# Hierarchical Regulated Iterative Network for Joint Task of Music Detection and Music Relative Loudness Estimation

Bijue Jia, Jiancheng Lv, Xi Peng, Yao Chen, and Shenglan Yang

***Abstract*—One practical requirement of the music copyright management is the estimation of music relative loudness, which is mostly ignored in existing music detection works. To solve this problem, we study the joint task of music detection and music relative loudness estimation. To be specific, we observe that the joint task has two characteristics, i.e., temporality and hierarchy, which could facilitate to obtain the solution. For example, a tiny fragment of audio is *temporally* related to its neighbor fragments because they may all belong to the same event, and the event classes of the fragment in the two tasks have a *hierarchical* relationship. Based on the above observation, we reformulate the joint task as hierarchical event detection and localization problem. To solve this problem, we further propose Hierarchical Regulated Iterative Networks (HRIN), which includes two variants, termed as HRIN-r and HRIN-cr, which are based on recurrent and convolutional recurrent modules. To enjoy the joint task's characteristics, our models employ an iterative framework to achieve encouraging capability in temporal modeling while designing three hierarchical violation penalties to regulate hierarchy. Extensive experiments on the currently largest dataset (i.e., OpenBMAT) show that the promising performance of our HRIN in the segment-level and event-level evaluations.**

***Index Terms*—music detection, music relative loudness estimation, event detection, event localization, neural networks, hierarchical classification**

## I. Introduction

MUSIC detection (MD) refers to the task of finding out whether a music event happens in an audio file and what time it starts and ends, i.e., splitting an audio recording and annotating each fragment as *music* or *non-music*. MD not only has the basic application in automatic retrieving and localizing audio data based on the type of content but also has a more practical application of monitoring music for copyright management. The practical application in the music industry is the royalty collection in broadcasting. As elaborated in [1]: the Austrian National Broadcasting Corporation (ORF) requires knowing where exactly the music appears in the soundtrack of TV production, and detecting the music is in the foreground or the background. ORF posed this requirement for the purpose of calculating the royalty fees, which are paid to a national agency according to certain rules. Ideally, the production team would provide a list of all the music segments occurring in TV production, but in reality, these lists are largely inaccurate. As a result, ORF has to guess the amount of music within a production more or less because manually annotating all productions is impossible. Also, the copyright fee will be different for music is used in the foreground or the background [2]. Hence, it is highly expected to develop method of music relative loudness estimation (MRLE), i.e., annotating each fragment as *fg-music*, *bg-music*, or *non-music*.

In the past, research mainly focused on the music/speech detection task, which is segmenting and annotating audio as *music*, *speech*, or *noise*. Early work [3] explored the distinguishable features between music and speech from the perspective of signal processing. Using these handcrafted features, later research [4-6] added subsequent classifiers to do music/speech detection. Recent works [7-9] focused on automatically-learned features from spectrogram images and used neural networks as classifiers. In contrast to simple music/speech detection task, the emphasis point of MD task is different: music is used to accentuate scenes, therefore speech and any noise signals might be present concurrently.

In recent years, researchers started to focus on the joint task of MD and MRLE. The Music Information Retrieval Evaluation eXchange (MIREX) competition also changed their goal of music detection task from music/speech detection to the joint task of MD and MRLE since 2018. Existing work [10] solved the joint task (i.e., MD and MRLE) by treating it as an image classification rather than sequence labeling. They cut the audio spectrogram into blocks of 128 frames and treated each block as an independent training sample. Thus, this approach ignores any sequential information.

More interestingly, we observe that the joint task of MD and MRLE has a unique characteristic that differs from the simple music/speech detection task: there exists a hierarchy in-between the labels of MD task and MRLE task. Since the MRLE task extends from the MD task to further classifying the music event into a foreground music event or background music event, categories of the two tasks naturally constitute a hierarchy of two levels, as depicted in Fig. 1. With this insight, the joint task is highly related to the hierarchical classification problem. To our best knowledge, no study has been constructed this way for the joint task of MD and MRLE.

In this paper, we reformulate the joint task of MD and MRLE as the event detection & localization problem. We

This work is supported in part by the National Key Research and Development Program of China under Contract 2017YFB1002201, in part by the National Natural Science Fund for Distinguished Young Scholar under Grant 61625204, and in part by the State Key Program of the National Science Foundation of China under Grant 61836006. *(Corresponding author: Jiancheng Lv)*

B. Jia, J. Lv, X. Peng, Y. Chen, and S. Yang are with the College of Computer Science, State Key Laboratory of Hydraulics and Mountain River Engineering, Sichuan University, Chengdu 610065, China (e-mail: jiabijue@outlook.com; lvjiancheng@scu.edu.cn; pengx.gm@gmail.com; chenyaoscu@outlook.com; yangshenglan@stu.scu.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3030484, IEEE/ACM Transactions on Audio, Speech, and Language Processing



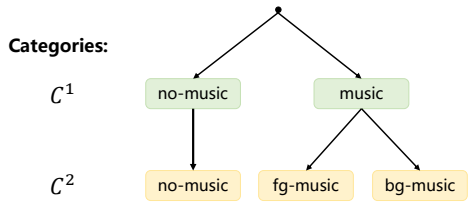



Fig. 1. Taxonomy of the event classes in the joint-task of MD and MRLE. $C^1$ is the set of event categories of the $1^{st}$ hierarchical level, including two classes of MD task: *non-music* and *music*. $C^2$ is the set of event categories of the $2^{nd}$ hierarchical level, consisting of three classes of MRLE task: *non-music*, *fg-music* (short for foreground-music), and *bg-music* (abbreviation of background-music). It is worth mentioning that in the common hierarchical setting, *non-music* should not appear in the $C^2$ level. As it will be used for constructing the model, we retain *non-music* in $C^2$ for clarity (more details are discussed in Section IV).

propose Hierarchical Regulated Iterative Network (HRIN) to detect event type in two hierarchical levels. HRIN uses an iterative structure to guarantee the temporal modeling. The proposed HRIN implements two variants: with- and without-convolutional unit. In both variants, it comprises two outputs, one output layer corresponding to one hierarchical level. For each output, there is a corresponding loss function to supervise the learning. In addition, we introduce three hierarchical violation penalties to regulate hierarchy. The contributions of this paper include the following:

- We reformulate the joint task of MD and MRLE as a Hierarchical Event Detection and Localization (HEDL) problem. We take into consideration that the joint task possesses two innate characteristics: temporality and hierarchy.
- We propose the HRIN model, including two architecture variants HRIN-r and HRIN-cr, to solve the HEDL problem that we define. Our model takes advantage of an iterative structure to model temporality. Moreover, the proposed HRIN consists of three penalties that regulate event predictions to obey the hierarchical structure.
- We show that HRIN comfortably establishes itself as a good model for the joint task of MD and MRLE, achieving better or at least competitive results over both the segment-level and event-level evaluations.

The rest of this paper is organized as follows. In Section II, we retrospect some related work in the area of music detection, music relative loudness estimation, multi-task learning, event detection and localization, and hierarchical classification. In Section III, we describe in detail the motivations behind our model and give a formal definition of the joint task of MD and MRLE. Details of network architectures and loss function of our proposed model are described in Section IV. In Section V, dataset, experiment details, evaluation metrics, and performance analysis are given. Finally, we summarize the paper and our future work in Section VI.

## II. Related Work

**Music Detection** Many works have been proposed for the single task of music detection. Seyerlehner *et al.* [1] proposed the manually-designed feature called Continuous Frequency Activation (CFA) for music/non-music detection. Benito-Gorron *et al.* [8] explored different neural networks and trained them to solve speech detection and music detection separately and simultaneously. For speech detection only, they classified audio fragment into two classes of *no-speech* and *speech*; for music detection only, they categorized audio fragment into *non-music* and *music*; for simultaneous speech and music detection, they classified audio fragment into four classes which are *no-speech*, *speech*, *non-music*, and *music*. Lemaire *et al.* [9] focused on the situation where speech and music may overlap. They proposed a Temporal Convolutional Network (TCN) to solve the music/speech detection under the overlapping situation. In contrast to these methods, our method focuses on joint-task learning and uses the relationship between tasks to promote each task's performance mutually.

**Music Relative Loudness Estimation** Music relative loudness estimation is a sub-task derived from the traditional music detection task. This task is normally combined with the music detection task as a joint task. Meléndez-Catalán *et al.* proposed a CNN based method called MMG [10] (named by the initials of authors' last name) for the joint task in the 2018 MIREX competition. Meléndez-Catalán *et al.* also used MMG as the benchmark model to test the dataset they proposed called OpenBMAT. In the MMG model, they segmented the spectrogram of the minute-long audio into frames, treated each frame as a single 2-dimensional sample, and classified each sample into one music event category. They abandoned the sequential relation between frames and dealt with it in an image classification way. Compared to theirs, our method keeps the sequential relations between frames and models the frame sequence.

**Multi-Task Learning** The general multi-task learning problem [11] has been studied for a long time, and many works have been done in different research areas such as music information retrieval [12-16], computer vision [17-23], natural language processing [24-30] and so on. Here the most related research is music information retrieval. Böck *et al.* [13] proposed to use recurrent neural network for predicting probabilities of beats/downbeats and use dynamic Bayesian network to align the predicted beat and downbeat positions to the global best solution. Vogl [14] proposed a system to detect drum instrument onsets along with the corresponding beats and downbeats using different neural networks, taking into consideration the additional meta-information like bar boundaries, tempo, and meter. Bittner *et al.* [15] presented a multi-task deep learning architecture that jointly estimates outputs for various tasks, including multiple f0, melody, vocal, and bassline estimation. Böck *et al.* [16] proposed a multi-task learning approach by globally aggregating the skip connections of a beat tracking system built around temporal convolutional networks and feeding them into a tempo classification layer for simultaneous tempo estimation and beat tracking of musical audio. Even though the above works and our work all resort to a joint task learning scheme to improve one task by learning from another task(s), our work presents to construct a joint task hierarchically to further refines and boosts the performance of each task.

**Event Detection and Localization** In addition to the MD

task and MRLE task, many other tasks belong to event detection and localization problem in the music information retrieval area [31-33]. Durand *et al.* [34] used an ensemble of convolutional networks plus an HMM for downbeat detection and localization. Schroeter *et al.* [33] introduced Occurrence Count Learning for the weakly-supervised temporal detection and localization problem with experiments on drum transcription task and piano onset detection task. The majority of event detection and localization problems have been researched for the broader audio data (including speech, music, and other sounds). Adavanne *et al.* [35] proposed a convolutional recurrent neural network for sound event detection and localization of multiple overlapping sound events in 3D space. Kong *et al.* [36] proposed a time-frequency segmentation framework trained on weakly labeled data to tackle the sound event detection and localization problem. Nguyen *et al.* [37] proposed a two-step system to do sound event localization and detection; the first step is to detect the sound events and estimate the directions-of-arrival separately; the second step is to combine the results of the event detector and direction-of-arrival estimator. The above research differs from our work in the aspect that our research focuses on the event detection and localization problem whose events are organized hierarchically.

**Hierarchical Classification** Various research has been conducted in the area of hierarchical text classification. Mao *et al.* [38] proposed an end-to-end reinforcement learning approach to hierarchical text classification where objects are labeled by placing them at the proper positions in the label hierarchy. Aly *et al.* [39] proposed to use simple shallow capsule networks for hierarchical multi-label text classification and demonstrated that capsule networks are especially advantageous for rare events and structurally diverse categories. For hierarchical document classification, Huang *et al.* [40] proposed the Hierarchical Attention-based Recurrent Neural Network (HARNN) for classifying documents into the most relevant categories level by level via integrating texts and the hierarchical category structure. In the area of hierarchical image classification, there are also some studies. Zhu *et al.* [41] introduced a variant of the traditional CNN model, named the Branch Convolutional Neural Network (B-CNN), to output multiple predictions ordered from coarse to fine, along the concatenated convolutional layers corresponding to the hierarchical structure of the target classes. Wang *et al.* [42] proposed a deep fuzzy tree model that could learn a better tree structure and classifiers for hierarchical classification with a fuzzy rough set theory guarantee. For the general hierarchical classification, Wehrmann *et al.* [43] proposed the Hierarchical Multi-Label Classification Networks (HMCN), which simultaneously optimizes local and global loss functions for discovering local hierarchical class-relationships and global information from the entire class hierarchy while penalizing hierarchical violations. Unlike these works, our method focuses on the event detection and localization problem–especially the joint task of MD and MRLE–to perform hierarchical classification per time step.

In the above works, none of them has applied the hierarchical classification to the joint task of music detection and music relative loudness estimation. In this work, we reformulate this joint task as Hierarchical Event Detection and Localization (HEDL) problem. Based on this, we propose the Hierarchical Regulated Iterative Network (HRIN) to solve this problem. Specifically, we design HRIN to capture relationships between event classes and to regulate the predictions to obey the hierarchical structure.

## III. Motivation and Problem Formulation

In this section, we begin by elaborating on the motivation of our work. Based on this motivation, the joint task of MD and MRLE constructs in a way that is different from previous studies; thus, we also give the formal definition of the joint task mathematically.

### A. Motivation

In essence, both the MD and MRLE tasks are event detection and localization problem, and a typical way to solve this problem consists of three phases: first, slicing an audio recording into fragments along the time axis; second, detecting which event type each fragment belongs to; third, calculating the onset (start time) and offset (end time) of each event based on the location of the corresponding fragment in the original audio. Based on the three phases, our work is then motivated by two aspects: 1) constructing per-timestep-detection as a hierarchical classification problem; 2) modeling temporal relation among time steps.

*1) Hierarchical Classification:* Because the MRLE task is derived from the MD task to further categorize the music event into foreground music or background music, we observe those event categories of the two tasks naturally form a hierarchy of two levels, which is shown in Fig. 1. Based on this observation, the joint task of MD and MRLE is highly-related to the hierarchical classification problem. For a fragment at one time-step, detecting the two tasks' events can be constructed as categorizing the fragment into event classes of the two hierarchical levels.

*2) Temporal Modeling:* Fragments in an audio recording are temporally correlated, especially those that are close in time. The reason is that the time length of a fragment is short (could be in milliseconds, which is better for localizing the precise onset and offset), but an event may last seconds or minutes. It means that a series of adjacent fragments in a period may belong to the same event class, so we need to ensure the continuity of an event and design a model that can learn the temporal relation among time steps. Recurrent neural network (RNN) has shown superiority in sequence data modeling, so in this work, we use an iterative structure to improve the performance of continuous event detection.

### B. Problem Formulation

As discussed before, the MD task and MRLE task are both event detection and localization problem, and the event categories of the two tasks form a hierarchy of two-level. The fundamental problem formulation is as follows:

Let $\mathcal{D}$ be the training data with $N$ samples:

$$\mathcal{D} := \{(\mathbf{X}_i, \mathbf{Y}_i^1, \mathbf{Y}_i^2) : 0 < i \leq N\} \tag{1}$$

First, each $\mathbf{X}_i$ is assumed to be an observable audio sequence, i.e., $\mathbf{X}_i = (\mathbf{x}_i[t])_{t=1}^{T_i} \in \mathbb{R}^{T_i \times \lambda}$. In our work, this is the spectrogram which is converted from an audio sample (details of the conversion are provided in Section V-B). Here the spectrogram is viewed as a time-series of $T_i$ frames and each frame is $\lambda$-dimensional (we treat each frame as a fragment). Second, the observable variables $\mathbf{Y}_i^1 = (\mathbf{y}_i^1[t])_{t=1}^{T_i} \in \{0,1\}^{T_i \times |C^1|}$ and $\mathbf{Y}_i^2 = (\mathbf{y}_i^2[t])_{t=1}^{T_i} \in \{0,1\}^{T_i \times |C^2|}$ are defined as two binary sequences that indicate the presence of events on the $1^{st}$ and $2^{nd}$ hierarchical level, respectively. The $k^{th}$ position of $\mathbf{y}_i^l[t]$ denotes if the $c_k \in C^l$ event exists for $l = \{1, 2\}$. $|C^1| = 2$ is the number of event classes of the $1^{st}$ level–the MD task level, while $|C^2| = 3$ is the number of event classes of the $2^{nd}$ level–the MRLE task level. Event class *music* on the $1^{st}$ level is the super-class, and *fg-music* and *bg-music* on the $2^{nd}$ level are its corresponding sub-classes. Both of the two event sequences $\mathbf{Y}_i^1$ and $\mathbf{Y}_i^2$ are supposed to be functions of their mutual observation $\mathbf{X}_i$:

$$(\mathbf{y}_i^1[t])_{t=1}^{\tau} = F^1((\mathbf{x}_i[t])_{t=1}^{\tau}), \forall \tau \leq T_i \tag{2}$$

$$(\mathbf{y}_i^2[t])_{t=1}^{\tau} = F^2((\mathbf{x}_i[t])_{t=1}^{\tau}), \forall \tau \leq T_i \tag{3}$$

The functions $F^1$ and $F^2$ together is designed as a neural network $\mathcal{F}$ in Section IV.

The **hierarchical structure** $\mathcal{S}$ (shown in Fig. 1) is defined as:

- $c_{music}^1 \prec c_{fg-music}^2$, where $c_{music}^1 \in C^1$ and $c_{fg-music}^2 \in C^2$
- $c_{music}^1 \prec c_{bg-music}^2$, where $c_{music}^1 \in C^1$ and $c_{bg-music}^2 \in C^2$

where $\prec$ is a partial order representing the PARENT-OF relationship.

Based on the above, the joint task of MD and MRLE is formulated as the following problem:

**Hierarchical Event Detection and Localization (HEDL) problem**: given the training data $\mathcal{D}$ with the hierarchical structure $\mathcal{S}$, our goal is to learn a model $\mathcal{F}$, which can be used to estimate the two hierarchical event sequences $\mathbf{Y}^1 = (\mathbf{y}^1[t])_{t=1}^{T} \in \{0,1\}^{T \times |C^1|}$ and $\mathbf{Y}^2 = (\mathbf{y}^2[t])_{t=1}^{T} \in \{0,1\}^{T \times |C^2|}$ underlying an unseen audio sequence $\mathbf{X} \in \mathbb{R}^{T \times \lambda}$.

## IV. HRIN

We propose the Hierarchical Regulated Iterative Network (HRIN), a two-output deep neural network specifically designed to solve the Hierarchical Event Detection and Localization (HEDL) problem. HRIN propagates gradients from the two network outputs–each one corresponds to each hierarchical level. A corresponding loss function to each output is used for back-propagating the gradients from the event classes in the corresponding level. We use three penalties to regulate the hierarchy. In this section, we first present two variants of HRIN: a recurrent-only (HRIN-r) architecture and a convolutional-recurrent (HRIN-cr) architecture. Then we give a detailed description of the loss function.

### A. Network Architecture

Under the problem formulation discussed in Section III-B, we propose the HRIN and design two variants HRIN-r and HRIN-cr of its network architecture.

*1) HRIN-r:* The overview of HRIN-r, along with its respective notation, is depicted in Fig. 2. The whole HRIN-r is an iterative structure, consisting of $T$ iterative blocks that repeat along the time axis. Next, we give a detailed description of the iterative block at time-step $t$ with its input and outputs.

In one iterative block, data flows along two paths: the first path starts from the input layer, flows through a recurrent unit, and ends up at the output layer for the $1^{st}$ hierarchical level; the second path begins with the same input layer, concatenates with the output of the first path's recurrent unit, passes by another recurrent unit, and terminates at the output layer for the $2^{nd}$ hierarchical level. Thus in one iterative block, two outputs are corresponding to two hierarchical levels, respectively.

In the rest of this subsection, we give the mathematical description of one iterative block (the one at time-step $t$) only; other iterative blocks can be computed in the same way using the same network module with the same parameters. We still follow the notation symbols in Section III-B. But for simplicity, we omit the subscript $i$ in this subsection. The iterative block at time-step $t$ takes as input the $t$-th frame $\mathbf{x}[t] \in \mathbb{R}^{\lambda \times 1}$ of audio spectrogram $\mathbf{X}$. Then $\mathbf{x}[t]$ is computed through two paths.

In the first path, $\mathbf{x}[t]$ is first computed by a recurrent unit whose basic computation cell is GRU [44]. A GRU cell has two inputs and two outputs: it takes in the current external input $\mathbf{x}[t]$ and the last hidden state $\mathbf{h}[t-1]$, and it exports the current output $\mathbf{o}[t]$ and the current hidden state $\mathbf{h}[t]$. Inside a GRU cell, there are four computation steps: first, compute the reset gate $\mathbf{r}[t]$ using weight matrix $\mathbf{W}_r$ and bias vector $\mathbf{b}_r$; second, compute the update gate $\mathbf{z}[t]$ using weight matrix $\mathbf{W}_z$ and bias vector $\mathbf{b}_z$; third, calculate the reset hidden state $\hat{\mathbf{h}}[t]$ using weight matrix $\mathbf{W}_{\hat{\mathbf{h}}}$ and bias vector $\mathbf{b}_{\hat{\mathbf{h}}}$; fourth, update the hidden state and obtain the current one $\mathbf{h}[t]$. At last, the current output $\mathbf{o}[t]$ is computed from the current hidden state $\mathbf{h}[t]$ using weight matrix $\mathbf{W}_o$ and bias vector $\mathbf{b}_o$. To depict the GRU cell specifically for the first path, we add the superscript "1" for every notation. Let $\mathbf{o}^1[t]$ denote the output of the first path's recurrent unit with only one GRU cell and is given by:

$$\mathbf{r}^1[t] = \sigma(\mathbf{W}_r^1(\mathbf{h}^1[t-1] \odot \mathbf{x}[t]) + \mathbf{b}_r^1) \tag{4}$$

$$\mathbf{z}^1[t] = \sigma(\mathbf{W}_z^1(\mathbf{h}^1[t-1] \odot \mathbf{x}[t]) + \mathbf{b}_z^1) \tag{5}$$

$$\hat{\mathbf{h}}^1[t] = tanh(\mathbf{W}_{\hat{\mathbf{h}}}^1((\mathbf{r}^1[t] \cdot \mathbf{h}^1[t-1]) \odot \mathbf{x}[t]) + \mathbf{b}_{\hat{\mathbf{h}}}^1) \tag{6}$$

$$\mathbf{h}^1[t] = (1 - \mathbf{z}^1[t])\mathbf{h}^1[t-1] + \mathbf{z}^1[t] \cdot \hat{\mathbf{h}}^1[t] \tag{7}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3030484, IEEE/ACM Transactions on Audio, Speech, and Language Processing



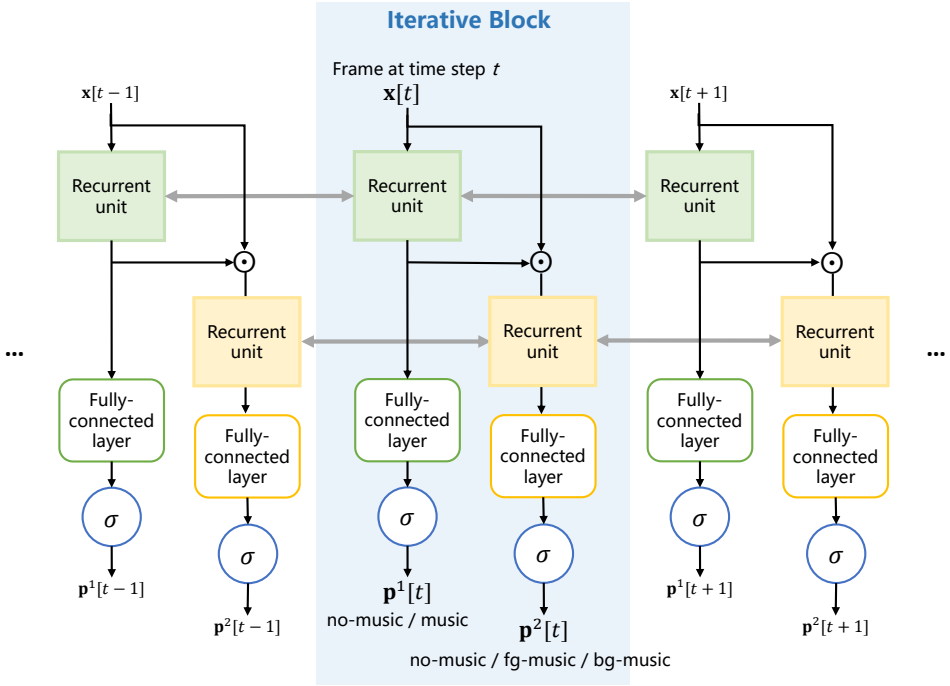



Fig. 2. The network architecture of HRIN-r (better viewed in color). The area under a blue background is an Iterative Block. The details can be found in Section IV-A1.

$$\mathbf{o}^1[t] = \sigma(\mathbf{W}_o^1\mathbf{h}^1[t] + \mathbf{b}_o^1) \tag{8}$$

where $\mathbf{W}_r^1$, $\mathbf{W}_z^1$, $\mathbf{W}_{\hat{\mathbf{h}}}^1$, $\mathbf{W}_o^1$, $\mathbf{b}_r^1$, $\mathbf{b}_z^1$, $\mathbf{b}_{\hat{\mathbf{h}}}^1$ and $\mathbf{b}_o^1$ are the parameters of the first path's recurrent unit. $\odot$ denotes vector concatenation. $\sigma$ is sigmoid activation function and $tanh$ is tangent activation function.

Then $\mathbf{o}^1[t]$ is passed into the subsequent fully-connected layer with sigmoid as the ultimate activation function. Hence the prediction for the $1^{st}$ hierarchical level, $\mathbf{p}^1[t] \in \mathbb{R}^{|C^1|}$, is given by:

$$\mathbf{p}^1[t] = \sigma(\mathbf{W}_p^1\mathbf{o}^1[t] + \mathbf{b}_p^1) \tag{9}$$

where again $\sigma$ is necessarily sigmoidal, $\mathbf{W}_p^1$ is the weight matrix and $\mathbf{b}_p^1$ is the bias vector. The $k^{th}$ position of $\mathbf{p}^1[t]$ denotes probability $P(c_k|\mathbf{x}[t])$ for $c_k \in C^1$. Until now, the formulas (4) to (9) are the explicit expressions of $F^1$ mentioned in (2).

In the second path, instead of merely inputting $\mathbf{x}[t]$ or $\mathbf{o}^1[t]$, we concatenate them both and use the concatenated variable as input to the second path. By doing this, the second path obtains information from not only the raw input but also the first path's output. We denote the concatenated variable as $\mathbf{x}^2[t]$, and it is given by:

$$\mathbf{x}^2[t] = \mathbf{x}[t] \odot \mathbf{o}^1[t] \tag{10}$$

Then $\mathbf{x}^2[t]$ traverses the second path's recurrent unit and output variable $\mathbf{o}^2[t]$:

$$\mathbf{r}^2[t] = \sigma(\mathbf{W}_r^2(\mathbf{h}^2[t-1] \odot \mathbf{x}^2[t]) + \mathbf{b}_r^2) \tag{11}$$

$$\mathbf{z}^2[t] = \sigma(\mathbf{W}_z^2(\mathbf{h}^2[t-1] \odot \mathbf{x}^2[t]) + \mathbf{b}_z^2) \tag{12}$$

$$\hat{\mathbf{h}}^2[t] = tanh(\mathbf{W}_{\hat{\mathbf{h}}}^2((\mathbf{r}^2[t] \cdot \mathbf{h}^2[t-1]) \odot \mathbf{x}^2[t]) + \mathbf{b}_{\hat{\mathbf{h}}}^2) \tag{13}$$

$$\mathbf{h}^2[t] = (1 - \mathbf{z}^2[t])\mathbf{h}^2[t-1] + \mathbf{z}^2[t] \cdot \hat{\mathbf{h}}^2[t] \tag{14}$$

$$\mathbf{o}^2[t] = \sigma(\mathbf{W}_o^2\mathbf{h}^2[t] + \mathbf{b}_o^2) \tag{15}$$

where $\mathbf{W}_r^2$, $\mathbf{W}_z^2$, $\mathbf{W}_{\hat{\mathbf{h}}}^2$ and $\mathbf{W}_o^2$ are weight matrices; $\mathbf{b}_r^2$, $\mathbf{b}_z^2$, $\mathbf{b}_{\hat{\mathbf{h}}}^2$ and $\mathbf{b}_o^2$ are bias vectors. Those are the parameters of the second path's recurrent unit. The basic computation operations of the first path's and the second path's recurrent units are the same, as is shown in Fig. 3; the difference is that they use different parameters.

At last, $\mathbf{o}^2[t]$ is sent into the subsequent fully-connected layer with sigmoid as the final activation function and $\mathbf{p}^2[t] \in \mathbb{R}^{|C^2|}$ is the output:

$$\mathbf{p}^2[t] = \sigma(\mathbf{W}_p^2\mathbf{o}^2[t] + \mathbf{b}_p^2) \tag{16}$$

where once again $\sigma$ is the sigmoidal function. $\mathbf{W}_p^2$ and $\mathbf{b}_p^2$ are the weight matrix and bias vector of the second path's fully connected layer, respectively. $\mathbf{p}^2[t]$ is therefore the prediction for the $2^{nd}$ hierarchical level. The $k^{th}$ position of $\mathbf{p}^2[t]$ denotes probability $P(c_k|\mathbf{x}[t])$ for $c_k \in C^2$. The formulas (10) to (16) are the detailed expressions of $F^2$ mentioned in (3).

In a word, one iterative block of HRIN takes one variable $\mathbf{x}[t]$ as its input and outputs two variables $\mathbf{p}^1[t]$ and $\mathbf{p}^2[t]$, each one representing the prediction per hierarchical level. As for the whole HRIN, it receives $\mathbf{X} = (\mathbf{x}[t])_{t=1}^T$ as the input and gives $\mathbf{P}^1 = (\mathbf{p}^1[t])_{t=1}^T$ and $\mathbf{P}^2 = (\mathbf{p}^2[t])_{t=1}^T$ as the outputs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3030484, IEEE/ACM Transactions on Audio, Speech, and Language Processing



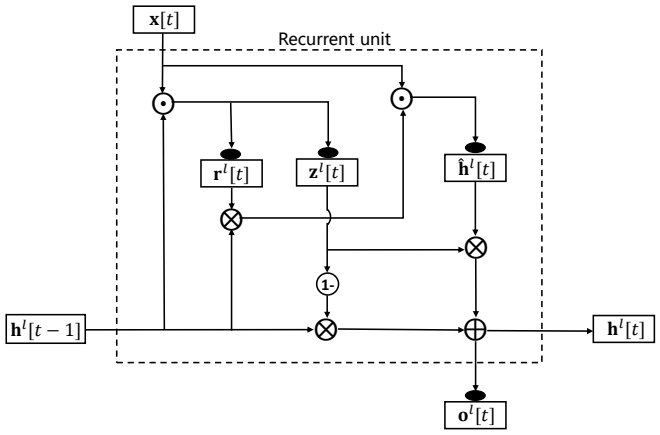



Fig. 3. Diagram of the internal data flows of a simplified version of the recurrent unit. This recurrent unit only contains one unidirectional GRU. The basic computing operations of the recurrent units in the first and second paths are the same. Superscript $l = 1$ corresponds to the first path, and $l = 2$ corresponds to the second path.

*2) HRIN-cr:* We want to enhance the ability to capture input features in our model. Thus we design another HRIN variant called HRIN-cr, a slightly modified version of HRIN-r where a convolutional unit is added before each recurrent unit. The overview of HRIN-cr, along with its respective notation, is depicted in Fig. 4. At the beginning of the first computation path of HRIN-cr, we first compute $\mathbf{x}[t]$ by a convolutional unit, which we denote as a $conv\_unit^1$. Let $\tilde{\mathbf{x}}[t]$ denote the output of the convolutional unit and is given by:

$$\tilde{\mathbf{x}}[t] = conv\_unit^1(\mathbf{x}[t]) \tag{17}$$

Then the rest of computations in the first path are the same as HRIN-r except that all $\mathbf{x}[t]$ are substituted with $\tilde{\mathbf{x}}[t]$ in (4), (5) and (6). In the second path, similarly we denote the convolutional unit as $conv\_unit^2$ and compute:

$$\tilde{\mathbf{x}}^2[t] = conv\_unit^2(\mathbf{x}^2[t]) \tag{18}$$

and other computations in the second path of HRIN-cr are as same as those in the second path of HRIN-r except that all $\mathbf{x}^2[t]$ are replaced by $\tilde{\mathbf{x}}^2[t]$ in (11), (12) and (13). $conv\_unit^1$ and $conv\_unit^2$ are composed of 1-D convolutional layers, and their basic computation operations are the same (details are discussed in Section V-D).

### B. Loss Function

Predictions of the $1^{st}$ hierarchical level are $\mathbf{P}_i^1 = (\mathbf{p}_i^1[t])_{t=1}^{T_i} \in \mathbb{R}^{T_i \times |C^1|}$, and the corresponding ground-truths are $\mathbf{Y}_i^1 = (\mathbf{y}_i^1[t])_{t=1}^{T_i} \in \{0,1\}^{T_i \times |C^1|}$. To learn the information of the $1^{st}$ hierarchical level, we minimize the binary cross-entropy between $\mathbf{P}_i^1$ and $\mathbf{Y}_i^1$:

$$\begin{aligned} L^1 = -\frac{1}{N \times T_i} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \sum_{j=1}^{|C^1|} \Big[ & \mathbf{y}_{ij}^1[t] \cdot log(\mathbf{p}_{ij}^1[t]) \\ & + (1 - \mathbf{y}_{ij}^1[t]) \cdot log(1 - \mathbf{p}_{ij}^1[t]) \Big] \end{aligned} \tag{19}$$

where $N$ means that we have $N$ samples and subscript $i$ denotes the $i^{th}$ sample. $T_i$ is the time length of the $i^{th}$ sample and $t$ represents the $t^{th}$ time-step. $|C^1|$ is the number of event classes of the $1^{st}$ level and subscript $j$ denotes the $j^{th}$ component of vector $\mathbf{y}_i^1[t]$.

For the $2^{nd}$ hierarchical level, the predictions are $\mathbf{P}_i^2 = (\mathbf{p}_i^2[t])_{t=1}^{T_i} \in \mathbb{R}^{T_i \times |C^2|}$ and the corresponding ground-truths are $\mathbf{Y}_i^2 = (\mathbf{y}_i^2[t])_{t=1}^{T_i} \in \{0,1\}^{T_i \times |C^2|}$. Similarly to the $1^{st}$ level, we still use binary cross-entropy to define the loss between $\mathbf{P}_i^2$ and $\mathbf{Y}_i^2$ to learn the information of the $2^{nd}$ hierarchical level:

$$\begin{aligned} L^2 = -\frac{1}{N \times T_i} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \sum_{j=1}^{|C^2|} \Big[ & \mathbf{y}_{ij}^2[t] \cdot log(\mathbf{p}_{ij}^2[t]) \\ & + (1 - \mathbf{y}_{ij}^2[t]) \cdot log(1 - \mathbf{p}_{ij}^2[t]) \Big] \end{aligned} \tag{20}$$

where $|C^2|$ is the number of event classes of the $2^{nd}$ level and subscript $j$ denotes the $j^{th}$ component of vector $\mathbf{y}_i^2[t]$.

Although the concatenation (10) makes sure that the information from the $1^{st}$ level plays the role of guiding for the information from $2^{nd}$ level, the guarantee for the hierarchy is not enough. There still exist hierarchical violations. A hierarchical violation is a phenomenon that the predicted score of a child node is larger than the prediction of its parent node [45, 46]. It means that if the model is not confident of detecting *music* event in a frame, it could not be any more confident of detecting *fg-music* or *bg-music* event in that frame. In the hierarchical tree–as is shown in Fig. 1– of the joint task of MD and MRLE, two potential hierarchical violations are: 1) the prediction score of *fg-music* is greater than the score of *music*, and 2) the score of *bg-music* is greater than that of *music*. To regulate these two hierarchical violations, we present the two penalty losses. Similar to [45, 46], these penalty losses are defined as:

$$L^{fg} = \frac{1}{N \times T_i} \sum_{i=1}^{N} \sum_{t=1}^{T_i} max\{0, \mathbf{p}_{i2}^2[t] - \mathbf{p}_{i2}^1[t]\}^2 \tag{21}$$

$$L^{bg} = \frac{1}{N \times T_i} \sum_{i=1}^{N} \sum_{t=1}^{T_i} max\{0, \mathbf{p}_{i3}^2[t] - \mathbf{p}_{i2}^1[t]\}^2 \tag{22}$$

Apart from the above penalties, we add another regularization term to constrain the prediction score of *non-music*. Different from *fg-music bg-music* and *music*, the *non-music* node in the $1^{st}$ hierarchical level has only one child which is itself. Therefore, the prediction scores of *non-music* in the $1^{st}$ and $2^{nd}$ hierarchical level should be equal. We use mean squared error to ensure this equality and present another penalty loss:

$$L^{no} = \frac{1}{N \times T_i} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \frac{1}{2} (\mathbf{p}_{i1}^2[t] - \mathbf{p}_{i1}^1[t])^2 \tag{23}$$

The final loss function of HRIN is to optimize:

$$\min_{\theta} (L^1 + L^2 + \alpha L^{fg} + \beta L^{bg} + \gamma L^{no}) \tag{24}$$

where we employ penalty factors $\alpha$, $\beta$, and $\gamma$ for balancing the strength of each penalty loss. When penalty factors are too large, the optimization process might be affected. On the contrary, if penalty factors are too small, the network would lack the ability to learn the innate trait of hierarchy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3030484, IEEE/ACM Transactions on Audio, Speech, and Language Processing



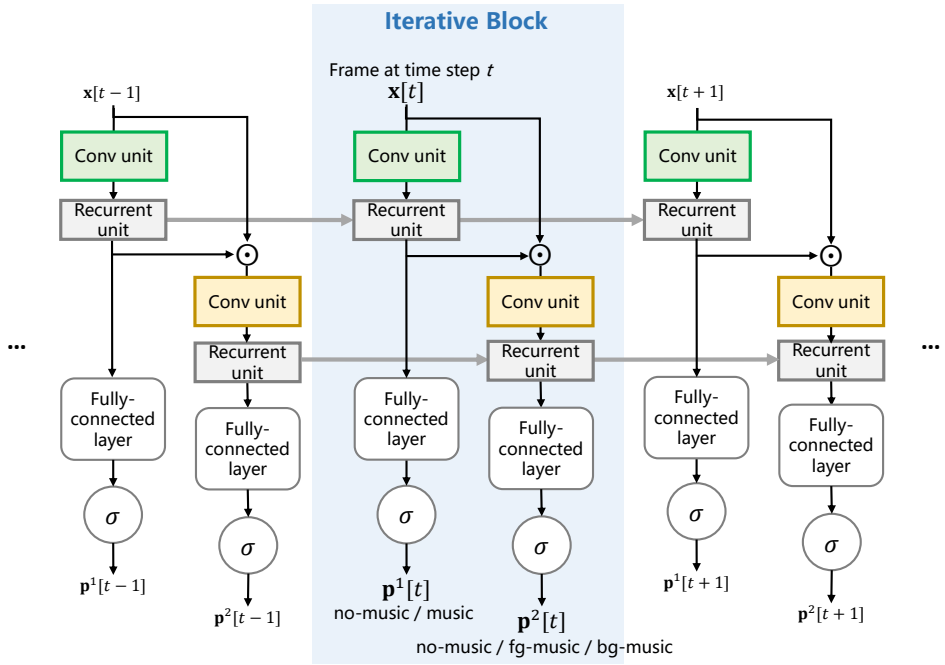



Fig. 4. The network architecture of HRIN-cr (better viewed in color). The area under the blue background is an Iterative Block. The details can be found in Section IV-A2.

## V. Experiments

### A. Dataset

We use the OpenBMAT (Open Broadcast Media Audio from TV) dataset [2] to train and evaluate our proposed model. OpenBMAT was released in the year 2019. It is an open dataset for the task of music detection that brings together all the appropriate characteristics for this task and for the task of estimating the music's relative loudness. OpenBMAT is the only large dataset that includes annotations about the relative loudness of music; other datasets either are not large enough or contain only the annotations of the music detection task [1, 4, 47-49]. OpenBMAT dataset includes 27.4 hours of broadcast audio from eight different TV program types (children, documentary, entertainment, music, news, series & films, sports, and talk) of four countries (France, Germany, Spain, and the United Kingdom). It consists of 1647 one-minute-long audio excerpts, and each audio excerpt has two corresponding annotations–one for MD task and the other for the MRLE task. We treat one *(audio excerpt, MD annotation, MRLE annotation)* triple as one sample, and use 80 percent samples as the training set. The rest of the samples in the dataset are split equally as the development set and the test set, respectively.

### B. Pre-processing

Before sent into the HRIN model, each audio needs to be pre-processed. First, each audio is resampled to 8000 Hz and converted to mono audio. A sliding window with 512 frame size and 128 hop size is used to convert each audio into a spectrogram. Then a mel filter bank with 128 frequency bins is applied on the spectrogram to obtain the mel-spectrogram. Finally, the mel-spectrogram is converted to a logarithmic scale. We normalize the log mel-spectrogram to a zero mean and unit variance, and these two values are computed on the training set. For one-minute-long audio, its corresponding log mel-spectrogram obtained by the above pre-processing is a matrix of size $128 \times 3751$. The two dimensions of this matrix are time and frequency, respectively. This matrix is what we denote as $\mathbf{X}$ (subscript is omitted) in Section III-B, which can be viewed as a time-series of 3751 frames, and each frame is a 128-dimensional vector.

Each annotation also needs to be processed as the event sequence so that it can be computed mathematically in HRIN. Initially, an annotation file in the dataset contains multiple rows, and each row contains the tab-separated onset (seconds), offset (seconds), and the event class (rows are ordered by onset time). Examples of MD annotation and MRLE annotation are shown in Fig. 5. Since each frame represents 0.016-second audio (computed via $hop\_size/sampling\_rate$), we can calculate the onset and offset of each frame in seconds and find the event class of this frame in the annotation file. For each frame, we use a 2-dimension 0-1 vector to represent its MD event and 3-dimension 0-1 vector to represent its MRLE event. Finally, we get two binary sequences that indicate the MD events and MRLE events for each frame sequence; the two binary sequences are what we denote as $\mathbf{Y}^1$ and $\mathbf{Y}^2$ (subscripts are omitted) in Section III-B.

### C. Post-processing

The outputs $\mathbf{P}^1$ and $\mathbf{P}^2$ of HRIN are sequences of predicted probabilities for the two hierarchical levels. To get the predicted annotations, first, a threshold of 0.5 is applied to every element of $\mathbf{P}^1$ and $\mathbf{P}^2$ so that we get two sequences of 0-1

| | | |
|---|---|---|
| 0.0 | 49.59 | no-music |
| 49.59 | 60.0 | music |

(a) Music detection annotation example.

| | | |
|---|---|---|
| 0.0 | 49.59 | no-music |
| 49.59 | 56.09 | bg-music |
| 56.09 | 57.17 | fg-music |
| 57.17 | 60.0 | bg-music |

(b) Music relative loudness estimation annotation example.

Fig. 5. Examples of MD annotation and MRLE annotation of the audio "France_Documentary_20283948_158691662.wav".

vector $\hat{\mathbf{Y}}^1$ and $\hat{\mathbf{Y}}^2$ for MD task and MRLE task respectively. For each task, we can find out the predicted event class of each frame per time-step and calculate its onset and offset in seconds. Concatenating the onsets and offsets of consecutive events that belong to the same category, we obtain the final annotation for each task like the ones shown in Fig. 5.

### D. Experiment setting

We implement the proposed model using PyTorch on a single NVIDIA TITAN Xp GPU card. For HRIN-r, we design one recurrent unit architecture as 4-layer bidirectional GRU with the hidden size of 100 and no dropout. Recurrent units on two computation paths use the same architecture but different parameters. We use a mini-batch size of 32 to fully utilize the single card GPU with 12 GB RAM in training. Adam optimizer [50] with an initial learning rate of 0.0001 and a learning rate decay strategy is used for its stable convergence. For HRIN-cr, we design one recurrent unit architecture as 2-layer bidirectional GRU with the hidden size of 50 and no dropout. We design one convolutional unit as three 1-D convolutional layers. Each convolutional layer is followed by a batch normalization layer [51], a ReLU activation function [52], and a max-pooling layer. The convolutional unit's output is flattened (reshaped) along the channel and frequency dimension before fed into the recurrent unit. The numbers of feature maps of the convolutional layers are 64, 128, and 256, respectively. Kernel sizes are 3, and strides are 2. In training, we set mini-batch size as 8 and use Adam optimizer with an initial learning rate of 0.0001 and a learning rate decay strategy. For both HRIN-r and HRIN-cr, we set all penalty factors $\alpha$, $\beta$, and $\gamma$ to 1. We release the PyTorch implementation of our code online [1].

### E. Evaluation metrics

For both the music detection task and music relative loudness estimation task, there are two ways to measure the performance: segment-level evaluation and event-level evaluation. Both of the evaluations are performed between the reference annotation (i.e., ground truth) and the estimation annotation (i.e., prediction).

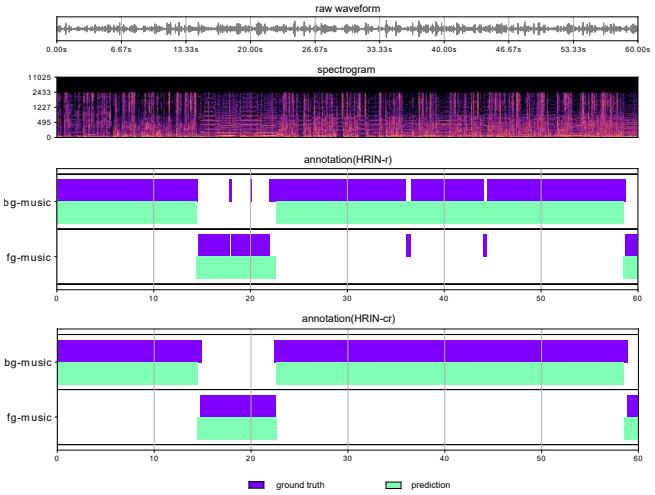
*1) Segment-level evaluation:* In the segment-level evaluation, each annotation is split into segments of 10ms. First, we compute the intermediate statistics for each event class C, which include: True Positives (TPc): both the reference segment's and the estimation segment's class is C; False Positives (FPc): the reference segment's class is not C, but the estimation segment's class is C; True Negatives (TNc): both the reference segment's and the estimation segment's class is not C; False Negatives (FNc): the reference segment's class is C, but the estimation segment's class is not C. Based on these basic statistics, we then report class-wise Precision (Pc), Recall (Rc) and F-measure (Fc):

$$Pc = \frac{TPc}{TPc + FPc} \tag{25}$$

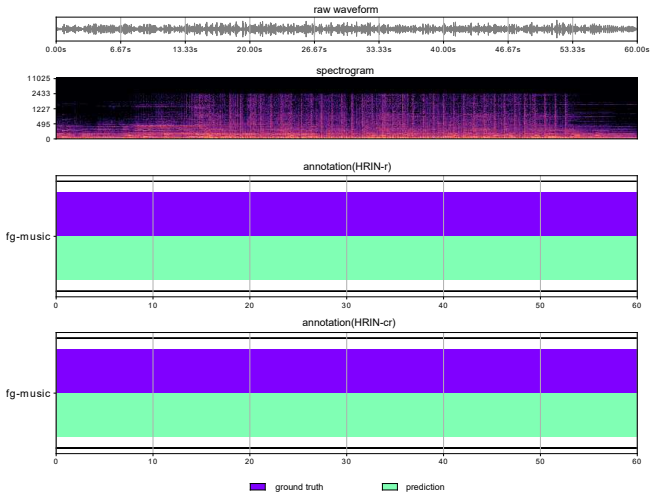
$$Rc = \frac{TPc}{TPc + FNc} \tag{26}$$

$$Fc = \frac{2 \times Pc \times Rc}{Pc + Rc} \tag{27}$$

[1]https://github.com/jiabijue/md_mrle




(a) Visualization of "France_Documentary_26950733_173548309".




(b) Visualization of "Spain_Documentary_101066_71587093".

Fig. 6. Visualizations of two audio samples from the test set, regarding the raw signal, spectrogram, and annotations. Annotations show the comparisons between the predictions from HRIN-r and HRIN-cr and the corresponding ground-truths. We only illustrate annotations on the MRLE task in the figure.

TABLE I

COMPARISONS WITH BASELINES AND DIFFERENT NETWORK ARCHITECTURES ON THE MD TASK. SEGMENT-LEVEL EVALUATIONS: OVERALL ACCURACY (ACC), CLASS-WISE F-MEASURE (CLASS_F). EVENT-LEVEL EVALUATIONS: CLASS-WISE F-MEASURE (500 MS TOLERANCE, ONSET-ONLY) (CLASS_F_500_ON). *Shown are averages±standard_deviations across tenfold cross-validation runs with random seeds. The bold numbers are significantly better according to a paired t-test.*

| | Segment-level | | | Event-level | |
|---|---|---|---|---|---|
| | Acc | Music_F | Non-Music_F | Music_F_500_on | Non-Music_F_500_on |
| MMG [2] | 0.8895 | 0.8860 | 0.8928 | - | - |
| LSTM | 0.8277±0.0161 | 0.8054±0.029 | 0.8401±0.0173 | 0.1908±0.0537 | 0.3128±0.0610 |
| GRU | 0.8620±0.0113 | 0.8521±0.0160 | 0.8669±0.0116 | 0.2221±0.0283 | 0.3514±0.0473 |
| CNN-GRU | 0.8633±0.0358 | 0.8360±0.0601 | 0.8802±0.0248 | 0.1469±0.0206 | 0.2830±0.0435 |
| HRIN-r (LSTM) | 0.8556±0.0168 | 0.8540±0.0237 | 0.8560±0.0145 | 0.3649±0.0559 | 0.3511±0.0608 |
| HRIN-r (proposed) | 0.8766±0.0114 | 0.8766±0.0147 | 0.8771±0.0120 | **0.4037**±0.0591 | **0.3902**±0.0418 |
| HRIN-cr (proposed) | **0.8929**±0.0110 | **0.8873**±0.0145 | **0.8972**±0.0119 | 0.3013±0.0532 | 0.3079±0.0552 |

TABLE II

COMPARISONS WITH BASELINES AND DIFFERENT NETWORK ARCHITECTURES ON THE MRLE TASK. *Shown are averages±standard_deviations across tenfold cross-validation runs with random seeds. The bold numbers are significantly better according to a paired t-test.*

| | Segment-level | | | | Event-level | | |
|---|---|---|---|---|---|---|---|
| | Acc | Fg-Music_F | Bg-Music_F | Non-Music_F | Fg-Music_F_500_on | Bg-Music_F_500_on | Non-Music_F_500_on |
| MMG [2] | 0.8271 | 0.7360 | **0.7728** | 0.8901 | - | - | - |
| LSTM | 0.8469±0.0127 | 0.6258±0.0831 | 0.6996±0.0411 | 0.8401±0.0173 | 0.1932±0.0231 | 0.2019±0.0325 | 0.3128±0.0610 |
| GRU | 0.8750±0.0058 | 0.6803±0.0676 | 0.7567±0.0158 | 0.8669±0.0116 | 0.3260±0.0247 | 0.2483±0.0211 | 0.3514±0.0473 |
| CNN-GRU | **0.8800**±0.0232 | 0.7279±0.0674 | 0.7451±0.0661 | 0.8802±0.0248 | 0.2444±0.0479 | 0.1644±0.0175 | 0.2830±0.0435 |
| HRIN-r (LSTM) | 0.8632±0.0075 | 0.6634±0.0567 | 0.7328±0.0245 | 0.8599±0.0122 | 0.2097±0.0437 | 0.1945±0.0297 | 0.2687±0.0379 |
| HRIN-r (proposed) | 0.8798±0.0107 | 0.7227±0.0414 | **0.7712**±0.0181 | 0.8788±0.0137 | **0.3687**±0.0615 | **0.3359**±0.0274 | **0.4856**±0.0273 |
| HRIN-cr (proposed) | **0.8843**±0.0112 | **0.7363**±0.0430 | 0.7580±0.0553 | **0.8941**±0.0106 | 0.2324±0.0911 | 0.2279±0.0666 | 0.4578±0.0518 |

As well as the overall Accuracy (Acc):

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{28}$$

where TP is the sum of TPc over all event classes C; and FP, TN, FN are obtained in the same way.

*2) Event-level evaluation:* In the event-level evaluation, we consider each annotated segment of the ground truth as an event. Firstly, we compute the intermediate statistics for the onsets (and offsets) of each class C, which include: True Positives (TPc): an estimation event of class C that starts (and ends) at the same temporal positions as a reference event of class C, taking into account a tolerance time-window. False Positives (FPc): an estimation event of class C that starts (and ends) at temporal positions where no reference event of class C does, taking into account a tolerance time-window. False Negatives (FNc): a reference event of class = C that starts (and ends) at temporal positions where no estimation event of class C does, taking into account a tolerance time-window. The frequently-used tolerance time-windows are +/- 500 ms, +/- 1000 ms. Secondly, based on these statistics, we report class-wise Precision, Recall, and F-measure in event-level evaluation, which is computed in the same way as in segment-level evaluation.

### *F. Baseline model*

We compare our model with several baseline models, and summarize the MD task results and MRLE task results in Table I and Table II respectively. We first compare our model with the baseline model called MMG on OpenBMAT dataset [2]. MMG only has segment-level evaluations, so the event-level evaluations are not listed. Also, it's not clear that if MMG reports the best or average results on multiple runs, and we use the average values to compare its scores. We can observe that both the HRIN-r and HRIN-cr obtain slightly better results than MMG on most of the metrics and gets superior results on the accuracy of the MRLE task. We can see from Table I that for the MD task, the proposed HRIN-cr outperforms MMG on all segment-level metrics. We can observe from Table II that HRIN-r outperforms MMG on accuracy and achieves comparable results on *bg-music* f-measure regarding the segment-level evaluations of MRLE task. Moreover, HRIN-cr beats MMG on three out of four segment-level metrics: accuracy, *fg-music* f-measure, and *non-music* f-measure.

We build another three baseline models LSTM, GRU, and CNN-GRU, which have very similar network architectures as our proposed HRIN except that they have only one computation path and one output for the $2^{nd}$ hierarchical level only. The $1^{st}$ level's predictions are computed by combining the predictions of event class *fg-music* and *bg-music* to generate the prediction of *music*, at the same time keeping the prediction of event class *non-music*. In the proposed models, predictions of the $2^{nd}$ hierarchical level can be viewed as local predictions, and those of the $1^{st}$ hierarchical level can be viewed as global predictions. The three baseline models only perform local learning, attempting to discover the specificity that dictates the class relationships in particular regions of the class hierarchy, later combining the local predictions to generate the final classification. So the three baselines' mechanism is different

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3030484, IEEE/ACM Transactions on Audio, Speech, and Language Processing



TABLE III
IMPACT OF HIERARCHICAL PENALTIES ON THE MD TASK. *Shown are averages±standard_deviations across 10-fold cross-validation runs with random seeds. The bold numbers are significantly better according to a paired t-test.*

| | Segment-level | | | Event-level | |
|---|---|---|---|---|---|
| | Acc | Music_F | Non-Music_F | Music_F_500_on | Non-Music_F_500_on |
| without $L^{no}$, $L^{fg}$, $L^{bg}$ | 0.8663±0.0182 | 0.8632±0.0276 | 0.8675±0.0138 | 0.3408±0.0459 | 0.3154±0.0293 |
| without $L^{fg}$, $L^{bg}$ | **0.8731**±0.0190 | 0.8708±0.0238 | **0.8742**±0.0178 | 0.3462±0.0404 | 0.3381±0.0418 |
| without $L^{no}$ | **0.8737**±0.0101 | **0.8727**±0.0136 | **0.8743**±0.0119 | 0.3451±0.0287 | 0.3216±0.0305 |
| HRIN-r | **0.8766**±0.0114 | **0.8766**±0.0147 | **0.8771**±0.0120 | **0.4037**±0.0591 | **0.3902**±0.0418 |

TABLE IV
IMPACT OF HIERARCHICAL PENALTIES ON THE MRLE TASK. *Shown are averages±standard_deviations across 10-fold cross-validation runs with random seeds. The bold numbers are significantly better according to a paired t-test.*

| | Segment-level | | | | Event-level | | |
|---|---|---|---|---|---|---|---|
| | Acc | Fg-Music_F | Bg-Music_F | Non-Music_F | Fg-Music_F_500_on | Bg-Music_F_500_on | Non-Music_F_500_on |
| without $L^{no}$, $L^{fg}$, $L^{bg}$ | 0.8741±0.0117 | **0.7247**±0.0543 | 0.7530±0.0366 | 0.8678±0.0149 | 0.3542±0.0575 | 0.3140±0.0442 | 0.4491±0.0511 |
| without $L^{fg}$, $L^{bg}$ | **0.8777**±0.0130 | 0.7062±0.0312 | 0.7656±0.0299 | **0.8755**±0.0172 | 0.3383±0.0483 | 0.3164±0.0254 | 0.4666±0.0287 |
| without $L^{no}$ | **0.8785**±0.0089 | 0.7157±0.0500 | 0.7633±0.0224 | **0.8760**±0.0133 | **0.3652**±0.0479 | **0.3328**±0.0265 | **0.4838**±0.0419 |
| HRIN-r | **0.8798**±0.0107 | **0.7227**±0.0414 | **0.7712**±0.0181 | **0.8788**±0.0137 | **0.3687**±0.0615 | **0.3359**±0.0274 | **0.4856**±0.0273 |

TABLE V
COMPARISON OF DIFFERENT CONCATENATIONS ON THE MD TASK. *Shown are averages±standard_deviations across 10-fold cross-validation runs with random seeds. The bold numbers are significantly better according to a paired t-test.*

| | Segment-level | | | Event-level | |
|---|---|---|---|---|---|
| | Acc | Music_F | Non-Music_F | Music_F_500_on | Non-Music_F_500_on |
| HRIN-cr (concate after conv) | 0.8731±0.0095 | 0.8604±0.0206 | 0.8738±0.0169 | **0.3333**±0.0278 | **0.3658**±0.0265 |
| HRIN-cr (concate before fc) | 0.8706±0.0248 | 0.8528±0.0324 | 0.8743±0.0196 | **0.3304**±0.0514 | 0.3589±0.0150 |
| HRIN-cr | **0.8929**±0.0110 | **0.8873**±0.0145 | **0.8972**±0.0119 | 0.3013±0.0532 | 0.3079±0.0552 |

TABLE VI
COMPARISON OF DIFFERENT CONCATENATIONS ON THE MRLE TASK. *Shown are averages±standard_deviations across 10-fold cross-validation runs with random seeds. The bold numbers are significantly better according to a paired t-test.*

| | Segment-level | | | | Event-level | | |
|---|---|---|---|---|---|---|---|
| | Acc | Fg-Music_F | Bg-Music_F | Non-Music_F | Fg-Music_F_500_on | Bg-Music_F_500_on | Non-Music_F_500_on |
| HRIN-cr (concate after conv) | 0.8778±0.0080 | 0.7141±0.0364 | 0.7306±0.0286 | 0.8777±0.0121 | 0.2159±0.0514 | 0.2219±0.0207 | 0.4296±0.0246 |
| HRIN-cr (concate before fc) | 0.8738±0.0179 | 0.7015±0.0616 | 0.7369±0.0414 | 0.8761±0.0192 | 0.2175±0.0680 | 0.2245±0.0408 | 0.4141±0.0239 |
| HRIN-cr | **0.8843**±0.0112 | **0.7363**±0.0430 | **0.7580**±0.0553 | **0.8941**±0.0106 | **0.2324**±0.0911 | 0.2279±0.0666 | **0.4578**±0.0518 |

TABLE VII
COMPARISONS OF DIFFERENT PENALTY FACTORS ON THE MD TASK. *Shown are averages±standard_deviations across 10-fold cross-validation runs with random seeds.*

| Penalty factors | Segment-level | | | Event-level | |
|---|---|---|---|---|---|
| $(\alpha, \beta, \gamma)$ | Acc | Music_F | Non-Music_F | Music_F_500_on | Non-Music_F_500_on |
| (0.5, 0.5, 0.5) | 0.8506±0.0185 | 0.8529±0.0215 | 0.8582±0.0153 | 0.3396±0.0048 | 0.3189±0.0033 |
| (0.8, 0.8, 0.6) | 0.8568±0.0260 | 0.8603±0.0299 | 0.8629±0.0219 | 0.3475±0.0073 | 0.3106±0.0136 |
| (1.0, 1.0, 1.0) | **0.8766**±0.0114 | **0.8766**±0.0147 | **0.8771**±0.0120 | **0.4037**±0.0591 | 0.3902±0.0418 |
| (1.1, 1.1, 1.0) | 0.8647±0.0286 | 0.8581±0.0321 | 0.8613±0.0250 | 0.3502±0.0397 | 0.3066±0.0220 |
| (1.3, 1.3, 0.7) | 0.8676±0.0268 | 0.8615±0.0294 | 0.8635±0.0240 | 0.3440±0.0088 | 0.3908±0.0066 |
| (1.5, 1.5, 1.1) | 0.8619±0.0224 | 0.8638±0.0249 | 0.8602±0.0198 | 0.3415±0.0074 | **0.3993**±0.0106 |

from our proposed model because our model has a more explicitly supervised signal to learn the hierarchy. These are demonstrated in Table I and Table II: the proposed models achieve better results than three baselines on all metrics of MD and on most metrics of MRLE.

We also add another baseline model HRIN-r (LSTM), which is a modification of HRIN-r, and it uses LSTM as the backbone of the recurrent unit instead. The comparisons are shown in Table I and II. As we can observe, our model outperforms the four baselines on most metrics for the MD task and MRLE task. Specifically, if we make several comparison pairs: LSTM and HRIN-r (LSTM), GRU and HRIN-r, and CNN-GRU and HRIN-cr, we can see that all HRINs obtain better results than their counterparts on the majority of evaluations.

TABLE VIII
COMPARISONS OF DIFFERENT PENALTY FACTORS ON THE MRLE TASK. *Shown are averages±standard_deviations across 10-fold cross-validation runs with random seeds.*

| Penalty factors | Segment-level | | | | Event-level | | |
|---|---|---|---|---|---|---|---|
| $(\alpha, \beta, \gamma)$ | Acc | Fg-Music_F | Bg-Music_F | Non-Music_F | Fg-Music_F_500_on | Bg-Music_F_500_on | Non-Music_F_500_on |
| (0.5, 0.5, 0.5) | 0.8604±0.0004 | 0.6882±0.0850 | 0.7195±0.0014 | 0.8589±0.0157 | 0.3412±0.0101 | 0.2710±0.0129 | 0.4240±0.0174 |
| (0.8, 0.8, 0.6) | 0.8640±0.0158 | 0.6921±0.0285 | 0.7406±0.0261 | 0.8683±0.0230 | 0.3579±0.0236 | 0.2912±0.0106 | 0.4501±0.0301 |
| (1.0, 1.0, 1.0) | **0.8798**±0.0107 | **0.7227**±0.0414 | **0.7712**±0.0181 | **0.8788**±0.0137 | **0.3687**±0.0615 | **0.3359**±0.0274 | **0.4856**±0.0273 |
| (1.1, 1.1, 1.0) | 0.8591±0.0155 | 0.6891±0.0173 | 0.7467±0.0246 | 0.8629±0.0267 | 0.3546±0.0172 | 0.3016±0.0393 | 0.4246±0.0496 |
| (1.3, 1.3, 0.7) | 0.8601±0.0133 | 0.6927±0.0222 | 0.7470±0.0185 | 0.8684±0.0237 | 0.3576±0.0351 | 0.3024±0.0255 | 0.4349±0.0286 |
| (1.5, 1.5, 1.1) | 0.8573±0.0133 | 0.6996±0.0125 | 0.7385±0.0187 | 0.8620±0.0224 | 0.3490±0.0203 | 0.2965±0.0309 | 0.4206±0.0141 |

Two typical audio files from the test set are visualized in Fig. 6 in the forms of the raw waveform, spectrogram, and annotations for the MRLE task. The first audio contains more than one events and there exists event switching (e.g., from *bg-music* to *fg-music* then to *bg-music*). We observe from Fig. 6a that both HRIN-r and HRIN-cr are able to capture the persistence of a certain event in general. However, they lack the ability to precisely detect the onset and offset of an event, which explains why the event-level evaluations are much lower than segment-level evaluations, and this situation not only appears in our model but also exists in baseline models. The second audio consists of only one event throughout the whole time, and from Fig. 6b, we find that both HRIN-r and HRIN-cr are able to deal with this kind of situation well.

### G. Ablation study

We discuss the effectiveness of our proposed model from different aspects. The detailed ablation studies are given in the following.

*1) Analysis of penalty losses:* We first analyze the benefit of penalty losses $L^{fg}$, $L^{bg}$ and $L^{no}$ used in our proposed model. In each experiment, we select HRIN-r as our network backbone. We run the ablation experiments of three settings: the first setting is removing all penalty losses, the second setting is using only the penalty loss for *non-music*, and the third setting is using only the penalty losses for *fg-music* and *bg-music*. Comparing experiments under the first and second setting (i.e., row "without $L^{no}$, $L^{fg}$, $L^{bg}$" and row "without $L^{fg}$, $L^{bg}$" in Table III), we can see that by using $L^{no}$, the performances on all metrics of MD task are promoted. Further, we can observe in the same table that by adding $L^{fg}$ and $L^{bg}$ (i.e., row "without $L^{no}$, $L^{fg}$, $L^{bg}$" comparing with row "without $L^{no}$"), results on all metrics of MD task are also increased. On the whole, by adding three penalties, our model outperforms the ablated model on the MD task, especially significantly improving event-level evaluations. On MRLE task, we can see comparison results in Table IV. Most metrics of the ablated models are surpassed by the proposed model except for segment-level *fg-music* f-measure. The reason might be that *fg-music* is already distinctly enough to be classified than that of *bg-music* and *non-music* because the foreground music volume in audio is high. Therefore, the loss term may have little effect on *fg-music*. All in all, we can find that the performances have appreciable gains after adding three penalties, which reveals that the design of penalty losses distinctly improves performance.

*2) Comparison of Different Concatenations:* We analyze different concatenations to demonstrate the advantages of the proposed HRIN-cr. We conduct experiments on three concatenation strategies, i.e., (a) concatenating the raw input and the output of the recurrent unit at the first path (this is also the default concatenation approach used in this paper), (b) concatenating the outputs of the convolutional units at two paths (denote as "concat after conv"), and (c) concatenating the outputs of the recurrent units at two paths (denote as "concat before fc"). To better illustrate the difference between these three concatenations, we visually show these three strategies in Fig. 7. From Table V–VI, one could observe that different concatenations lead to different performances. Specifically, our default strategy (HRIN-cr) significantly better than the other two strategies in terms of 10 out of 12 performance metrics.

*3) Influence of parameters:* We then analyze the influence of the penalty factors. We run experiments by searching different factor combinations and then compare the performances of different factor values. In the experiments, we select HRIN-r as our network backbone. The comparison results for the MD task and MRLE task are illustrated in Table VII and Table VIII respectively. As shown in the two tables, with different settings of penalty factor values, the performances are also different. We can see that the setting $\alpha = 1, \beta = 1, \gamma = 1$ achieves better results on four evaluations for the MD task and all evaluations for the MRLE task.

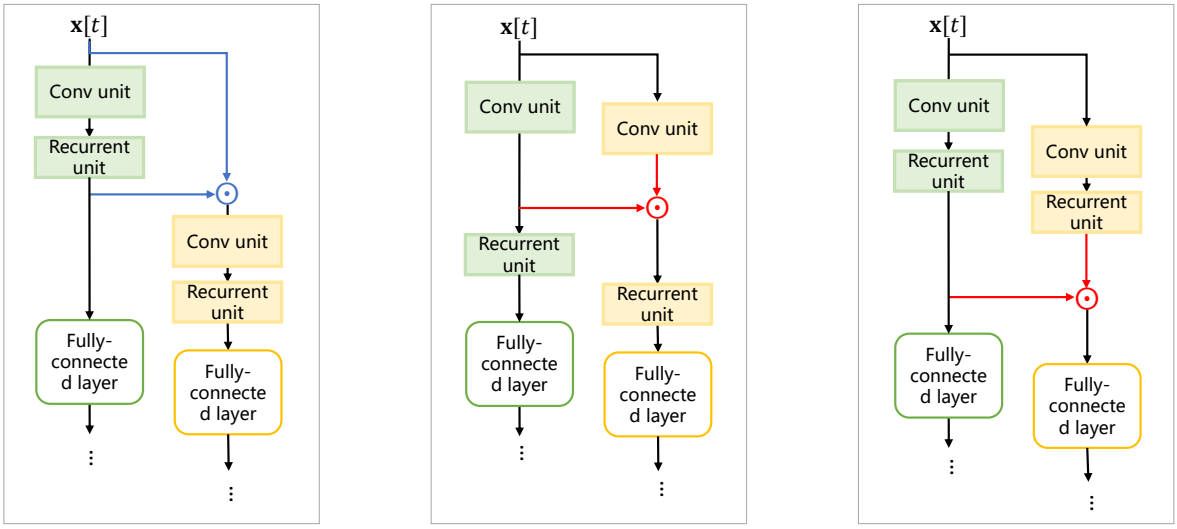
## VI. CONCLUSION

To facilitate music copyright management, we proposed reformulating the joint task of music detection and music relative loudness estimation as hierarchical event detection and localization problem. Accordingly, we propose a novel model, termed as Hierarchical Regulated Iterative Network (HRIN), which enjoys the temporality and hierarchy characteristics of the joint task to solve the problem. To the best of our knowledge, HRIN could be the first approach to model the hierarchical characteristic of this joint task. To verify our method's effectiveness, we carry out extensive experiments on the OpenBMAT dataset, which is the largest dataset in the community.

In the future, we plan to use transfer learning to implement this. Because the goal of transfer learning is to utilize datasets (usually are sufficient) or models of existing tasks and apply them on other correlative tasks (may not have sufficient data or labels), transfer learning could be an excellent choice to it.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TASLP.2020.3030484, IEEE/ACM Transactions on Audio, Speech, and Language Processing






(a) default (b) concat after conv (c) concat before fc

Fig. 7. Comparisons of three different concatenations.

The task of music/speech detection is different from the MD task; however, they are correlative. Under transfer learning, the first step is to use a neural network to learn some features from music/speech datasets, and the second step is to design a model to solve the joint task of MD and MRLE while utilizing the learned features. How to do it explicitly is leaving for future work.

## VII. Acknowledgement

The authors would like to thank all anonymous reviewers for their suggestions to improve this paper.

## References

[1] K. Seyerlehner, T. Pohle, M. Schedl, and G. Widmer, "Automatic music detection in television productions," in *Proc. of the 10th International Conference on Digital Audio Effects (DAFx'07)*. Citeseer, 2007.

[2] B. Meléndez-Catalán, E. Molina, and E. Gómez, "Open broadcast media audio from tv: A dataset of tv broadcast audio with relative music loudness annotations," *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, 2019.

[3] J. Saunders, "Real-time discrimination of broadcast speech/music," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 2. IEEE, 1996, pp. 993–996.

[4] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *1997 IEEE international conference on acoustics, speech, and signal processing*, vol. 2. IEEE, 1997, pp. 1331–1334.

[5] M. J. Carey, E. S. Parris, and H. Lloyd-Thomas, "A comparison of features for speech, music discrimination," in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, vol. 1. IEEE, 1999, pp. 149–152.

[6] L. Ericsson, *Automatic speech/music discrimination in audio files*. Citeseer, 2010.

[7] J. Schlüter and R. Sonnleitner, "Unsupervised feature learning for speech and music detection in radio broadcasts," in *Proceedings of the 15th International Conference on Digital Audio Effects*, 2012.

[8] D. de Benito-Gorron, A. Lozano-Diez, D. T. Toledano, and J. Gonzalez-Rodriguez, "Exploring convolutional, recurrent, and hybrid deep neural networks for speech and music detection in a large audio dataset," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2019, no. 1, p. 9, 2019.

[9] Q. Lemaire and A. Holzapfel, "Temporal convolutional networks for speech and music detection in radio broadcast," in *20th International Society for Music Information Retrieval Conference, Delft, The Netherlands, 2019*. ISMIR, 2019, pp. 229–236.

[10] B. Meléndez-Catalán, E. Molina, and E. Gomez, "Music and/or speech detection mirex 2018 submission," *Music Information Retrieval Evaluation eX-change*, 2018.

[11] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[12] G. Peeters and H. Papadopoulos, "Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1754–1769, 2010.

[13] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks." in *ISMIR*, 2016, pp. 255–261.

[14] R. Vogl, M. Dorfer, G. Widmer, and P. Knees, "Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks." in *ISMIR*, 2017, pp. 150–157.

[15] R. M. Bittner, B. McFee, and J. P. Bello, "Multitask learning for fundamental frequency estimation in music," *arXiv preprint arXiv:1809.00381*, 2018.

[16] S. Böck, M. E. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other," in *20th International Society for Music Information Retrieval Conference (ISMIR 2019)*, 2019.

[17] X. Wang, D. Fouhey, and A. Gupta, "Designing deep networks for surface normal estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 539–547.

[18] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[19] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3994–4003.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[21] T. Gebru, J. Hoffman, and L. Fei-Fei, "Fine-grained recognition in the wild: A multi-task domain adaptation approach," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1349–1358.

[22] I. Kokkinos, "Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6129–6138.

[23] Z. Zhang, Z. Cui, C. Xu, Z. Jie, X. Li, and J. Yang, "Joint task-recursive learning for rgb-d scene understanding," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[24] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text clas-

sification with multi-task learning," *arXiv preprint arXiv:1605.05101*, 2016.

[25] H. Zhang, L. Xiao, Y. Wang, and Y. Jin, "A generalized recurrent neural architecture for text classification with multi-task learning," *arXiv preprint arXiv:1707.02892*, 2017.

[26] L. Xiao, H. Zhang, W. Chen, Y. Wang, and Y. Jin, "Mcapsnet: Capsule network for text with multi-task learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4565–4574.

[27] J. Chen, X. Qiu, P. Liu, and X. Huang, "Meta multi-task learning for sequence modeling," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[28] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," *arXiv preprint arXiv:1901.11504*, 2019.

[29] M. Rei and A. Søgaard, "Jointly learning to label sentences and tokens," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6916–6923.

[30] S. Zhao, T. Liu, S. Zhao, and F. Wang, "A neural multi-task learning framework to jointly model medical named entity recognition and normalization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 817–824.

[31] S. Böck, "Event detection in musical audio," Ph.D. dissertation, PhD thesis. Johannes Kepler University Linz, Linz Austria, 2016.

[32] B. Jia, J. Lv, and D. Liu, "Deep learning-based automatic downbeat tracking: a brief review," *Multimedia Systems*, vol. 25, no. 6, pp. 617–638, 2019.

[33] J. Schroeter, K. Sidorov, and D. Marshall, "Weakly-supervised temporal localization via occurrence count learning," *arXiv preprint arXiv:1905.07293*, 2019.

[34] S. Durand, J. P. Bello, B. David, and G. Richard, "Robust downbeat tracking using an ensemble of convolutional networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 76–89, 2016.

[35] S. Adavanne, A. Politis, J. Nikunen, and T. Virtanen, "Sound event localization and detection of overlapping sources using convolutional recurrent neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 1, pp. 34–48, 2018.

[36] Q. Kong, Y. Xu, I. Sobieraj, W. Wang, and M. D. Plumbley, "Sound event detection and time–frequency segmentation from weakly labelled data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 777–787, 2019.

[37] T. N. T. Nguyen, D. L. Jones, R. Ranjan, S. Jayabalan, and W. S. Gan, "A two-step system for sound event localization and detection," *arXiv preprint arXiv:1911.11373*, 2019.

[38] Y. Mao, J. Tian, J. Han, and X. Ren, "Hierarchical text classification with reinforced label assignment," *arXiv preprint arXiv:1908.10419*, 2019.

[39] R. Aly, S. Remus, and C. Biemann, "Hierarchical multi-label classification of text with capsule networks," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, 2019, pp. 323–330.

[40] W. Huang, E. Chen, Q. Liu, Y. Chen, Z. Huang, Y. Liu, Z. Zhao, D. Zhang, and S. Wang, "Hierarchical multi-label text classification: An attention-based recurrent network approach," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1051–1060.

[41] X. Zhu and M. Bain, "B-cnn: branch convolutional neural network for hierarchical classification," *arXiv preprint arXiv:1709.09890*, 2017.

[42] Y. Wang, Q. Hu, P. Zhu, L. Li, B. Lu, J. M. Garibaldi, and X. Li, "Deep fuzzy tree for large-scale hierarchical visual classification," *IEEE Transactions on Fuzzy Systems*, 2019.

[43] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *International Conference on Machine Learning*, 2018, pp. 5075–5084.

[44] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[45] J. Wehrmann and R. C. Barros, "Bidirectional retrieval made simple," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7718–7726.

[46] J. Wehrmann, A. Mattjie, and R. C. Barros, "Order embeddings and character-level convolutions for multimodal alignment," *Pattern Recognition Letters*, vol. 102, pp. 15–22, 2018.

[47] "Gtzan speech and music dataset," http://opihi.cs.uvic.ca/sound/music_speech.tar.gz.

[48] "Muspeak speech and music detection dataset," http://mirg.city.ac.uk/datasets/muspeak/muspeak-mirex2015-detection-examples.zip.

[49] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[50] D. P. Kingma and J. B. Adam, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[51] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015, pp. 448–456.

[52] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines vinod nair," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010, pp. 807–814.