1$ if and only if $\mathcal{I}_c^{\text{Est}} = \mathcal{I}_{c\downarrow}^{\text{Est}}$. γ_c is also called intersection-over-union or Jaccard index. Note that the consistency measures can trivially be maximized by setting, e.g., $\mathcal{I}_c^{\text{Est}} = \mathcal{I}$ or $\mathcal{I}_c^{\text{Est}} = \emptyset$ for all $c \in \mathbf{C}$. However, obtaining good detection results (e.g., in terms of F-measure) while simultaneously preserving consistency is non-trivial.

IV. Orchestral Datasets

As mentioned in Section II, orchestral and opera music are seldom explored in MIR. In particular, no standard datasets for IAD on real orchestral recordings exist. The instrument classes commonly considered in IAD datasets for popular music (e.g., [47], which contains guitar, drums, base etc.) do not usually appear in orchestral music. We thus assembled our own datasets for training and evaluating our system, based on existing datasets and our own annotation efforts.

For effectively training our deep learning system, we require a dataset of several hours length. For such a size, manually annotating instrument activity for all 18 classes in our hierarchy would be prohibitively expensive. We thus consider two ways of obtaining orchestral recordings with aligned instrument annotations:

1) Use multi-track recordings of orchestral pieces, where activity annotations can easily be obtained from the individual tracks.
2) Use music synchronization techniques to align a score representation to an audio recording of a piece. Instrument activity in the recording can then be transferred from the aligned score.

A third possible option would be to use artificial recordings of pieces based on synthesized score representations. Recently, Sarkar et al. [48] released a dataset of synthesized, multi-track recordings of classical pieces. However, their dataset contains, for the most part, chamber music rather than full orchestra pieces. Their work also demonstrates that synthesizing convincing renditions of classical music is a challenging task in itself. Here, rather than creating artificial recordings of orchestral scores, we instead collect multiple real recordings per score for synchronization (option 2).

A. Multi-Track Datasets

Due to the challenges of recording orchestra pieces in a multi-track fashion[3], only a few such datasets have been released. One of these is Phenicx Anechoic [40], which contains clean multi-track recordings of four orchestral excerpts by different composers with note on- and offsets manually annotated for each track. We derive instrument activity from these annotations. Böhm et al. [46] provide multi-track recordings for three movements of Beethoven's Symphony No. 8, resulting in the longest multi-track dataset we use. The individual tracks are mostly free of cross-talk. We therefore use a simple energy thresholding procedure on the tracks to obtain instrument activity annotations.

Since both datasets are annotated based on clean multi-track recordings, we expect the derived activity labels for Phenicx and Beethoven Anechoic to be highly reliable. Nevertheless, there remains some ambiguity in defining note on- and offsets, especially for strings and woodwind instruments [49].

Prätzlich et al. [45] provide multi-track audio for three numbers from Carl-Maria von Weber's opera "Der Freischütz". However, due to their recording setup, the individual tracks incur a large amount of cross-talk coming from other sources. We obtained annotations for this dataset using music synchronization, see below.

B. Music Synchronization

Audio-to-score alignment techniques are used to temporally align a recorded music performance with the corresponding musical score. Once an alignment is obtained, information about instruments or pitches played can be transferred from the aligned score to the recorded performance. Here, we use audio-to-score alignment to transfer instrument activity from a symbolic score to several recorded performances of a piece. A popular dataset annotated in this fashion is MusicNet [50], which contains chamber rather than orchestra pieces. Note that automatic audio-to-score alignments may introduce annotation errors. We thus expect the resulting activity labels to be less reliable compared to those obtained from multi-track data.

Even though a large amount of MIDI files for classical music pieces can be found online, only few correspond to full orchestral scores with separate MIDI tracks per instrument. The lack of available score-data represents a big bottleneck for this approach to obtaining instrument annotations. For this work, we manually encoded a score representation of the first act of Richard Wagner's opera "Die Walküre" (requiring several months of work for musically trained annotators). For some other musical works—namely, several movements of Beethoven's Symphony No. 3, Dvorak's Symphony No. 9, and Tschaikowsky's Violin Concerto—we obtained clean orchestral scores from the Mutopia project.[4] We choose these works because they each contain many instruments from the hierarchy we employ. Furthermore, they belong to the classical and romantic periods and are thus stylistically similar to the remaining pieces we use. We

[3]Orchestra musicians usually perform within the same room and in close proximity, making it very difficult to obtain clean tracks without cross-talk coming from other instruments playing simultaneously.

[4][Online]. Available: https://www.mutopiaproject.org/

TABLE I
RECORDINGS OF ORCHESTRAL AND OPERA WORKS USED IN THIS PAPER

Subset/Composer	Work	# Versions	Dur. (min)
Ours			
Wagner	Die Walküre, Act 1	6	389
Beethoven	Symphony No. 3, Mvmt. 1	1	17
	Mvmt. 2	5	72
	Mvmt. 3	5	29
	Mvmt. 4	5	57
Dvorak	Symphony No. 9, Mvmt. 1	5	44
	Mvmt. 2	5	56
	Mvmt. 4	1	11
Tschaikowsky	Violin Concerto, Mvmt. 1	5	91
	Mvmt. 2	5	32
	Mvmt. 3	1	10
Freischütz Digital [45]			
Weber	Der Freischütz, No. 6	1	5
	No. 8	1	9
	No. 9	1	7
Phenicx Anechoic [40]			
Mozart	Aria from Don Giovanni	1	4
Beethoven	Symphony No. 7, Mvmt. 1 (Excerpt)	1	3
Bruckner	Symphony No. 8, Mvmt. 2 (Excerpt)	1	1
Mahler	Symphony No. 1, Mvmt. 4 (Excerpt)	1	2
Beethoven Anechoic [46]			
Beethoven	Symphony No. 8, Mvmt. 1	1	8
	Mvmt. 2	1	4
	Mvmt. 4	1	7
			858

For some pieces, multiple versions are available. The last column gives the total duration of all versions for a piece. Freischütz Digital, Phenicx Anechoic, and Beethoven Anechoic are existing multi-track datasets.

then obtained six orchestral audio versions (i.e., performances, recordings) of these pieces from commercial CD releases and created instrument activity annotations using a state-of-the-art score-to-audio synchronization pipeline [51], [52]. Care had to be taken to verify that there are no structural differences between score and recorded versions, which would corrupt the alignment results. Thus, the annotation process can be considered as a semi-automatic approach, where one must expect alignment errors in the order of 0.2 s [53]. Such errors need to be kept in mind when interpreting evaluation results.

C. Dataset Overview and Split

An overview of all recordings used in this work is given in Table I. The multi-track datasets we use each contain one recording per piece and contribute a total duration of around 50 minutes of orchestral music. For the pieces annotated via music synchronization, we have several versions per piece. In total, our dataset contains roughly 14 hours of orchestral music, making it amenable to deep learning. We make all instrument activity annotations publicly available on our accompanying website.[5]

Note that not all instrument classes are present in every recording and that there is a large imbalance among activity of different classes. We define the fraction δ_c of frames where class $c \in \mathbf{C}$ is active as

$$\delta_c = \frac{|\mathcal{I}_c^{\text{Ref}}|}{|\mathcal{I}^{\text{Ref}}|} \in [0,1]\,. \tag{5}$$

Additionally, for any $h < H$ and $c \in \mathbf{C}^h$, we denote the fraction of items where both c and some other class $c' \in \mathbf{C}^h$ of the same

[5]Link provided in Section I.

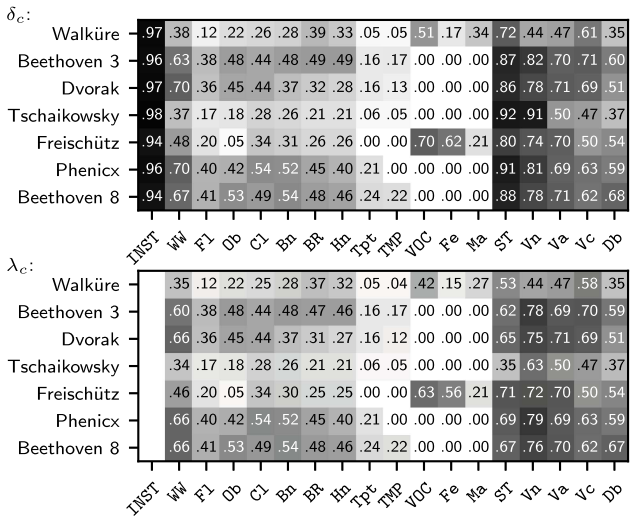

Fig. 3. Activity (δ_c, upper matrix) and multi-labeledness (λ_c, lower matrix) of instrument classes for subsets of our dataset, relative to the total length of recordings in that subset.

hierarchy level are active by

$$\lambda_c = \frac{1}{|\mathcal{I}^{\text{Ref}}|}\left|\cup_{c' \in \mathbf{C}^h \setminus \{c\}} \mathcal{I}_c^{\text{Ref}} \cap \mathcal{I}_{c'}^{\text{Ref}}\right|. \tag{6}$$

Note that $0 \leq \lambda_c \leq \delta_c$. Intuitively, λ_c captures the amount of "multi-labeledness" for class c. Fig. 3 shows the values of δ_c and λ_c for each class c in the different subsets of our dataset. We can observe that strings and woodwind instruments are the most common classes. Only "Die Walküre" and Freischütz Digital contain singing. For most classes and pieces, there is a large

amount of joint activity among classes on the same hierarchy level, indicated by λ_c being close to δ_c. A notable exception is Vn in the Tschaikowsky recordings ($\delta_c = 0.92, \lambda_c = 0.35$), due to the many solo parts of the violin in this concerto.

To train and evaluate our IAD system, we split our dataset into train and test recordings. We put different movements into train and test in order to investigate whether our models are capable of generalizing to new musical content or whether they overfit to specific compositions (see also [54]). We also choose different versions in train and test to control for varying recording characteristics and aspects of interpretation (reducing the impact of the so-called album-effect [55]). Among the multi-track datasets, we select No. 6 of Freischütz Digital, the second movement of Beethoven's Symphony No. 8, and all recordings in Phenicx Anechoic for our test set. From the remaining pieces, we choose the first movement of Beethoven's Symphony No. 3, the fourth movement of Dvorak's Symphony No. 9 and the third movement of Tschaikowsky's Violin Concerto for testing. Since we do not have multiple opera works that could be distributed into train and test sets, we choose an excerpt of the Wagner opera act (measures 697 to 955, corresponding to around twelve minutes of music), omit this excerpt during training, and use it for testing. In all cases, we choose one version for testing and use the remaining five versions for training.

V. Model Architecture

In this section, we give details on the architecture of our model for hierarchical classification. Note that the main technical focus of our work is on hierarchical structures and consistency losses rather than a particular architecture and alternative architectures (e. g., based on ResNets [56]) could also be used here. We employ a convolutional neural network (CNN) inspired by standard VGG-like CNN architectures [57]. The architecture is illustrated in Table II. The network takes a harmonic CQT representation (HCQT, [58]) of an audio excerpt as input and outputs a vector of 18 values in $[0, 1]$, corresponding to activity of the 18 classes in $\mathbf{C}$ predicted for the center frame of the input excerpt.

The HCQT input consists of 201 frames (roughly 4.7 seconds), computed using a hop-size of 512 on recordings sampled at 22 050 Hz (i. e., frame rate of 43 Hz). The constant-Q spectrum ranges from C1 to B7 with three bins per pitch, meaning 252 bins in total. For the harmonic CQT, five harmonic representations (including one subharmonic) are stacked in channel dimension. The final input tensor has a size of (201, 252, 5).

The network consists of three stages, separated by doubled lines in the table. Inspired by [59], [60], we first process each input tensor with a large pre-filtering kernel of size (15, 15) and strides (1, 1), followed by a kernel of size and stride (1, 3), i.e., the kernel is applied in pitch direction only. The resulting intermediate feature map has a pitch axis with a single bin per pitch. The second stage of our network applies three conv-conv-pool processing blocks, as in [57], [61]. In the final stage, we use a convolutional filter of size (5, 1) and stride (1, 1) to aggregate temporal context (as in [59]) and apply several dense layers to obtain the final output. We further use batch normalization after each learnable layer and apply dropout before dense layers. All layers are followed by a leaky ReLU activation, except for the final dense layer, which is followed by a sigmoid.

TABLE II
Network Architecture Used for Our IAD System

Layer (Kernel size), (Strides)	Output Shape	Parameters
Input	(201, 252, 5)	
Conv2D (15, 15), (1, 1)	(201, 252, 64)	72 000
Batch normalization	(201, 252, 64)	256
Conv2D (1, 3), (1, 3)	(201, 84, 64)	12 288
Batch normalization	(201, 84, 64)	256
Conv2D (3, 3), (1, 1)	(199, 82, 64)	36 864
Batch normalization	(199, 82, 64)	256
Conv2D (3, 3), (1, 1)	(197, 80, 64)	36 864
Batch normalization	(197, 80, 64)	256
MaxPool2D (3, 3), (3, 3)	(65, 26, 64)	
Conv2D (3, 3), (1, 1)	(63, 24, 128)	73 728
Batch normalization	(63, 24, 128)	512
Conv2D (3, 3), (1, 1)	(61, 22, 128)	147 456
Batch normalization	(61, 22, 128)	512
MaxPool2D (3, 3), (3, 3)	(20, 7, 128)	
Conv2D (3, 3), (1, 1)	(18, 5, 256)	294 912
Batch normalization	(18, 5, 256)	1024
Conv2D (3, 3), (1, 1)	(16, 3, 256)	589 824
Batch normalization	(16, 3, 256)	1024
MaxPool2D (3, 3), (3, 3)	(5, 1, 256)	
Conv2D (5, 1), (1, 1)	(1, 1, 512)	655 360
Batch normalization	(1, 1, 512)	2048
Squeeze	(512)	
Dropout 0.5	(512)	
Dense	(256)	131 328
Batch normalization	(256)	1024
Dropout 0.5	(256)	
Dense	(128)	32 896
Batch normalization	(128)	512
Dropout 0.5	(128)	
Dense	(18)	2322
Output: Sigmoid	(18)	

We train our network by minimizing a binary cross-entropy loss for a maximum of 1000 epochs (with 320 batches per epoch, each containing 32 input excerpts) using the Adam optimizer with a learning rate of 0.002. We additionally use early stopping by terminating training after the validation loss (evaluated on a randomly selected subset of the training set) has not decreased for 15 epochs. We further half the learning rate after 10 epochs without improvement of the validation loss. We use label smoothing inside the cross-entropy loss as regularization (thus, 0-labels are replaced with 0.02 and 1-labels are replaced by 0.98).

Each of the 252 bins in the input excerpts is individually normalized to be zero-mean and unit-variance (for this, mean and standard deviation per bin are estimated on the training set). As is common practice [42], [62], [63], [64], we augment training excerpts by randomly applying time warping, pitch shifting, masking of time-frequency bins, adding random noise, or applying random equalization. Training the model on our dataset takes around 14 hours on an RTX 2080 Ti

At test time, we evaluate our network for every frame in the test recordings, apply a median filter of length 0.5 seconds on the sequence of predictions for each class, and then downsample to a feature rate of 5 Hz. This is common practice in musical

TABLE III
RESULTS FOR OUR **HC** (HIERARCHICAL CLASSIFICATION) APPROACH TO IAD ON OUR ORCHESTRAL DATASETS

	P	R	F	S	$\gamma^{\downarrow}$	$\gamma^{\uparrow}$	δ
INST	0.99	1.00	0.99	0.67	0.99	1.00	0.96
WW	0.86	0.87	0.87	0.80	0.92	1.00	0.59
Fl	0.76	0.62	0.69	0.90			0.35
Ob	0.75	0.73	0.74	0.85			0.38
Cl	0.74	0.70	0.72	0.82			0.42
Bn	0.74	0.76	0.75	0.80			0.42
BR	0.79	0.70	0.74	0.85	0.94	0.99	0.45
Hn	0.75	0.68	0.71	0.84			0.41
Tpt	0.76	0.50	0.60	0.97			0.16
TMP	0.79	0.57	0.66	0.98			0.11
VOC	0.93	0.87	0.90	0.99	0.96	0.99	0.12
Fe	0.96	0.81	0.88	1.00			0.06
Ma	0.90	0.87	0.88	0.99			0.06
ST	0.95	0.95	0.95	0.65	0.99	1.00	0.87
Vn	0.90	0.93	0.91	0.66			0.77
Va	0.81	0.88	0.84	0.64			0.64
Vc	0.85	0.88	0.87	0.74			0.64
Db	0.86	0.87	0.87	0.83			0.56
Avg. ($\mathbf{C}$)	0.84	0.79	0.81	0.83	0.96	1.00	
Avg. ($\mathbf{C}^2$)	0.86	0.79	0.82	0.85			
Avg. ($\mathbf{C}^1$)	0.82	0.77	0.79	0.84			

activity detection systems (e. g., [63], [65]) and makes our evaluation results more robust to annotation errors introduced by music synchronization (see Section IV). We finally binarize these predictions using a threshold of 0.5. Inference requires 3.5 seconds per minute of input audio (given pre-computed HCQT representations).

VI. MAIN RESULTS

In this section, we present the main evaluation results for our IAD system and, additionally, demonstrate that hierarchy information reduces the need for fine-grained labels during training.

We begin with our main experiment. We train the model described in Section V on the training subset of our dataset and subsequently evaluate it on the test set (according to the split described in Section IV). Recall that we choose a joint classification approach **HC** (short for hierarchical classification), i. e., the network outputs predictions for all 18 classes in $\mathbf{C}$, see also Section III.

Evaluation results on the test set are shown in Table III. Columns contain different metrics, computed over the entire test set (note that the consistency metrics γ are only defined for classes in $\mathbf{C}^h$ for $h > 1$). Rows correspond to different classes in $\mathbf{C}$, and the last three rows show averages over classes. In particular, we report averages for families ($\mathbf{C}^2$) and instruments ($\mathbf{C}^1$).[6]

Overall, evaluation results are moderately high with an average F-measure of 0.81 and specificity of 0.83. For comparison, Hung and Yang [16] achieve an average F-measure of 0.89 for IAD on recordings of small classical ensembles with seven different instrument classes. However, our results vary across classes. We can observe that classifying instrument families works better on average ($F = 0.82$) than classifying fine-grained classes ($F = 0.79$). For example, for woodwinds (WW), the family F-measure of 0.87 is much higher than the detection results obtained for the individual woodwind instruments (e. g., for clarinets: $F_{\mathtt{CL}} = 0.72$). Some very common classes yield high F-measures, but low specificity (e.g., for strings: $\delta_{\mathtt{ST}} = 0.87$, $F_{\mathtt{ST}} = 0.95$, and $S_{\mathtt{ST}} = 0.65$), indicating that our system produces many false positive predictions for these classes. In Section VIII, we will conduct some additional analyses to better understand these detection results with regard to possible confounding effects.

For singing activity (VOC), we obtain a family F-measure of 0.90 and the difference to the results for individual vocal classes female (Fe) and male (Ma, for both: $F = 0.88$) is small. For reference, one obtains accuracies of around 0.91 for singing voice detection on popular music [61], [63]. With regards to consistency, $\gamma^{\uparrow}$ is always high. Therefore, predictions on a fine-grained class are almost always accompanied by a prediction for the parent class. However, $\gamma^{\downarrow}$ is lower for some classes such as WW ($\gamma^{\downarrow}_{\mathtt{WW}} = 0.92$). Thus, for about eight percent of frames where the woodwind family is predicted as active, neither of the woodwind instruments is identified. In Section VII, we will consider loss terms for improving these consistency issues.

To demonstrate the impact of our hierarchical classification approach **HC**, we now analyze whether utilizing the instrument hierarchy during training can reduce the amount of fine-grained labels required. To this end, we compare **HC** with a flat classification baseline **FC**. There, our model is trained to only produce predictions for classes in $\mathbf{C}^1$ (i. e., instrument-level). At test time, we obtain predictions for classes in $\mathbf{C} \setminus \mathbf{C}^1$ by aggregating predictions from lower levels in a bottom-up fashion.[7] For example, VOC is predicted if and only if the model has classified the input as Fe or Ma. By construction, the predictions obtained in this way are always consistent, so $\gamma_c = \gamma_c^{\uparrow} = \gamma_c^{\downarrow} = 1$ for all classes c. The **FC** baseline allows us to determine the detection quality that can be achieved without informing the model about the class hierarchy.

We now reduce the amount of fine-grained instrument labels that are available during training. For **FC** (flat classification baseline), we do so by reducing the size of the training set, since this approach does not utilize class labels at higher levels for training. For **HC** (hierarchical classification), we disable the cross-entropy loss associated with classes in $\mathbf{C}^1$ on a portion of the training items, but we still use the instrument family and activity labels for these items.[8] This experiment setup is illustrated in Fig. 4.

Fig. 5 shows the results of this experiment. The upper plot shows results for higher-level classes (families and instrument

[6]These are macro averages, i. e., computed as the arithmetic mean of the results for the individual classes. As such, both common and uncommon classes contribute equally to these averages.

[7]This baseline is referred to as Strategy B in [2]. Note that we cannot obtain predictions for timpani (TMP) using this baseline and thus omit TMP from consideration in the following discussion.

[8]Disabling the cross-entropy loss for learning from partial labels was suggested in [66]. Gururani and Lerch [19] used this technique in the context of polyphonic instrument classification. They did not consider hierarchical information.

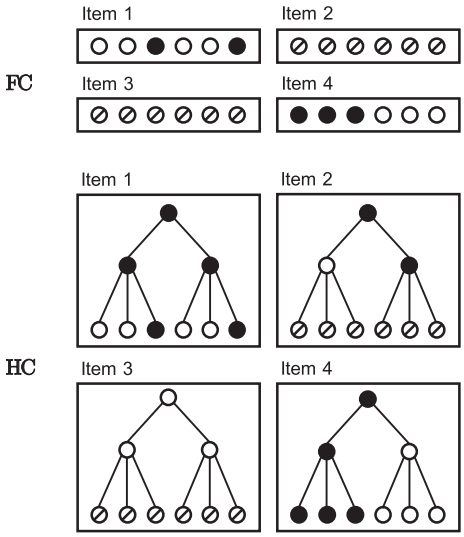

Fig. 4. Experiment setup for reducing the amount of instrument-level labels available for training. Boxes correspond to training items and circles indicate class annotations. Crossed-out circles correspond to missing annotations. For the **FC** (flat classification) approach, we can only use training items for which fine-grained labels are given. For **HC** (hierarchical classification), we can utilize higher-level information even if instrument labels are unavailable.

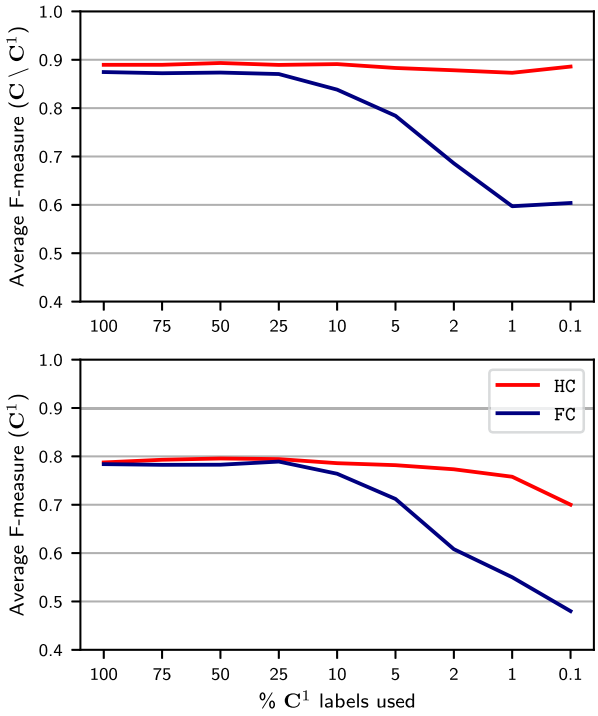

Fig. 5. Results for reducing the amount of instrument-level (i.e., $\mathbf{C}^1$) labels available for training. Average F-measures are plotted separately for instrument classes (lower plot) and higher-level classes ($\mathbf{C} \setminus \mathbf{C}^1$, upper plot). Lines correspond to hierarchical classification (**HC**) and flat classification (**FC**), respectively.

activity), whilst the lower plot contains results for fine-grained classes. When utilizing all fine-grained labels of the training dataset, we observe almost identical average F-measures for $\mathbf{C}^1$ with both **FC** and **HC** (lower plot, leftmost point). For $\mathbf{C} \setminus \mathbf{C}^1$ (upper plot), **HC** yields slightly higher average F-measures at 0.89 (compared to $F = 0.87$ for **FC**). When reducing the amount of $\mathbf{C}^1$ labels used, **HC** outperforms **FC** on both fine-grained and higher-level classes. For example, at 10% of labels used, we obtain an average F-measure of 0.84 on $\mathbf{C} \setminus \mathbf{C}^1$ for **FC**, which drops to $F = 0.60$ for 0.1% of labels. Meanwhile, the results for **HC** on $\mathbf{C} \setminus \mathbf{C}^1$ stay roughly constant at around $F = 0.88$. **HC** also yields higher F-measures for $\mathbf{C}^1$, with $F = 0.76$ at 1% of labels used as opposed to $F = 0.55$ for **FC**.

We have seen that, by utilizing higher-level structure when fine-grained labels are scarce, our hierarchical classification approach **HC** can still yield good results, even for small amounts of instrument-level labels. This opens up the possibility of incorporating partially labeled data for training, where the instrument family is known, but fine-grained labels are unavailable (e.g., monotimbral recordings of brass or string ensembles).

VII. Consistency Losses

As discussed in Section VI, our hierarchical IAD approach **HC** outperforms the flat classification baseline **FC** in terms of F-measures. However, the predictions of **FC** are always consistent, making the output of that system easier to understand compared to **HC**, which may produce inconsistent outputs. In this section, we will investigate additional loss terms for **HC** that can address this shortcoming.

In our previous work [2], which this article extends, we describe two loss terms for improving bottom-up and top-down consistency of predictions, respectively. In the following, we write p_c for the probability predicted by a model for class c on a given training item. In order to improve bottom-up consistency, we minimize the following loss term, which is adapted from [32]:

$$\mathcal{L}_{\uparrow} = \frac{1}{|\mathbf{C} \setminus \mathbf{C}^H|} \sum_{h=2}^{H} \sum_{c \in \mathbf{C}^h} \sum_{c' \in c\downarrow} \max\{0, p_{c'} - p_c\}^2 \qquad (7)$$

This sum contains terms for every pair (c, c') of a parent class c and a child class c', where the model incurs a loss whenever $p_{c'} > p_c$. Thus, the model is penalized for any bottom-up inconsistency. Similarly, to improve top-down consistency, we introduced the following loss term in [2]:

$$\mathcal{L}_{\downarrow} = \frac{1}{|\mathbf{C} \setminus \mathbf{C}^1|} \sum_{h=2}^{H} \sum_{c \in \mathbf{C}^h} \max\left\{0, p_c - \max_{c' \in c\downarrow} p_{c'}\right\}^2, \qquad (8)$$

Here, the model incurs a penalty whenever the output p_c for a parent class c is larger than $p_{c'}$ for any of the child classes c' (e.g., in case of a top-down inconsistent output). These losses are combined with the standard cross entropy loss $\mathcal{L}_{\text{BCE}}$ using weights $\alpha, \beta \in \mathbb{R}$, yielding the final loss

$$\mathcal{L} = \mathcal{L}_{\text{BCE}} + \alpha\mathcal{L}_{\downarrow} + \beta\mathcal{L}_{\uparrow} \qquad (9)$$

for our model. We will denote the hierarchical classification approach trained with these additional losses as $\texttt{HC}^{\alpha,\beta}$

Results for training with these additional consistency losses are shown in Table IV. Here, we set $\alpha = \beta = 10$ (as in [2]). With regard to the detection evaluation measures like F-measure and

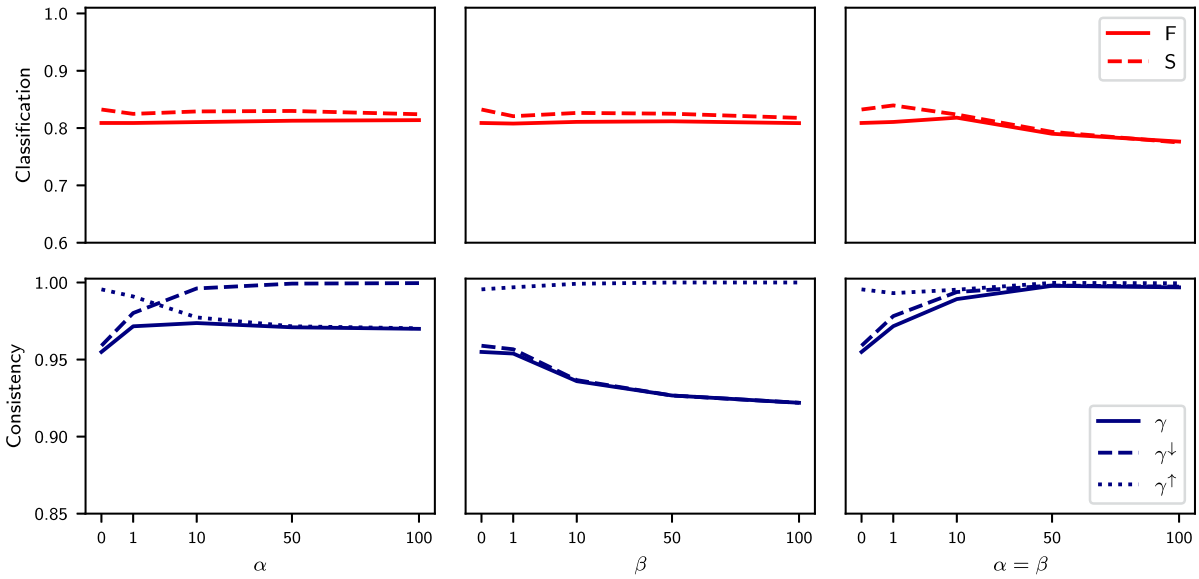

Fig. 6. Results for different choices of α (coefficient for the top-down loss $\mathcal{L}_{\downarrow}$) and β (coefficient for the bottom-up loss $\mathcal{L}_{\uparrow}$). The upper row shows classification results in terms of F-measure (F) and specificity (S). The lower row shows measures of top-down ($\gamma^{\downarrow}$), bottom-up ($\gamma^{\uparrow}$) and overall consistency (γ). In the first column $\beta = 0$, and in the second column $\alpha = 0$. Measures are averaged over all classes.

TABLE IV
RESULTS FOR THE $\texttt{HC}^{\alpha,\beta}$ STRATEGY THAT INCLUDES HIERARCHY INFORMATION AND CONSISTENCY LOSSES DURING TRAINING. HERE, WE SET $\alpha = \beta = 10$

	P	R	F	S	$\gamma^{\downarrow}$	$\gamma^{\uparrow}$	δ
INST	0.99	0.99	0.99	0.73	1.00	1.00	0.96
WW	0.88	0.85	0.87	0.83	0.98	0.99	0.59
Fl	0.76	0.65	0.70	0.89			0.35
Ob	0.76	0.74	0.75	0.86			0.38
Cl	0.73	0.77	0.75	0.79			0.42
Bn	0.73	0.82	0.77	0.78			0.42
BR	0.79	0.70	0.74	0.85	1.00	1.00	0.45
Hn	0.73	0.70	0.71	0.82			0.41
Tpt	0.80	0.54	0.65	0.97			0.16
TMP	0.78	0.59	0.67	0.98			0.11
VOC	0.93	0.87	0.90	0.99	1.00	0.99	0.12
Fe	0.93	0.83	0.88	1.00			0.06
Ma	0.90	0.88	0.89	0.99			0.06
ST	0.94	0.96	0.95	0.60	1.00	1.00	0.87
Vn	0.88	0.95	0.91	0.58			0.77
Va	0.81	0.88	0.84	0.63			0.64
Vc	0.85	0.91	0.88	0.71			0.64
Db	0.86	0.87	0.87	0.82			0.56
Avg. ($\mathbf{C}$)	0.84	0.81	0.82	0.82	0.99	1.00	
Avg. ($\mathbf{C}^2$)	0.87	0.79	0.83	0.85			
Avg. ($\mathbf{C}^1$)	0.81	0.80	0.80	0.82			

specificity, we obtain similar results as in our previous experiments that did not employ consistency losses (see in Table III). For example, we get an average $F = 0.82$ and $S = 0.82$ for $\texttt{HC}^{\alpha,\beta}$ compared to $F = 0.81$ and $S = 0.83$ for HC. However, the top-down consistency scores $\gamma^{\downarrow}$ have improved (e.g., $\gamma^{\downarrow}_{\texttt{BR}} = 1.00$ and $\gamma^{\downarrow}_{\texttt{WW}} = 0.98$ for $\texttt{HC}^{\alpha,\beta}$ compared to $\gamma^{\downarrow}_{\texttt{BR}} = 0.94$ and $\gamma^{\downarrow}_{\texttt{WW}} = 0.92$ for HC). Thus, the additional loss terms can improve consistency while retaining the overall quality of results.

A more detailed analysis of the impact of the loss weights α and β is provided in Fig. 6. Here, we either use solely $\mathcal{L}_{\downarrow}$ (by setting $\beta = 0$ and increasing α, first column), use solely $\mathcal{L}_{\uparrow}$ (setting $\alpha = 0$ and increasing β, second column), or use both losses simultaneously ($\alpha = \beta$, third column). The upper row shows classification results in terms of average F-measure (F) and specificity (S), while the lower row shows average consistency scores. As expected, by using solely $\mathcal{L}_{\downarrow}$, we are able to improve $\gamma^{\downarrow}$. However, $\gamma^{\uparrow}$ decreases for large values of α. The opposite behavior can be observed for using only $\mathcal{L}_{\uparrow}$. In both cases, γ decreases while F-measure and specificity remain roughly constant. By utilizing both $\mathcal{L}_{\downarrow}$ and $\mathcal{L}_{\uparrow}$, we are able to improve γ. However, F and S are reduced for large values of $\alpha = \beta$. We conclude that both $\mathcal{L}_{\downarrow}$ and $\mathcal{L}_{\uparrow}$ are required to increase consistency, while $\alpha = \beta$ should be chosen small enough in order not to deteriorate detection results.

Overall, our results suggest that, while consistency is a necessary condition for interpretable system outputs, it is not sufficient to achieve good classification results. Our losses can be used to induce more consistent detection outputs for our model, but high consistency needs to be balanced with preserving classification results.

VIII. ANALYSIS OF CONFOUNDING FACTORS

In this section, we aim at a deeper understanding of the behavior of our model. In particular, we analyze how the detection of an instrument class is affected by the presence of other instruments playing simultaneously. To this end, we systematically evaluate our model on audio inputs for which we can calculate the relative salience of different classes within the overall mixture (concretely, we use the multi-track orchestral datasets discussed

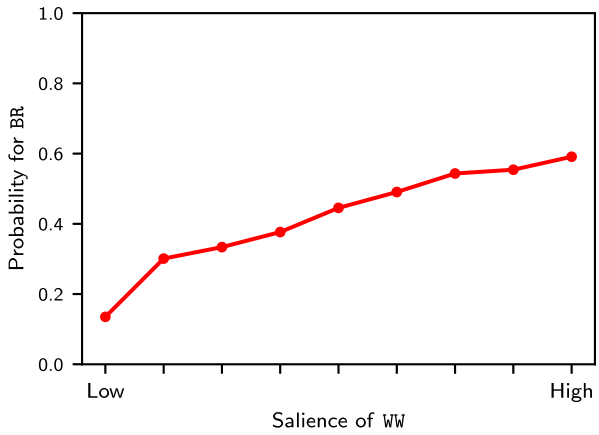

Fig. 7. Average probabilities output by our model for brass depending on the salience of woodwinds within the orchestral mixture. Here we only consider frames where no brass is active at all, i. e., the ideal probability output would be 0.

in Section IV). In this way, we can reach conclusions about confounding effects exploited by our model.

A representative example of an analysis result is shown in Fig. 7. In this example, we are interested in the interactions between brass (BR) and woodwind (WW) instruments. In orchestra music, brass and woodwinds are often active simultaneously. For this analysis, we select frames for which no brass (BR) is active and then observe the predictions for BR depending on the salience of woodwind (WW) instruments. The vertical axis shows average probabilities predicted by our model for BR. The probabilities are averaged over all frames where woodwinds have a certain salience within the orchestral mixture (horizontal axis). Here, salience is measured as a signal-to-noise ratio, with woodwinds considered as signal and all other instrument sounds considered as noise. We observe that the model outputs low probabilities of around 0.1 for BR if WW is inactive. However, these outputs gradually increase as woodwinds become more salient in the input. We conclude that brass detection is highly sensitive to woodwind instruments, even if no brass instrument is active in the mixture. This is a confounding effect.

We can reach a number of similar conclusions by performing systematic analyses:

1) The model also exploits the presence of woodwind instruments for detecting brass if brass instruments are present in the mixture.
2) Our model is biased towards predicting string activity, even when no strings are active in the input. Thus, the model exploits the fact that strings are active in most frames of the recordings in our dataset (see also Fig. 3).
3) Predictions for strings are not affected much by the presence of other instrument families.
4) Looking at the model behavior for woodwind instrument classes, we find that detection of flutes is improved by the simultaneous activity of other woodwind instruments, similar to the behavior observed for BR.

It is important to note that it may indeed be desirable for our model to use these confounding factors for detection, since these are strong cues for IAD on many orchestral works from the classical and romantic periods (as present in our dataset). Yet, use of these confounding factors may also limit the generalization ability of our model to music with other instrument statistics, e. g., works with many brass-only sections. To reduce the impact of these effects, one may collect additional training data (which is cumbersome, see Section IV). Entirely removing confounding effects from our system may be impossible, however, as we also discuss in the next section.

IX. Conclusion

In this article, we investigated instrument activity detection in the context of complex orchestral music recordings. We showed that utilizing information about hierarchical relationships between instruments is helpful, especially when only few fine-grained instrument-level labels are available. Furthermore, we demonstrated how one can increase the consistency of predictions across hierarchy levels using additional consistency losses, while preserving detection quality. To perform these experiments, we collected a large dataset of real-world opera and orchestra recordings with aligned instrument activity annotations. Finally, we analyzed the behavior of our detection system and identified confounding effects exploited by our model.

Future work may make use of more complex instrument hierarchies (e.g., hierarchies that incorporate knowledge about different sound production techniques), train more complex detection models, collect additional data for training and testing (e.g., using music synchronization or synthesizers), or extend the scenario considered here towards orchestral music from other epochs (e.g., Baroque).

However, even when collecting additional data, it is likely that our model will continue to exploit confounding effects arising from the training data, as shown in our analysis in Section VIII. This may be desirable (in case that training and test conditions are very similar) or undesirable (when generalizing to music from different styles). Our analysis results mirror those obtained for other systems for music classification. For example, Kelz and Widmer [67] found that a neural network for piano transcription trained on combinations of notes struggles with correctly classifying unknown note combinations. The system performs transcription for certain notes by considering other, unrelated notes as well. The confounding effects exploited in MIR systems can often be even less intuitive. In [68], for example, a system for recognizing Latin music styles (which are usually characterized by their rhythms) was shown to exploit tempo information. Similarly, the authors in [69] found that a system for genre recognition is sensitive to sounds outside the human hearing range. In [70], a deep learning system for leitmotif detection in opera recordings is introduced. It is shown to rely heavily on spectral statistics as opposed to melody or rhythm. Such problems are amplified when evaluating a system on music styles not seen during training. For example, the authors in [71] showed that off-the-shelf music classifiers perform poorly on a large and diverse music database and are highly sensitive to encoding artifacts.

In order to tackle this challenge for IAD, one may consider using source separation as pre-processing, thereby reducing the

impact of unrelated instruments on detection outputs. However, due to the small amount of multi-track orchestral data available, no off-the-shelf systems for source separation in orchestra recordings are currently available, leaving this as an avenue for future work.

ACKNOWLEDGMENT

We thank Christof Weiß for helpful discussions. The authors are with the International Audio Laboratories Erlangen, a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS. The authors gratefully acknowledge the compute resources and support provided by the Erlangen Regional Computing Center (RRZE).

REFERENCES

[1] B. L. Sturm, "A simple method to determine if a music information retrieval system is a "horse"," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1636–1644, Oct. 2014.

[2] M. Krause and M. Müller, "Hierarchical classification for singing activity, gender, and type in complex music recordings," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2022, pp. 406–410.

[3] A. J. Eronen and A. P. Klapuri, "Musical instrument recognition using cepstral coefficients and temporal features," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Istanbul, Turkey, 2000, pp. II753–II756.

[4] M. Grasis, J. Abeßer, C. Dittmar, and H. M. Lukashevich, "A multiple-expert framework for instrument recognition," in *Proc. Int. Symp. Sound Music Motion*, Marseille, France, 2014, pp. 619–634.

[5] S. K. Tjoa and K. J. R. Liu, "Musical instrument recognition using biologically inspired filtering of temporal dictionary atoms," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2010, pp. 435–440.

[6] C. Joder, S. Essid, and G. Richard, "Temporal integration for audio classification with application to musical instrument classification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 17, no. 1, pp. 174–186, Jan. 2009.

[7] S. Essid, G. Richard, and B. David, "Hierarchical classification of musical instruments on solo recordings," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Toulouse, France, 2006, pp. 817–820.

[8] K. Patil and M. Elhilali, "Biomimetic spectro-temporal features for music instrument recognition in isolated notes and solo phrases," *EURASIP J. Audio Speech Music Process.*, vol. 2015, 2015,Art. no. 27.

[9] P. Hamel, S. Wood, and D. Eck, "Automatic identification of instrument classes in polyphonic and poly-instrument audio," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Kobe, Japan, 2009, pp. 399–404.

[10] J. Eggink and G. J. Brown, "A missing feature approach to instrument identification in polyphonic music," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Hong Kong, China, 2003, pp. 553–556.

[11] T. Kitahara, "Computational musical instrument recognition and its application to content-based music information retrieval," Ph.D. dissertation, Kyoto Univ., Kyoto, Japan, 2007.

[12] D. Little and B. Pardo, "Learning musical instruments from mixtures of audio with weak labels," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2008, pp. 127–132.

[13] F. Fuhrmann and P. Herrera, "Polyphonic instrument recognition for exploring semantic similarities in music," in *Proc. 13th Int. Conf. Digit. Audio Effects*, Graz, Austria, 2010, pp. 1–8.

[14] T. Heittola, A. P. Klapuri, and T. Virtanen, "Musical instrument recognition in polyphonic audio using source-filter model for sound separation," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Kobe, Japan, 2009, pp. 327–332.

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

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

[17] Y.-N. Hung, Y.-A. Chen, and Y.-H. Yang, "Multitask learning for frame-level instrument recognition," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* Brighton, U.K., 2019, pp. 381–385.

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

[19] S. Gururani and A. Lerch, "Semi-supervised audio classification with partially labeled data," in *IEEE Int. Symp. Multimedia*, Naple, Italy, 2021, pp. 111–114.

[20] A. Kratimenos, K. Avramidis, C. Garoufis, A. Zlatintsi, and P. Maragos, "Augmentation methods on monophonic audio for instrument classification in polyphonic music," in *Proc. IEEE Eur. Signal Process. Conf.*, Amsterdam, Netherlands, 2020, pp. 156–160.

[21] E. J. Humphrey, S. Durand, and B. McFee, "OpenMIC-2018: An open data-set for multiple instrument recognition," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Paris, France, 2018, pp. 438–444.

[22] J. Cramer, V. Lostanlen, A. Farnsworth, J. Salamon, and J. P. Bello, "Chirping up the right tree: Incorporating biological taxonomies into deep bioacoustic classifiers," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Barcelona, Spain, 2020, pp. 901–905.

[23] Y. Xu, Q. Huang, W. Wang, and M. D. Plumbley, "Hierachical learning for DNN-based acoustic scene classification," in *Proc. Workshop Detection Classification Acoust. Scenes Events*, Budapest, Hungary, 2016, pp. 110–114.

[24] A. Jati, N. Kumar, R. Chen, and P. G. Georgiou, "Hierarchy-aware loss function on a tree structured label space for audio event detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Brighton, U. K., 2019, pp. 6–10.

[25] A. Zharmagambetov et al., "Improved representation learning for acoustic event classification using tree-structured ontology," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Singapore, 2022, pp. 321–325.

[26] F. Zih-Sing and L. Su, "Hierarchical classification networks for singing voice segmentation and transcription," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Delft, The Netherlands, 2019, pp. 900–907.

[27] E. Manilow, G. Wichern, and J. L. Roux, "Hierarchical musical instrument separation," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Montreal, Canada, 2020, pp. 376–383.

[28] H. F. Garcia, A. Aguilar, E. Manilow, and B. Pardo, "Leveraging hierarchical structures for few-shot musical instrument recognition," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2021, pp. 220–228.

[29] Y. Wang, J. Salamon, N. J. Bryan, and J. P. Bello, "Few-shot sound event detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* Barcelona, Spain, 2020, pp. 81–85.

[30] I. Nolasco and D. Stowell, "Rank-based loss for learning hierarchical representations," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Singapore, 2022, pp. 3623–3627.

[31] C. N. Silla Jr and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining Knowl. Discov.*, vol. 22, no. 1/2, pp. 31–72, 2011.

[32] J. Wehrmann, R. Cerri, and R. C. Barros, "Hierarchical multi-label classification networks," in *Proc. Int. Conf. Mach. Learn.*, Stockholm, Sweden, 2018, pp. 5225–5234.

[33] L. Bertinetto, R. Müller, K. Tertikas, S. Samangooei, and N. A. Lord, "Making better mistakes: Leveraging class hierarchies with deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2020, pp. 12503–12512.

[34] R. Cerri, R. C. Barros, and A. C. P. de Leon Ferreira de Carvalho, "Hierarchical multi-label classification using local neural networks," *J. Comput. Syst. Sci.*, vol. 80, no. 1, pp. 39–56, 2014.

[35] E. Giunchiglia and T. Lukasiewicz, "Coherent hierarchical multi-label classification networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 9662–9673.

[36] M. Krause, M. Müller, and C. Weiß, "Singing voice detection in opera recordings: A case study on robustness and generalization," *Electronics*, vol. 10, no. 10, pp. 1214:1–14, 2021.

[37] C. Dittmar, B. Lehner, T. Prätzlich, M. Müller, and G. Widmer, "Cross-version singing voice detection in classical opera recordings," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Málaga, Spain, 2015, pp. 618–624.

[38] E. Parada-Cabaleiro et al., "Identifying emotions in opera singing: Implications of adverse acoustic conditions," in *Proc. Int. Conf. Digit. Libraries Musicol.*, Paris, France, 2018, pp. 376–382.

[39] Z. Tang and D. A. A. Black, "Melody extraction from polyphonic audio of Western opera: A method based on detection of the singer's formant," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Taipei, Taiwan, 2014, pp. 161–166.

[40] M. Miron, J. J. Carabias-Orti, J. J. Bosch, E. Gómez, and J. Janer, "Score-informed source separation for multichannel orchestral recordings," *J. Elect. Comput. Eng.*, vol. 2016, 2016, Art. no. 8363507.

[41] Y. Han and C. Raphael, "Informed source separation of orchestra and soloist," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Utrecht, The Netherlands, 2010, pp. 315–320.
[42] M. Taenzer, J. Abeßer, S. I. Mimilakis, C. Weiß, H. Lukashevich, and M. Müller, "Investigating CNN-based instrument family recognition for Western classical music recordings," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Delft, The Netherlands, 2019, pp. 612–619.
[43] S. Kolozali, M. Barthet, G. Fazekas, and M. B. Sandler, "Knowledge representation issues in musical instrument ontology design," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Miami, Florida, USA, 2011, pp. 465–470.
[44] T. Magnusson, "Musical organics: A heterarchical approach to digital organology," *J. New Music Res.*, vol. 46, no. 3, pp. 286–303, 2017.
[45] T. Prätzlich, M. Müller, B. W. Bohl, and J. Veit, "Freischütz Digital: Demos of audio-related contributions," in *Proc. Demos Late Breaking News Int. Soc. Music Inf. Retrieval Conf.*, Malága, Spain, 2015.
[46] C. Böhm, D. Ackermann, and S. Weinzierl, "A multi-channel anechoic orchestra recording of Beethoven's Symphony no. 8 op. 93," *J. Audio Eng. Soc.*, vol. 68, no. 12, pp. 977–984, 2021.
[47] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "MedleyDB: A multitrack dataset for annotation-intensive MIR research," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Taipei, Taiwan, 2014, pp. 155–160.
[48] S. Sarkar, E. Benetos, and M. B. Sandler, "EnsembleSet: A new high quality dataset for chamber ensemble separation," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Bengaluru, India, 2022, pp. 625–632.
[49] C. Liang, L. Su, Y. Yang, and H. Lin, "Musical offset detection of pitched instruments: The case of violin," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Málaga, Spain, 2015, pp. 281–287.
[50] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning features of music from scratch," in *Proc. Int. Conf. Learn. Representations*, Toulon, France, 2017.
[51] T. Prätzlich, J. Driedger, and M. Müller, "Memory-restricted multiscale dynamic time warping," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.* Shanghai, China, 2016, pp. 569–573.
[52] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync Toolbox: A Python package for efficient, robust, and accurate music synchronization," *J. Open Source Softw.*, vol. 6, no. 64, pp. 3434:1–4, 2021.
[53] C. Weiß, V. Arifi-Müller, T. Prätzlich, R. Kleinertz, and M. Müller, "Analyzing measure annotations for Western classical music recordings," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, New York, USA, 2016, pp. 517–523.
[54] C. Weiß, H. Schreiber, and M. Müller, "Local key estimation in music recordings: A case study across songs, versions, and annotators," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2919–2932, 2020.
[55] A. Flexer, "A closer look on artist filters for musical genre classification," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Vienna, Austria, 2007, pp. 341–344.
[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, Nevada, USA, 2016, pp. 770–778.
[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, San Diego, California, USA, 2015.
[58] R. M. Bittner, B. McFee, J. Salamon, P. Li, and J. P. Bello, "Deep salience representations for F0 tracking in polyphonic music," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Suzhou, China, 2017, pp. 63–70.
[59] C. Weiß, J. Zeitler, T. Zunner, F. Schuberth, and M. Müller, "Learning pitch-class representations from score–audio pairs of classical music," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2021, pp. 746–753.
[60] J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *Proc. Int. Workshop Content-Based Multimedia Indexing*, Bucharest, Romania, 2016, pp. 1–6.
[61] J. Schlüter and B. Lehner, "Zero-mean convolutions for level-invariant singing voice detection," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Paris, France, 2018, pp. 321–326.
[62] J. Abeßer and M. Müller, "Jazz bass transcription using a U-net architecture," *Electronics*, vol. 10, no. 6, 2021, Art. no. 670.
[63] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Málaga, Spain, 2015, pp. 121–126.
[64] D. S. Park et al., "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Graz, Austria, 2019, pp. 2613–2617.
[65] B. Lehner, G. Widmer, and R. Sonnleitner, "On the reduction of false positives in singing voice detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Florence, Italy, 2014, pp. 7480–7484.
[66] T. Durand, N. Mehrasa, and G. Mori, "Learning a deep convnet for multi-label classification with partial labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 647–657.
[67] R. Kelz and G. Widmer, "An experimental analysis of the entanglement problem in neural-network-based music transcription systems," in *Proc. AES Int. Conf. Semantic Audio*, Erlangen, Germany, 2017, pp. 194–201.
[68] B. L. Sturm, C. Kereliuk, and J. Larsen, "¿El Caballo Viejo? Latin genre recognition with deep learning and spectral periodicity," in *Proc. Int. Conf. Math. Comput. Music*, London, U.K., 2015, pp. 335–346.
[69] F. Rodríguez-Algarra, B. L. Sturm, and H. Maruri-Aguilar, "Analysing scattering-based music content analysis systems: Wheres the music?," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, New York City, NY, USA, 2016, pp. 344–350.
[70] M. Krause, M. Müller, and C. Weiß, "Towards leitmotif activity detection in opera recordings," *Trans. Int. Soc. Music Inf. Retrieval*, vol. 4, no. 1, pp. 127–140, 2021.
[71] C. C. S. Liem and C. Mostert, "Can't trust the feeling? How open data reveals unexpected behavior of high-level music descriptors," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, Montreal, Canada, 2020, pp. 240–247.

Michael Krause received the B.Sc. and M.Sc. degrees in computer science from RWTH Aachen University, Aachen, Germany, in 2017 and 2019 respectively. During the studies, he was supported by a scholarship of the German National Academic Foundation. He is currently working toward the Ph.D. degree on music information retrieval with International Audio Laboratories Erlangen under the supervision of Prof. Meinard Müller. His research interests include musical sound event detection, differentiable digital signal processing, and soft sequence alignment.

Meinard Müller (Fellow, IEEE) received the Diploma in mathematics and Ph.D. degree in computer science from the University of Bonn, Bonn, Germany, in 1997 and 2001, rspectively. After the Postdoctoral studies (2001-2003) from Japan and Habilitation (2003-2007) in multimedia retrieval from Bonn, he was a Senior Researcher with Saarland University, Saarbrcken, Germany, and the Max-Planck Institut für Informatik (2007-2012), Saarbrcken. Since 2012, he has been holding a Professorship in semantic audio signal processing with the International Audio Laboratories Erlangen, Erlangen, Germany, a joint institute of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and the Fraunhofer Institute for Integrated Circuits IIS. His research interests include music processing, music information retrieval, audio signal processing, and motion processing. He was a Member of the IEEE Audio and Acoustic Signal Processing Technical Committee (2010-2015), Senior Editorial Board of the IEEE Signal Processing Magazine (2018-2022), and the Board of Directors, International Society for Music Information Retrieval (2009-2021, being its President in 2020/2021). In 2020, he was elevated to IEEE Fellow for contributions to music signal processing.