Tackling the Cocktail Fork Problem for Separation and Transcription of Real-World Soundtracks

Darius Petermann, *Graduate Student Member, IEEE*, Gordon Wichern, *Member, IEEE*, Aswin Shanmugam Subramanian, Zhong-Qiu Wang, and Jonathan Le Roux, *Senior Member, IEEE*

***Abstract*—Emulating the human ability to solve the cocktail party problem, i.e., focus on a source of interest in a complex acoustic scene, is a long standing goal of audio source separation research. In this paper, we focus on the cocktail fork problem, which takes a three-pronged approach to source separation by separating an audio mixture such as a movie soundtrack or podcast into the three broad categories of speech, music, and sound effects (SFX - understood to include ambient noise and natural sound events). We evaluate several deep learning-based source separation models on this task using simple objective measures such as signal-to-distortion ratio (SDR) as well as objective metrics that better correlate with human perception. Furthermore, we thoroughly evaluate how source separation can influence the downstream transcription asks of speech recognition for speech and audio tagging for music and SFX. We also investigate the task of activity detection on the three sources as a way to further improve source separation and transcription. While we observe that source separation improves transcription performance in comparison to the original soundtrack, performance is still sub-optimal due to artifacts introduced by the separation process. Therefore, we thoroughly investigate how remixing of the three separated source stems at various relative levels can reduce artifacts and consequently improve transcription performance. We find that remixing music and SFX interferences at a target SNR of 17.5 dB reduces speech recognition word error rate, and similar impact from remixing is observed for tagging music and SFX content.**

***Index Terms*—Audio source separation, remixing, speech, music, sound effects, soundtrack, speech recognition, audio tagging, sound event detection.**

I. Introduction

OVER the last decade and especially with the recent advent of data-driven approaches, many studies have been investigating the separation of audio sources found in media content; whether addressing the separation of speech from non-speech in speech enhancement [1], [2], speech from other speech in speech separation [3], [4], [5], individual musical instruments in music source separation [6], [7], [8] or non-speech sound events (or sound effects) in universal sound separation [9], [10], [11], [12], source separation finds its fit in many fields of application. Separating an audio mixture (e.g., movie soundtrack) into the three broad categories of speech, music, and sound effects (understood to include ambient noise and natural sound events) has however been left largely unexplored despite a wide domain of practical applications. A system properly trained on this task could indeed offer many potential benefits from the consumer standpoint, such as enhancing the listening experience by means of independent volume control over the sources (i.e., remixing), enabling the automatic captioning of sound events, or improving speech transcription accuracy, to name a few.

In our preliminary work [13], we formalized this new task of music-sound effects (SFX)-speech separation for real-world soundtracks as the cocktail fork problem (CFP), as illustrated in Fig. 1, and introduced a dataset specifically tailored towards this task, the Divide and Remaster (DnR) dataset, to foster research on this topic. We benchmarked multiple existing popular source separation models and proposed a multi-resolution short-time Fourier transform (STFT) architecture to better address the variety of acoustic characteristics of the three source types. We reported model performance in terms of scale-invariant signal-to-distortion ratio (SI-SDR) [14] and showcased the benefit of our proposed model, which produced SI-SDR improvements over the mixture of 11.0 dB for music, 11.2 dB for speech, and 10.8 dB for sound effects. We additionally included an analysis on system performance under different overlapping source conditions. Finally, we investigated separation performance at different sampling rates.

In this work, we explore the natural extension of the CFP towards transcription, and explore multiple techniques for integrating source separation and transcription such that the performance of both can be improved. The DnR dataset includes time-stamped annotations for all three source types, in the form of speech transcription for speech, music genre labels for music, and sound-event tags for SFX. Specifically, we investigate transcription for the CFP with the following novel contributions:

- We present an activity detection system for the three parent classes (music, speech, SFX). By first detecting boundaries, we can then use sophisticated speech recognition and

Manuscript received 13 December 2022; revised 25 April 2023; accepted 30 May 2023. Date of current version 13 July 2023. The work of Darius Petermann was supported by an intern at MERL. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Romain Serizel. *(Corresponding author: Darius Petermann.)*

Darius Petermann is with the Department of Intelligent Systems Engineering, Indiana University Bloomington, IN 47408 USA (e-mail: daripete@iu.edu).

Gordon Wichern and Jonathan Le Roux are with the Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: wichern@merl.com; leroux@merl.com).

Aswin Shanmugam Subramanian was with the Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA. He is now with the the Microsoft Corporation, One Microsoft Way, Redmond, WA 98052 USA (e-mail: asubra13@alumni.jh.edu).

Zhong-Qiu Wang is with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: wang.zhongqiu41@gmail.com).

Digital Object Identifier 10.1109/TASLP.2023.3290428

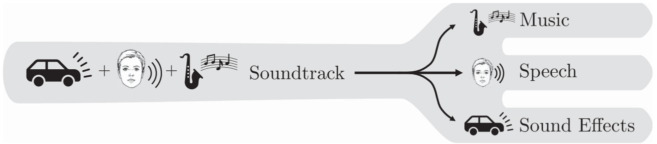


Fig. 1. Illustration of the cocktail fork problem: given a soundtrack consisting of an audio mixture of speech, music, and sound effects, the goal is to separate it into the three corresponding stems.

audio tagging models that expect pre-segmented chunks of audio as input.

- We showcase the benefit of integrating the output of our activity detection system with our source separation model by means of various conditioning mechanisms.
- We investigate the three classification tasks using well-established pre-trained models. For music genre and sound-event tagging, we evaluate YAMNet, a deep net that predicts 521 audio event classes from the AudioSet-YouTube corpus [15]. For speech transcription, we evaluate a conformer-based end-to-end automatic speech recognition (ASR) model provided by ESPnet [16]. The model is pre-trained on LibriSpeech [17], the same dataset DnR is based on.
- We explore the idea of source remixing, that is, the act of weighting and adding the separated sources back together, with the goal of increasing classification performance. We show that, in some cases, due to the imperfect nature of the separation, transcription can benefit from source remixing compared to using the raw separation output.
- In addition to SDR, we propose to evaluate our source separation models using two other metrics, that are arguably closer to human perception for audio quality assessment: PESQ [18] for speech and the 2f-model [19] for all three sources.

In this article, we aim at further investigating the CFP task, not only from the source separation angle as previously achieved, but from that of transcription as well. We hope this article serves as an important step to a system which would not only be capable of enhancing the listening experience but provide additional semantic understanding as well.

II. Related Work on the Interaction Between Separation and Downstream Tasks

The three broad categories of music, speech, and sound effects in audio signals can be found in many different types of settings, from podcasts to radio broadcasts, movies and TV-shows, their presence and overlap are ubiquitous. In more elementary scenarios, for example involving only one or two of the three classes, the task of source separation has been well investigated and has shown great promises towards various downstream tasks. For instance, separating individual instruments in musical signals has been found to be beneficial towards music transcription [20], source remixing in the context of music production [21], or even towards audio compression [22]. In the context of noisy speech, speech enhancement (or speech denoising) has shown great promise towards ASR tasks [23], [24]. More recent work proposed a joint audio-tagging and ASR system capable of fulfilling both tasks simultaneously [25]. Conversely, several works have considered how classification can improve performance in the context of general sound separation [10] or music separation [26], [27]. The interaction between voice activity detection and speech enhancement has also been explored [28], [29].

While the quality of source separation output has no doubt increased dramatically in the deep learning era, artifacts in the separated outputs remain a big problem. For speech enhancement in broadcast applications in particular, recent studies [30], [31] have shown that remixing the separated speech with some amount of the separated noise can substantially reduce artifacts and improve the listening experience compared to the noisy original. In this article, we explore how similar remixing ideas can benefit the downstream transcription tasks of ASR, sound event detection, and music genre classification.

Particularly in complex mixtures, where many sounds may interfere with the source of interest, transcription becomes a very challenging task. One way to address this obstacle and potentially improve on the classification output would be to somehow reduce the amount of interfering sources in the mixture and consequently benefit the transcription task. For speech, the idea of using speech enhancement as a front end for recognition has been widely explored [16], [32]. However, these front-end systems are not perfect and may lead to artifacts and unwanted residual noises in the clean speech estimates, which ultimately may negatively affect the downstream ASR task. In [33], the authors propose to overcome this limitation by introducing a mechanism controlling an optimal level of noise reduction for the ASR task. An approach for learning whether to use the enhanced speech or the noisy mixture for ASR was presented in [34], where the authors ultimately found a soft combination between the enhanced signal and the noisy mixture signal performed best. A related study [35] also demonstrated the benefit of remixing the enhanced speech and the noisy mixture for ASR. In this work, we not only evaluate remixing for ASR in the presence of difficult music and sound effect background signals, but also study the benefit of remixing separated signals for sound event tagging and music genre recognition.

In [36], the authors evaluated source separation as a pre-processing step to improve the sound-event detection (SED) task by first breaking down mixtures into their constituent sounds. SED was then applied by combining the separated sources and the input mixture at different stages in the architecture. Although the separator is not trained jointly in this case (and consequently no active remixing is performed during training), one could argue that the remixing may be done implicitly within the SED network. Source separation in music applications often focuses on remixing separated musical stems [37]. Separation has also been widely used for music transcription (i.e., scoring), either as a pre-processing front-end [38], [39] or within a joint approach [20], [40]. In [41], the authors extensively explore the idea of source separation specifically applied to choir ensemble mixtures, allowing for a set of potential downstream applications such as F_0 contour analysis, synthesis, transposition, unison analysis, as well as singing g roup remixing. In [42], the authors investigate existing source

TABLE I

Some Examples of Events Found in the DnR Metadata and How We Formulate Them in the Context of the CFP Transcription

Task	# classes	⟨ tags,	start_time,	end_time ⟩
Music	11	⟨ rock	2.3 s	19.5 s ⟩
SFX-Fg.	85	⟨ dog,bark,animal	10.1 s	13.8 s ⟩
SFX-Bg.	35	⟨ rain,thunder	1.0 s	30.6 s ⟩
Speech	5,000	⟨ ['the man walks ...']	5.9 s	24.2 s ⟩

"# Classes" indicates the vocabulary size for each transcription task (e.g., the number of tokens for ASR is 5,000).

separation algorithms and their perceptual impact on songs given various remixing scenarios. The claim suggests that existing separation approaches may suffer from imperfect separation, resulting in perceptible artifacts on individual source estimates, which can then jeopardize downstream task performance. To the best of our knowledge, there has not been any work investigating the task of music source remixing specifically targeted towards downstream labeling tasks such as genre recognition in a manner similar to what we explore in this work.

III. Methods

A. Problem Setup

In this work, we assume that we observe a single-channel mixture $x \in \mathbb{R}^T$ composed of three submixes:

$$x = y^{(s)} + y^{(m)} + y^{(e)}, \tag{1}$$

where $y^{(s)}$ is the submix containing all speech signals, $y^{(m)}$ that of all music signals, and $y^{(e)}$ that of all SFX. We use this latter term to broadly cover all sources not categorized as speech or music, and choose it over alternatives such as sound events or noise, as the term is especially relevant to our target application where x is a soundtrack. We here define the cocktail fork problem as that of recovering, from the audio soundtrack x, its music, speech, and sound effect submixes, as opposed to extracting individual musical instruments, speakers, or sound effects.

Additionally, we consider the case where the submixes have associated collections $l^{(s)}_{1:N_s}$, $l^{(m)}_{1:N_m}$, $l^{(e)}_{1:N_e}$ of metadata labels describing the content for each source type. Specifically, as illustrated by the examples in Table I , for speech we consider the speech recognition task where label $l^{(s)}_{i_s}$ represents the transcription of the i_s-th utterance and associated time boundaries, with index $i_s \in \{1, \ldots, N_s\}$, where N_s is the number of utterances in $y^{(s)}$. Similarly, $l^{(m)}_{i_m}$ represents a music genre label and associated time boundaries for the i_m-th music excerpt. For sound effects, $l^{(e)}_{i_e}$ represents a list of audio tags describing the i_e-th sound event, along with the associated time boundaries; these tags are further split into two sub-categories of foreground events (SFX-Fg) such as "dog barking" and background events (SFX-Bg) such as "traffic noise," which are treated separately for transcription.

In this work, we focus both on separation applications where the goal is to recover estimates $\hat{y}^{(s)}, \hat{y}^{(m)}, \hat{y}^{(e)}$ of the submixes, and on transcription applications where the goal is to estimate metadata label collections $\hat{l}^{(s)}_{1:N_s}$, $\hat{l}^{(m)}_{1:N_m}$, $\hat{l}^{(e)}_{1:N_e}$ given a mixture x and its associated source separation outputs $\hat{y}^{(s)}, \hat{y}^{(m)}, \hat{y}^{(e)}$.

In the same vein as some of the remixing approaches presented in Section II, we explore the idea of source remixing towards the three transcription downstream tasks. Our working hypothesis is that, while source separation likely helps the downstream networks to reach a better performance on their respective tasks, it is imperfect, with the presence of added interferences and artifacts, and adding back a *down-scaled* version of the predicted constituents in each of the target sources (e.g., music and SFX signals in the speech signal for ASR) prior to performing the transcription may help improve performance. More formally, this can be formulated as follows:

$$\tilde{y}^{(i)} = \tau_i^{(s)} \hat{y}^{(s)} + \tau_i^{(m)} \hat{y}^{(m)} + \tau_i^{(e)} \hat{y}^{(e)}, \quad i \in \{s, m, e\}, \tag{2}$$

where $\tilde{y}^{(s)}$, $\tilde{y}^{(m)}$, and $\tilde{y}^{(e)}$ denote the remixed separated sources, and $\tau_i^{(j)}$ denotes the time invariant gain applied to the separated estimate of source j to obtain the remixed source i, e.g., $\tau_m^{(s)}$ is the gain applied to the separated speech estimate $\hat{y}^{(s)}$ when remixing for music transcription. Taking the example of speech transcription, the gain $\tau_s^{(s)}$ applied to the speech estimate $\hat{y}^{(s)}$ is set to always remain at unit level, while the gains $\tau_s^{(m)}$ and $\tau_s^{(e)}$ of the interfering sources $\hat{y}^{(m)}$ and $\hat{y}^{(e)}$, respectively, are adjusted (either individually or jointly) to match a target SNR. In the individual case, the gains are set to

$$\tau_s^{(j)} = \frac{\|\hat{y}^{(s)}\|_2}{\|\hat{y}^{(j)}\|_2} 10^{-\mathrm{snr}_s^{(j)}/20}, \quad j \in \{m, e\}, \tag{3}$$

where $\mathrm{snr}_s^{(m)}$ is the desired SNR of the rescaled speech signal with respect to the rescaled music signal, and $\mathrm{snr}_s^{(e)}$ is defined similarly with respect to the rescaled sound effect signal. Alternatively, we can adjust the gain to reach a desired SNR $\mathrm{snr}_s^{(m+e)}$ of the speech signal with respect to the sum of the two interfering source signals $\hat{y}^{(m+e)} = \hat{y}^{(m)} + \hat{y}^{(e)}$, i.e.,

$$\tau_s^{(m+e)} = \frac{\|\hat{y}^{(s)}\|_2}{\|\hat{y}^{(m+e)}\|_2} 10^{-\mathrm{snr}_s^{(m+e)}/20}. \tag{4}$$

Note that, in all cases, we assume that the power of each separated source signal $\hat{y}^{(j)}$ is roughly equal to the power of the corresponding ground-truth source signal $y^{(j)}$, which holds in practice if the separation quality is reasonable. An overview of our proposed approach is shown in Fig. 2, and we now describe all stages in detail.

B. Source Separation With MRX

In our preliminary work [13], we found that time-frequency (TF) separation models generally worked well, and we observed additional benefit from jointly evaluating multiple TF resolutions to better handle the diverse acoustic characteristics present in mixtures of speech, music, and sound effects. Therefore, in this work, we use the multi-resolution crossnet (MRX) introduced in [13] and shown in Fig. 3 as our main network architecture. MRX takes a time-domain input mixture x and encodes it into I complex spectrograms $X_{W_i} = \mathrm{STFT}_{W_i}(x)$, $i \in \{1, \ldots, I\}$ with different STFT resolutions, where W_i denotes the i-th window length in milliseconds. Fig. 3 shows an example with $I = 3$ and $\{W_1, W_2, W_3\} = \{32, 64, 256\}$.

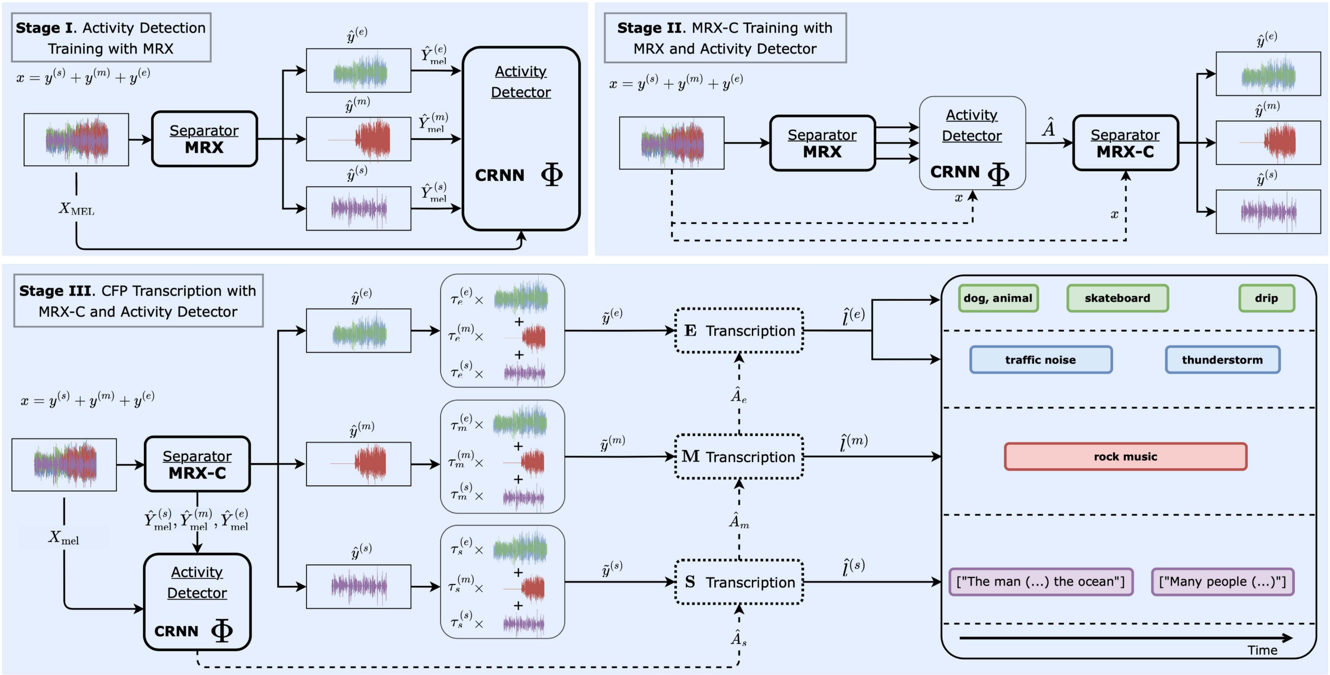

Stage I. Activity Detection Training with MRX
$x = y^{(s)} + y^{(m)} + y^{(e)}$
Separator MRX
X_{MEL}
$\hat{y}^{(e)}$
$\hat{y}^{(m)}$
$\hat{y}^{(s)}$
$\hat{Y}^{(e)}_{\mathrm{mel}}$
$\hat{Y}^{(m)}_{\mathrm{mel}}$
$\hat{Y}^{(s)}_{\mathrm{mel}}$
Activity Detector
CRNN Φ
Stage II. MRX-C Training with MRX and Activity Detector
$x = y^{(s)} + y^{(m)} + y^{(e)}$
Separator MRX
Activity Detector
CRNN Φ
$\hat{A}$
Separator MRX-C
x
x
$\hat{y}^{(e)}$
$\hat{y}^{(m)}$
$\hat{y}^{(s)}$
Stage III. CFP Transcription with MRX-C and Activity Detector
$x = y^{(s)} + y^{(m)} + y^{(e)}$
Separator MRX-C
X_{mel}
$\hat{Y}^{(s)}_{\mathrm{mel}}, \hat{Y}^{(m)}_{\mathrm{mel}}, \hat{Y}^{(e)}_{\mathrm{mel}}$
Activity Detector
CRNN Φ
$\hat{y}^{(e)}$
$\hat{y}^{(m)}$
$\hat{y}^{(s)}$
$\tau^{(e)}_{e}\times$ + $\tau^{(m)}_{e}\times$ + $\tau^{(s)}_{e}\times$
$\tau^{(e)}_{m}\times$ + $\tau^{(m)}_{m}\times$ + $\tau^{(s)}_{m}\times$
$\tau^{(e)}_{s}\times$ + $\tau^{(m)}_{s}\times$ + $\tau^{(s)}_{s}\times$
$\tilde{y}^{(e)}$
$\tilde{y}^{(m)}$
$\tilde{y}^{(s)}$
E Transcription
M Transcription
S Transcription
$\hat{A}_e$
$\hat{A}_m$
$\hat{A}_s$
$\hat{l}^{(e)}$
$\hat{l}^{(m)}$
$\hat{l}^{(s)}$
dog, animal
skateboard
drip
traffic noise
thunderstorm
rock music
["The man (...) the ocean"]
["Many people (...)"]
Time

Fig. 2. Overall separation, activity detection, and transcription pipeline. In **Stage I**, the activity detector Φ_{EI} is trained to provide frame-level activity labels for the three CFP classes (music, speech, SFX). In **Stage II**, MRX-C, a label-informed version of MRX, is trained using the activity label provided by Φ_{EI}. Lastly, in **Stage III**, we proceed to total transcription for a given CFP mixture by first estimating its sources and frame-level labels. The transcription for the three sources is then achieved on the source estimates and using the frame labels.

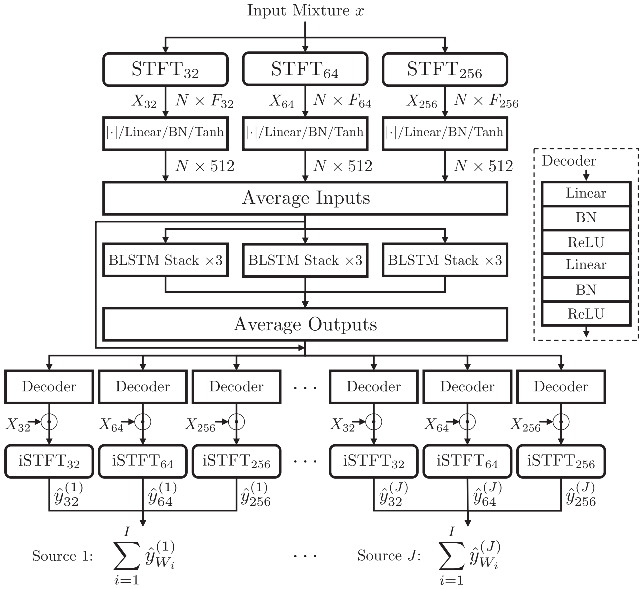


Fig. 3. Multi-resolution CrossNet (MRX) architecture.

We use the same hop size (e.g., 8 ms in the example of Fig. 3) for all resolutions, so they remain synchronized in time, and N denotes the number of STFT frames for all resolutions. In practice, we set the window size in samples to the nearest power of 2, and the number of unique frequency bins is denoted as F_{W_i}. Each resolution is then passed to a fully connected block to convert the magnitude spectrograms of dimension $N \times F_{W_i}$ into a consistent dimension of 512 across the resolution branches. This allows us to average them together prior to the bidirectional long short-term memory (BLSTM) stacks, whose outputs are averaged once again. MRX was inspired by the Cross-Unmix (XUMX) architecture proposed in [43]. However, in our case, the input averaging is intended to allow the network to efficiently combine inputs with multiple resolutions.

The average inputs and outputs of the BLSTM stacks are concatenated and decoded back into magnitude soft masks $\hat{M}^{(j)}_{W_i}$, one for each of the three sources $j \in \{s, m, e\}$ and each of the I original input resolutions W_i. The decoder consists of two stacks of fully-connected layers, each followed by batch normalization (BN) and rectified linear units (ReLU). For a given source j, each magnitude mask $\hat{M}^{(j)}_{W_i}$ is multiplied element-wise with the original complex mixture spectrogram X_{W_i} for the corresponding resolution, a corresponding time-domain signal $\hat{y}^{(j)}_{W_i}$ is obtained via inverse STFT, and the estimated time-domain signal $\hat{y}^{(j)}$ is obtained by summing the time-domain signals at each resolution:

$$\hat{y}^{(j)} = \sum_{i=1}^{I} \hat{y}^{(j)}_{W_i} = \sum_{i=1}^{I} \mathrm{iSTFT}\left(\hat{M}^{(j)}_{W_i} \odot X_{W_i}\right). \tag{5}$$

For the cocktail fork problem, the network has to estimate a total of $3I$ masks. Since ReLU is used as the final mask decoder nonlinearity, the network can freely learn weights for each resolution that best reconstruct the time-domain signal. We use the SI-SDR loss function [14], [44] between the estimated signal $\hat{y}^{(j)}$ and the ground-truth signal $y^{(j)}$ for $j \in \{s, m, e\}$.

We now investigate how source separation and activity detection may benefit from each other.

C. Stage I: Using Source Separation to Improve Activity Detection

As a first step towards transcribing the separated speech, music, and sound effects stems, we focus on detecting the temporal regions where each source is active. Only these detected temporal regions for each source are then fed to the appropriate downstream classifiers, namely, ASR for speech, genre recognition for music, and audio tagging for sound effects. End-to-end (E2E) ASR systems are typically sensitive to extended silent portions occurring between utterances as well as extensive utterance length, and may not work optimally when fed unsegmented signals. Similarly, for music and sound effects, we can pool over temporal regions to obtain more accurate tagging performance.

Formally, we define $A = (a_{j,n})_{j,n} \in \{0,1\}^{3\times N}$ as the ground-truth activity labels, with elements $a_{j,n} = 1$ indicating that source $j \in \{s, m, e\}$ is active at frame n, and $a_{j,n} = 0$ that it is inactive. Note that the ground-truth activity labels are defined by the presence of an active excerpt at frame n as defined in the metadata, rather than its energy, so pauses in a speech utterance or a song may still be considered active. Our goal is to obtain estimated activity labels $\hat{A} = (\hat{a}_{j,n})_{j,n} \in \mathbb{R}^{3\times N}$, using a neural network Φ applied to an input representation R derived from the mixture of interest:

$$\hat{A} = \Phi(R). \tag{6}$$

The network is trained to minimize the binary cross-entropy between A and $\hat{A}$ over the training set:

$$\begin{aligned}\mathcal{L}(A,\hat{A}) &= \sum_{j,n} \mathrm{BCE}(a_{j,n}, \hat{a}_{j,n}) \\ &= \sum_{j,n} \left(-a_{j,n}\log(\hat{a}_{j,n}) - (1-a_{j,n})\log(1-\hat{a}_{j,n})\right).\end{aligned} \tag{7}$$

At inference time, we apply a median filter over time and threshold $\hat{A}$ to determine boundaries. We denote by $\hat{A}_j$ the estimated activities for source j.

For the core architecture of the neural network Φ, we use a convolutional recurrent neural network (CRNN), as such architectures have proven to be highly effective in the context of sound event detection [11], [45], [46], with mel spectrograms as the input representation. The estimated activity labels $\hat{a}_{j,n}$ are in the range [0,1] as they are the outputs of a sigmoid activation function. Our baseline Φ_{B} simply takes the mel spectrogram X_{mel} of the mixture as input, and we denote its output as $\hat{A}^{(\mathrm{B,mix})} = \Phi_{\mathrm{B}}(X_{\mathrm{mel}})$.

In [36], multiple source-separation-based algorithms are proposed to improve sound event detection. We explore such approaches for the CFP as shown in Stage I of Fig. 2. In the first approach, **early integration** (EI), the input to a CRNN Φ_{EI} is formed by stacking the mel spectrograms of the mixture and estimated sources along the channel dimension, i.e.,

$$\hat{A}^{(\mathrm{EI})} = \Phi_{\mathrm{EI}}\left(\mathrm{stack}\left(X_{\mathrm{mel}}, \hat{Y}^{(s)}_{\mathrm{mel}}, \hat{Y}^{(m)}_{\mathrm{mel}}, \hat{Y}^{(e)}_{\mathrm{mel}}\right)\right). \tag{8}$$

The input is thus of shape $(4 \times N \times F)$ where F is the number of mel bands, and N the number of time frames. In **middle integration** (MI), X_{mel} and the $\hat{Y}^{(j)}_{\mathrm{mel}}$ are each individually input into the CNN block of a CRNN Φ_{MI}, and their outputs are stacked before being fed to the RNN block. Both CRNNs Φ_{EI} and Φ_{MI} are trained using Eq. (7), with the source separation network kept frozen. Finally, in **late integration** (LI), the baseline CRNN Φ_{B} is used to process independently the mel spectrograms X_{mel} of the mixture and $\hat{Y}^{(j)}_{\mathrm{mel}}$ of each estimated source to obtain the corresponding output probabilities $\hat{A}^{(\mathrm{B,mix})}$ and $\hat{A}^{(\mathrm{B},j)} = \Phi_{\mathrm{B}}(\hat{Y}^{(j)}_{\mathrm{mel}})$ (for example, $\hat{A}^{(\mathrm{B},s)} \in \mathbb{R}^{3\times N}$ denotes the estimated activity labels of all three sources within the separated speech estimate). These probabilities are then combined to obtain the late integration estimates $\hat{A}^{(\mathrm{LI})}$ as:

$$\hat{A}^{(\mathrm{LI})} = \frac{1}{2}\hat{A}^{(\mathrm{B,mix})} + \frac{1}{2}\left(\frac{\hat{A}^{(\mathrm{B},s)} + \hat{A}^{(\mathrm{B},m)} + \hat{A}^{(\mathrm{B},e)}}{3}\right). \tag{9}$$

In this work, all three fusion approaches will be compared in Section V. We found that the early integration approach was the one that worked best for the CFP activity detection. We refer the interested reader to the original work on the topic [36] for further technical details.

D. Stage II: MRX-C – Using Activity Detection to Improve Source Separation

As shown in Stage II of Fig. 2, we also explore how source separation can benefit from activity detection, by conditioning our MRX separation models on the activity detection output. In our conditioning approach, which we call MRX-C, we concatenate class probabilities $\hat{A}$ obtained at the output of one of the configurations of Section III-C with the MRX input mixture STFT at each resolution shown in Fig. 3. Beside concatenation, we also experimented with a FiLM [47] conditioning approach, which consisted in using a FiLM layer placed after each of the MRX STFT operations in Fig. 3. This approach led to very similar results to the concatenation counterpart in preliminary experiments, we thus opted to use the concatenation for all further experiments. Because the frame rate of the CRNN outputs may be lower than the STFT frame rate of MRX due to temporal pooling operations, we use nearest neighbor upsampling to match the frame rates.

We further explore the impact of using the ground-truth class labels during the training and inference stages as the upper bound oracle performance. We also explored running two consecutive iterations, or passes, of Stage I and Stage II of Fig. 2, which we denote MRX-C$_{\mathrm{2p}}$, where the suffix "2p" refers to "2 passes".

As a comparison to the approaches combining activity detection and separation explored in this section, we also consider a multi-task learning approach that jointly does activity detection and source separation [27], [29], referred to as MRX-MTL. Here, an additional decoder layer with three sigmoid outputs is added in parallel to the separation decoders in Fig. 3 to estimate activity detection. A time average pooling layer with a factor of eight is applied at the input of this decoder to be consistent with the

CRNN output resolution, and a weighted binary cross-entropy loss is added to the separation loss for training.

E. Stage III: CFP Transcription

In this section, we focus on downstream transcription tasks using the remixed source separation outputs and source activity boundaries. Specifically, we investigate audio tagging for music (in the form of music genre recognition) and sound effects, and ASR for speech.

1) Audio Tagging: To transcribe music and sound effects, we make use of the powerful pre-trained audio tagging model YAMNet[1], which predicts 521 audio event tags, including multiple music genres and sound effects classes. YAMNet is based on the MobileNet convolutional architecture [48], and has been trained on the AudioSet Ontology [49], a human-labeled corpus derived from short Youtube audio segments. YAMNet operates on 960 ms frames with 50% overlap, and outputs a vector of 521 class activity probabilities for each frame. The output layer uses sigmoid activation functions, so multiple classes (i.e., tags) can be active for each frame. In practice, we input the entire soundtrack (either the mixture, separated sources, or remixes) into YAMNet, and then average the class probabilities over segments estimated by the activity detector. Example annotations from a soundtrack are shown in Table I.

For music, we typically do not expect there to be multiple pieces of music playing simultaneously, so we limit the estimated tag to a single one describing the predominant music genre. As we are considering music genre classification, we only take into account the YAMNET outputs corresponding to music genre classes.

In real-world soundtracks, sound effects typically serve two main purposes: *background events* that usually entail longer and lower amplitude sounds that help set the scene (e.g., rain, traffic), and *foreground events* which are shorter and louder to help tell the story (e.g., gun shot, footsteps). For this reason, we further sub-divide the sound effects audio tagging task into foreground (SFX-Fg) and background (SFX-Bg) sub-tasks as illustrated in Table I. Unlike for music genre, we allow sound events to be labeled with multiple tags from the AudioSet ontology (e.g., dog, bark, animal). Furthermore, sound events from both SFX-Bg and SFX-Fg can overlap in time as shown in the example of Table I. While our goal is to separate and annotate a single sound effects stem, because of the widely different characteristics of SFX-Fg and SFX-Bg sound events, we consider them separately when creating synthetic mixtures in Section IV-A and when evaluating audio tagging performance in Section V-B. However, for evaluating source separation and activity detection performance in Section V-A, we consider SFX-Fg and SFX-Bg jointly as a single sound effects class.

2) Speech Transcription: We evaluate ASR using a model pre-trained on the LibriSpeech corpus. Specifically, we use a state-of-the-art end-to-end (E2E) model implemented in ESPnet [16] that is based on the Conformer [50] architecture and uses HuBERT [51] input features. In general, a soundtrack contains multiple speech utterances interspersed among non-speech regions, which may cause difficulty when feeding the unsegmented soundtrack (be it the mixture, separated sources, or remixes) into an ASR model. Therefore, we evaluate the E2E ASR model using the approach shown in Fig. 4, where the activity detector Φ_{EI} first retrieves individual utterances which can then be individually input to the E2E ASR model. The individual utterance hypotheses from the E2E ASR model are then concatenated and evaluated against the full-file reference.

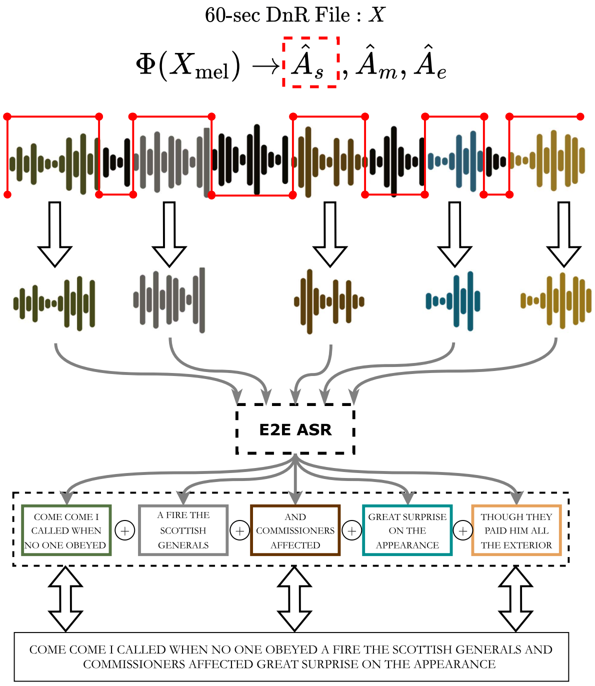

60-sec DnR File : X
$\Phi(X_{\text{mel}}) \rightarrow \hat{A}_s, \hat{A}_m, \hat{A}_e$
E2E ASR
COME COME I CALLED WHEN NO ONE OBEYED
A FIRE THE SCOTTISH GENERALS
AND COMMISSIONERS AFFECTED
GREAT SURPRISE ON THE APPEARANCE
THOUGH THEY PAID HIM ALL THE EXTERIOR
COME COME I CALLED WHEN NO ONE OBEYED A FIRE THE SCOTTISH GENERALS AND COMMISSIONERS AFFECTED GREAT SURPRISE ON THE APPEARANCE

Fig. 4. Illustration of our ASR evaluation pipeline using ESPnet. We first segment the full 60-second utterance into smaller chunks using Φ's output. Each of the resulting sub-utterances are then individually fed to ESPnet for decoding. Finally their hypotheses are concatenated and compared against the full-utterance reference.

IV. Experimental Setup

A. Divide and Remaster (DnR) Dataset

The DnR dataset was introduced in our preliminary work [13] with the goal of creating synthetic monophonic soundtracks for training and evaluating source separation algorithms. The dataset is publicly available along with the data creation scripts[2], and further details on the creation process are available in [13]. DnR is created by mixing speech from LibriSpeech [17], music from the Free Music Archive (FMA) [52], and sound effects from the Freesound Dataset 50 k (FSD50 K) [53]. All of the DnR building blocks contain audio at sampling rates of 44.1 kHz or greater (by default LibriSpeech is available at 16 kHz, but the 44.1 kHz mp3 files are also available), so we use 44.1 kHz as the default sampling rate for DnR. While the LibriSpeech, FMA, and FSD50 K files may have compression artifacts as a

[1] [Online]. Available: https://github.com/tensorflow/models/tree/master/research/audioset/yamnet

[2] [Online]. Available: https://cocktail-fork.github.io

TABLE II
DNR DATASET PROPERTIES

Source	{M,S,X}	{∅,S,X}	{M,∅,X}	{M,S,∅}	{M,∅,∅}	{∅,S,∅}	{∅,∅,X}
Frames	31910	6351	8702	3761	1330	1761	2737

	Music	Speech	SFX-FG	SFX-BG
λ	7	8	12	6
Target LUFS	-24.0	-17.0	-21.0	-29.0

The top table denotes the distribution of the seven overlapping USE-CASES found in the dataset. M, S, X indicate presence of Music, Speech, and SFX respectively, within a 1 second non-overlapping frame in the test set. The bottom table shows the class density (λ) and LUFS parameters used in the DNR creation procedure.

result of being stored in a lossy format (e.g., mp3), the 44.1 kHz sample rate enables real-world listening applications, and can easily be downsampled for transcription where high bandwidth is unnecessary. For FSD50 k, we manually classify each of the 200 class labels into one of 3 groups: foreground sounds (e.g., dog bark), background sounds (e.g., traffic noise), and speech/musical instruments (e.g., guitar, speech). Speech and musical instrument clips are filtered out to avoid confusion with our speech and music datasets, and we use different mixing rules for foreground and background events.

Many synthetic mixing pipelines for source separation such as wsj0-2mix [3] or speech/music/noise separation [54] create fully-overlapped mixtures, with arbitrarily selected source levels. However, models trained on these datasets may not robustly transfer to real-world situations without full-overlap [55]. Therefore, we took great care in DnR to make mixtures that are sufficiently realistic in terms of source overlap and relative amplitude level between the three classes. In order to ensure that a mixture could contain multiple full speech utterances and feature a sufficient number of onsets and offsets between the different classes, we decided to make each mixture 60 seconds long. We do not allow within-class overlap between speech and music clips, i.e., two music files (or two speech files) will not overlap, but foreground and background sound effects can overlap. This mixing procedure leads to, approximately, 55% of the DnR test set frames having speech, music, and sound effects active, 32% containing two of the three, 10% containing one of the three, and 3% silent frames (Table II).

Regarding the relative amplitude levels across the three classes, after analyzing studies such as [56] and informal mixing rules from industries such as motion pictures, video games, and podcasting, we follow an approach where speech is generally found at the forefront of the mix, followed by foreground sound effects, then music, and finally background sound effects. Table II shows some statistical properties of the DnR dataset. The top table denotes the total number of frames pertaining to each of the seven possible overlapping categories. The bottom table provides the class-specific file density and level in loudness units full-scale (LUFS) [57]. The number of files for each class is sampled from a zero-truncated Poisson distribution with expected value λ. The values of λ are chosen based on the average file length of each class, e.g., music and background effects tend to be longer. To add variability while keeping a realistic consistency over an entire mixture, we first sample an average LUFS value for each class in each mixture, uniformly from a range of ± 2.0 around the corresponding Target LUFS. Then each sound file added to the mix has its individual gain further adjusted by uniformly sampling from a range of ± 1.0. DnR contains 3,406 60-second mixtures for training, 487 for validation, and 973 for testing.

Along with the audio, DnR also contains transcription metadata in a format similar to the example of Table I. The start and end times correspond to where in the 60 s mixture the original clip was inserted. The gain applied to the clip is also available from the metadata, but is not shown in the example of Table I. For speech utterances, we use the unaltered transcription of the sentence from the LibriSpeech metadata. For music, we list all genres from the FMA annotations, but we use only the top-level genre (corresponding to 16 commonly used labels such as "jazz" or "rock") in our genre recognition experiments. For sound effects, we list all tags from the FSD50 K metadata, and also note whether the clip is used as a foreground or background event.

B. Source Separation

1) XUMX and MRX Models: We consider single-resolution XUMX baselines with various STFT resolutions. We opt to cover a wide range of window lengths $W \in \{32, 64, 128, 256\}$ to assess the impact of resolution on performance. For our proposed MRX model, we use three STFT resolutions of 32, 64, and 256 ms, which we found to work best on the validation set. We use XUMX_W to denote a model with a W ms window. We set the hop size to a quarter of the window size. For the MRX model, we determine hop size based on the shortest window. To parse the contributions of the multi-resolution and multi-decoder features of MRX, we also evaluate an architecture adding MRX's multi-decoder to the best single-resolution model (XUMX_{64}), referred to as $\text{XUMX}_{64,\text{multi-dec}}$. This results in an architecture of the same size (i.e., same number of parameters) as our proposed MRX model. In all architectures, each BLSTM layer has 256 hidden units and input/output dimension of 512, and the hidden layer in the decoder has dimension 512. For all MRX models, we use a sampling rate of 44.1 kHz.

2) Other Benchmarks: We also evaluate our own implementations of Conv-TasNet [44] and a temporal convolution network (TCN) with mask inference (MaskTCN). The sampling rate is again 44.1 kHz for all models. MaskTCN uses an identical TCN to the one used internally by Conv-TasNet, but the learned encoder and decoder are replaced with STFT and iSTFT operations. For MaskTCN, we use an STFT window/hop of 64/16 ms, and for the learned encoder/decoder of Conv-TasNet, we use 500 filters with a window size of 80 samples and a stride of 40. All TCN parameters in both Conv-TasNet and MaskTCN follow the best configuration of [44]. Additionally, we evaluate Open-Unmix (UMX) [7], the predecessor to XUMX, by training a separate model for each source, but without the parallel branches and averaging operations introduced by XUMX. The Conv-TasNet, UMX, XUMX, and MRX models all use SI-SDR [14], [44] as loss function, while MaskTCN uses the waveform domain L_1 loss.

All models are trained on 9 s chunks, except MaskTCN, trained on 6 s chunks, and Conv-TasNet, trained on 2 s chunks; we found these values to lead to best performance under our GPU memory constraints. All models are trained for 300 epochs using ADAM. The learning rate is initialized to 10^{-3}, and halved if the validation loss is not improved over 3 epochs.

C. Activity Detection

The activity detection CRNN models described in Section III-C follow the DCASE 2020 Task 4 baseline architecture and its extensions as described in [36], based on the publicly available implementation[3]. We use the ADAM optimizer with a learning rate of 10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We train the systems for 80 epochs and select the weights returning the lowest loss score on the validation set. Prior to training, the data is preprocessed in the following manner. First, we convert all the 60-second long input mixtures into mel spectrograms with 64 bands, using an FFT size of 2048 and hop-length of 512. During training, we randomly sample 10-second long chunks, which translate to 864 temporal frames, from the resulting features. Note that due to the CRNN pooling operations, the number of frames is reduced by a factor of 8 in contrast to the initial input temporal dimension, reducing the number of temporal frames from 864 at the input down to 108 at the output. The activity predictions are obtained by applying a median filter of size 5 over time and thresholding $\hat{A}$ at 0.5.

D. MRX-C Models

The conditioned MRX-C models described in Section III-D follow the same training pipeline as our MRX model described above. For the conditioning portion, the activity labels are obtained for the entirety of the DnR dataset prior to training by using our best performing activity detection model (later described in Section V-A), the early-integration model Φ_{EI}. During training and inference, the labels are upsampled to the length of the input mixture spectrogram using the nearest neighbor algorithm.

The multi-task learning model MRX-MTL is optimized on a compound objective function consisting of the SI-SDR and binary cross entropy losses, with their respective weights being 1.0 and 10.0.

E. Transcription

1) Music Genre Recognition: As YAMNet was originally trained on 16 kHz audio data, all audio involved in our YAMNet experiments is first resampled from 44.1 kHz down to 16 kHz. The audio signal goes through a set of transforms prior to network input. First the magnitude spectrogram is obtained through the STFT transform using a window size of 25 ms and hop size of 10 ms, the spectrogram is then mapped to 64 mel bins covering a 125-7500 Hz range in order to obtain its mel transform. Lastly, a stabilized log operation is applied to it. YAMNet uses 960 ms frames overlapped at 50%. Since the DnR mixtures are 60 s long, we obtain 124 YAMNet frames for each DnR mixture.

One challenge encountered when using YAMNet for the music classification task was the mapping from the 16 top-level genre labels used in the FMA dataset, to one of the 521 sound-event classes from the Audioset ontology used by YAMNet. Ten of the genres had straightforward mapping between FMA and Audioset, namely Pop, Rock, Soul-RnB, Jazz, Country, Electronic, Blues, Hip-hop, Folk, and Classical. The “International” genre used by FMA could be mapped to multiple Audioset genres (e.g., “Music of Asia”, “Music of Africa”), however, this one-to-many mapping provided very poor performance. The YAMNet “Vocal Music” category consistently yielded high confidence when fed music labeled as “International” by FMA, as these audio files typically contained singing with little-to-no instrumental parts. Therefore, we mapped “International” to “Vocal Music”. There are 5 FMA genres that we excluded from the classification task as they were overly broad and had no appropriate Audioset genre counterpart (Experimental, Easy Listening, Instrumental, Spoken, and Old-time/historic). Any clips containing these 5 genres were still used for evaluation of music separation and activity detection, but they were ignored when evaluating music genre recognition performance using YAMNet.

2) Sound Event Detection: To filter music and speech clips from FSD50 K during the DnR data creation process, we removed a clip if the majority of its tags were related to speech or music. The size of the FSD50 K tag vocabulary after this filtering was reduced from 200 to 146. Furthermore, for YAMNet compatibility, we had to exclude 26 additional tags for the followings reasons:

- The class is an actual parent class in AudioSet, therefore not covered in YAMNet output (e.g., “Domestic_sounds_and_home_sounds”).
- The class does not have any examples in AudioSet, therefore it is not covered in YAMNet output (e.g., “Gull_and_seagull”).
- The tag contains a music or speech label (e.g., “Male_speech_and_man_speaking”), even though a majority of its tags are not related to music and speech.

We then split the remaining 120 tags into SFX-Fg (85 tags) and SFX-Bg (35 tags). We treat the classification tasks for SFX-Fg and SFX-Bg separately, as we expect relatively poor performance for SFX-Bg due to its low relative level and long duration, which likely overlaps with multiple foreground events in DnR.

When using YAMNet outputs for the music genre, SFX-Fg, and SFX-Bg tasks, we first filter the 521 class probabilities output by YAMNet at each time frame to only contain those relevant for the given task (i.e., 11 classes for music, 85 for SFX-Fg, and 35 for SFX-Bg). In practice, we would use the boundaries provided by the activity detector to average the class probabilities across all relevant frames for a given segment, and then output any tags with a class probability above a threshold. However, in our experiments, whose results are described in Section V-B, we aim to evaluate how source separation and remixing can aid soundtrack transcription without having activity detection performance play an out-sized role. We therefore use oracle event boundaries for averaging YAMNet probabilities. This also

[3] [Online]. Available: https://github.com/turpaultn/dcase20_task4

allows us to use threshold independent metrics such as mean average (mAP) precision and area under the ROC curve (AUC), without having to account for missed detections and false alarms. Furthermore, it enables evaluation of the SFX-Fg and SFX-Bg tasks separately, given that our activity detector only outputs overall sound effect boundaries.

3) Automatic Speech Recognition: All audio is downsampled to 16 kHz, prior to being input to the HuBERT [51] feature extraction front-end of the pre-trained Conformer-based [50] ESPnet model.[4] As we will demonstrate in Section V-B, inputting an entire unsegmented 60 s DnR file, which contains multiple utterances interspersed with noise/silence regions, leads to highly sub-optimal ASR performance. Therefore, using the activity detector to first segment speech regions, and then passing each segment to the ASR model before concatenating all the outputs (as previously discussed and illustrated in Fig. 4) was essential to obtaining acceptable ASR performance. Since we evaluate the concatenated transcriptions with the ground-truth transcription from the entire soundtrack, we can easily compare performance using estimated activity detection boundaries and oracle boundaries in terms of word error rate (WER) and character error rate (CER) using the noisy mixture, separated speech stem, or remixed speech stem.

V. Experimental Results

A. CFP Separation and Activity Detection

In this section, we evaluate source separation performance in terms of signal reconstruction metrics for listening applications, as well as the interaction between source separation and sound activity detection.

SI-SDR [14] is perhaps the most widely used objective measure for deep learning-based source separation, and was used in our preliminary study [13]. However, as shown in [59], SI-SDR is not among the objective metrics most correlated with perceptual quality. Therefore, in this work, we also report results using the 2f-model [19], which combines two mid-level perceptual features from the Perceptual Evaluation of Audio Quality (PEAQ) [60] standard (we used the PEAQ implementation from [61]). The 2f-model was fit to output estimates of MUSHRA scores ranging from 0 to 100, and sound signals were upsampled from 44.1 kHz to 48 kHz for input into the PEAQ model. To avoid any influence due to the scaling of the output, which is particularly an issue for models trained with SI-SDR loss, we normalize all outputs to have the same LUFS as their corresponding ground truth. For the speech source, we also report wideband Perceptual Evaluation of Speech Quality (PESQ) [62] using the *python-pesq* toolkit,[5] with signals downsampled to 16 kHz.

Table III presents the SI-SDR, PESQ, and 2f values of various models trained and tested on DnR, in addition to the “No Processing” condition (lower bound, using the mixture as estimate) and oracle phase sensitive mask [58] (a form of upper bound). For each model, SI-SDR improvements are fairly consistent across source types, despite the differences in their relative levels in the mix, which can be seen in the “No Processing” SI-SDR. In general, we observe that our proposed multi-resolution (MRX) models outperform all single-resolution baselines on all source types in terms of SI-SDR, PESQ, and the PEAQ-based 2f-model scores. This implies that the network learns to effectively combine information from different STFT resolutions to more accurately reconstruct the target sources. The performance of $\text{XUMX}_{64,\text{multi-dec}}$ further confirms this hypothesis by performing nearly identically in comparison to XUMX_{64}, showing that the use of multiple decoders alone does not improve performance. We also observe that the single-source models (UMX) tend to perform comparably to the cross-source models (XUMX, MRX) for speech, but perform worse for music and SFX. We speculate that because music and SFX are quieter in the mix, it is harder for the network to isolate them effectively without the support of the other sources, while louder sources (here, speech) do not benefit from joint estimation.

TABLE III
SI-SDR [dB] Results of Baselines and Proposed Models on DnR

	Music		SFX		Speech		
Model	SI-SDR	2f-m.	SI-SDR	2f-m.	SI-SDR	2f-m.	PESQ
No processing	-6.8	10.3	-5.0	10.9	1.0	6.3	1.18
Oracle PSF [58]	11.6	48.3	13.7	47.0	17.8	52.1	3.68
Conv-TasNet [44]	0.3	14.2	2.0	13.5	8.5	19.8	1.49
MaskTCN [44]	1.7	20.5	3.8	21.8	9.7	27.6	1.63
UMX_{64} [7]	3.1	22.2	4.4	21.5	11.7	30.9	1.76
XUMX_{32} [43]	2.9	21.6	4.7	21.7	11.2	29.9	1.70
XUMX_{64} [43]	3.5	22.7	5.1	22.4	11.7	30.6	1.75
XUMX_{128} [43]	3.7	24.0	5.1	23.2	11.6	30.7	1.75
XUMX_{256} [43]	2.9	22.6	4.4	22.0	10.5	29.7	1.68
$\text{XUMX}_{64,\text{multi-dec}}$	3.5	23.6	5.0	22.5	11.8	31.3	1.82
MRX (proposed)	4.2	25.6	5.7	24.7	12.3	32.9	1.97
MRX-C (proposed)	**4.6**	**26.9**	**6.1**	**26.1**	**12.6**	**33.4**	**2.01**

MUSHRA score predictions from the 2F-model are denoted as 2F-M. An extra column is included for PESQ scores on speech.

TABLE IV
SI-SDR [dB] Results on the DnR Test Set

	Activity labels		SI-SDR			
	Oracle Train	Oracle Test	Music	Speech	SFX	Avg.
No Processing	—	—	-6.8	1.0	-5.0	-3.6
MRX	—	—	4.2	12.3	5.7	7.4
MRX-C	✓	✓	5.1	12.6	6.5	8.1
MRX-C	✓	×	4.4	12.5	5.9	7.6
MRX-C	×	×	4.6	12.6	6.1	7.8
MRX-C_{2p}	×	×	4.6	12.6	6.1	7.8
MRX-MTL	—	—	3.8	11.8	5.3	7.0

The columns “Oracle Train” and “Oracle test” denote whether the model has been trained and tested using oracle or predicted activity labels. MRX-C refers to the MRX conditioned on activity labels, MRX-C_{2p} to mrx conditioned on enhanced activity labels (2nd iterative pass), and MRX-MTL to a multi-task learning version of MRX which performs both SS and activity detection.

From Table III, we see that concatenating activity detection labels with the input spectrogram in the MRX-C model leads to the best overall separation performance, with some gains observed for all metrics compared to MRX. To further evaluate the impact of including activity detection labels as auxiliary inputs for source separation, Table IV compares different settings for activity-conditioned source separation. We note that MRX-C

[4] [Online]. Available: https://huggingface.co/espnet/simpleoier_librispeech_asr_train_asr_conformer7_hubert_ll60k_large_raw_en_bpe5000_sp

[5] [Online]. Available: https://github.com/ludlows/python-pesq, v0.0.4.

TABLE V
ACTIVITY DETECTION F-MEASURE FOR MUSIC (M), SPEECH (S), AND SOUND EFFECTS (X)

	Event-Based F-Measure			Segment-Based F-Measure		
	M	S	X	M	S	X
Baseline	0.77	0.84	0.50	0.97	0.97	0.92
MRX-MTL	0.78	0.87	0.43	0.97	0.97	0.92
Early Integration	**0.82**	0.88	**0.55**	**0.98**	**0.98**	**0.94**
Middle Integration	0.81	0.89	0.51	0.97	**0.98**	0.92
Late Integration	0.73	**0.91**	0.50	0.97	**0.97**	0.94

The baseline model uses the mixture as input, MRX-MTL uses multi-task learning for source separation and event detection, while the other rows insert source separation output at different locations inside the network.

TABLE VI
MAP AND MAUC CLASSIFICATION FOR MUSIC, SFX-FOREGROUND (SFX-FG.), AND SFX-BACKGROUND (SFX-BG.)

	Noisy		MRX-C		MRX-C Remix		GT	
Source	mAP	mAUC	mAP	mAUC	mAP	mAUC	mAP	mAUC
Music	0.233	0.682	0.297	0.723	0.300	0.718	0.336	0.772
SFX-Fg.	0.138	0.739	0.195	0.796	0.192	0.794	0.275	0.836
SFX-Bg.	0.137	0.653	0.143	0.656	0.148	0.666	0.258	0.767

MRX-C remix corresponds to the best remixing use-case scenario.

leads to an average SI-SDR improvement of 0.7 dB compared to MRX when using oracle information, meaning the upper bound in expected performance improvement is limited, however larger improvements are observed for the quieter and more difficult to separate sources (i.e., music and SFX). When we switch to the realistic setup that does not use oracle information at test time, we see that training using estimated activity detection probabilities leads to a 0.2 dB improvement compared to using oracle information. We also observe that running multiple passes between activity detection and separation (MRX-C_{2p}) led to no performance improvement. Finally, we also considered a multi-task learning setup, where activity detection was used only as an additional training objective (MRX-MTL), but observed a degradation in performance compared to our plain MRX model.

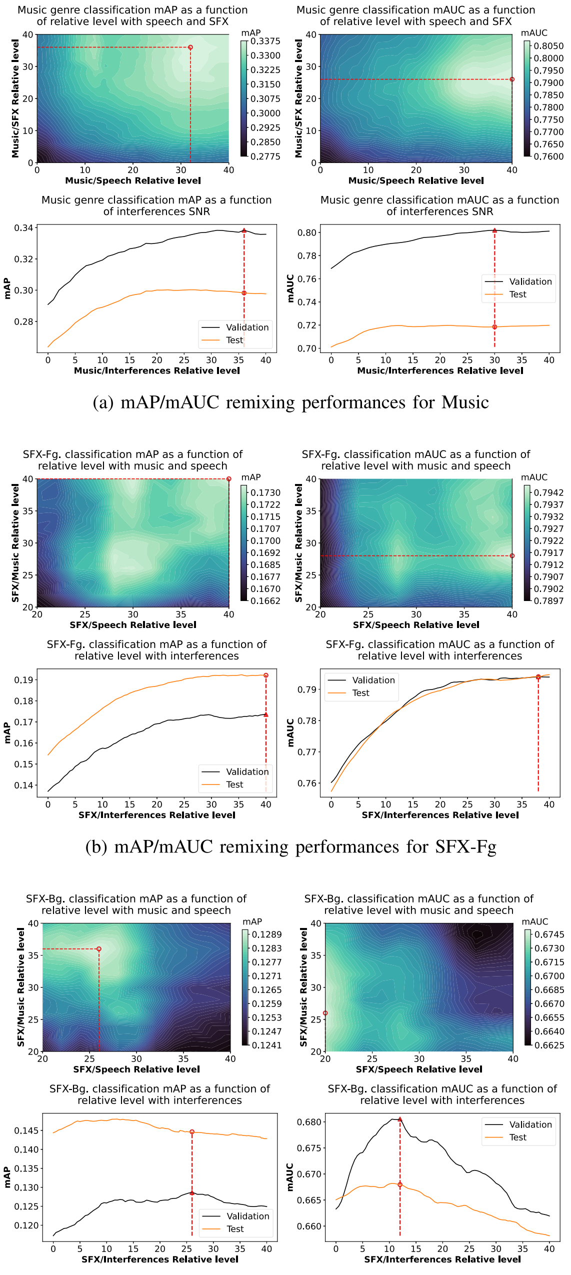
While we have just seen how activity detection can improve separation, we now turn our focus to how separation can aid activity detection. Owing to the well-established and intuitive nature of the metric, we follow a similar evaluation approach as in [36], [45], [63], [64] and report the per class F-measure. While additional metrics such as the polyphonic sound-event detection score (PSDS) [65] could also be included, we deem the F-measure for both segment and event-based evaluations to be sufficient in this case considering the lack of label subjectivity (i.e., we know the exact segment boundaries since our dataset is synthetic), and the well-balanced nature of our classification task. Table V displays the sound event detection performance in terms of parent-class F-measure computed using the SED EVAL package [66]. We used a threshold of 0.5, a collar of 750 ms for event-based metrics (with a 20% offset length), and a time resolution of 1 s for segment-based metrics. Compared to a Baseline taking the mixture signal as input, little improvement is observed using multi-task learning of activity detection and source separation (MRX-MTL). When integrating source separation output into our three-class activity detector, we observed the best performance using early or middle integration, which differs from the permutation-invariant analysis in [36] where late integration performed best. We suspect that integrating permutation-invariant source separation outputs at the input or at intermediate layers of the sound event detection network may cause difficulties during training, as the order in which separated outputs are stacked may change between training epochs. However, for problems such as the one studied in this article, where separated outputs have a fixed ordering, the activity detection network does not have this inconsistency problem and hence early and middle integration perform better than late integration. We also note that the activity detector used in MRX-C (Fig. 2) uses early integration.

B. CFP Transcription and Remixing

1) Audio Tagging (Music and SFX): We report audio tagging performance in terms of mAP and mAUC metrics using the implementation from [67]. Both mAP and mAUC allow us to evaluate audio tagging performance in a threshold-independent way for situations where multiple tags can be active for a single sound file, and are commonly reported for large-scale audio tagging [15], [53], [68]. Table VI displays the audio tagging performance for music, SFX-Fg, and SFX-Bg sources using oracle event boundaries. The "Noisy" column is obtained from the original mixture and represents lower-bound performance, and the "GT" column represents upper-bound performance obtained using the ground-truth isolated sources. The scores for music are generally higher than for SFX, because there is only a single genre tag for each music segment, the set of possible music labels is smaller than for SFX, and SFX-Fg and SFX-Bg sounds may overlap as discussed in Section IV-A. The "MRX-C" column from Table VI displays audio tagging performance using the separated outputs of MRX-C, our best separation model. It can be seen that for all sources in Table VI, source separation improves audio tagging performance compared to the noisy mixture. The improvements for Music and SFX-Fg are larger than those for SFX-Bg, likely because those sources have higher relative levels in the DnR mixes.

The "MRX-C Remix" column in Table VI shows the test set results of the systems which obtain the best performance on the validation set for each source in a grid search over remixing gains. We see that the benefit from source remixing is mixed for music (mAP improves, but mAUC decreases), SFX-Fg classification performance degrades for both metrics, while SFX-Bg demonstrates gains from remixing for both metrics.

Fig. 5 provides a detailed illustration of the grid search and the impact of different remixing gains on mAP and mAUC for music and SFX. The first row of Figs. 5(a), 5(b), and 5(c) shows remixing results on the validation set for the case where interfering source gains are adjusted individually as in (3), while the second row shows the results on the validation and test sets for the case where the gains are adjusted jointly as in (4). For each plot in Fig. 5, the data points denoted by the intersections with the red line indicate the results corresponding to the best validation set performance for the given metric over different remixing



(a) mAP/mAUC remixing performances for Music

(b) mAP/mAUC remixing performances for SFX-Fg

(c) mAP/mAUC remixing performances for SFX-Bg

Fig. 5. mAUC and mAP music genre classification mappings as a function of Music SNR against interfering sources. The contour plots (first row) show the classification performance on the validation set depending on the SNR against each interfering source individually, while the second row shows the performance on the validation and test sets for the scenario where the two interfering sources are combined in order to compute the SNR. The red symbols along the red dashed lines denote the performance on the test set (circle) where the best validation set performance is observed (triangle). (a) mAP/mAUC remixing performances for Music. (b) mAP/mAUC remixing performances for SFX-Fg. (c) mAP/mAUC remixing performances for SFX-Bg.

SNR values. While music and SFX-Fg obtain little benefit from source remixing, in the case of SFX-Bg, performance peaks at a much lower SNR, meaning that this source specifically benefits from remixing the other sources for its classification task.

TABLE VII
WER and CER (%) Results Using Different Boundary Types and Input Waveforms

	Oracle Boundaries		VAD Boundaries		No Boundaries	
Data	WER	CER	WER	CER	WER	CER
Libri. Test-Clean	1.8	0.5	—	—	—	—
DnR Speech GT	1.8	1.3	2.0	1.4	27.5	19.5
Noisy Mixture	16.4	12.3	17.5	13.0	81.7	63.7
MRX-C	6.5	4.1	7.0	4.5	**36.6**	**25.5**
MRX-C Remix	**5.7**	**3.7**	**6.3**	**4.1**	45.5	32.2

The performance on LibriSpeech Test-Clean is also included for reference.

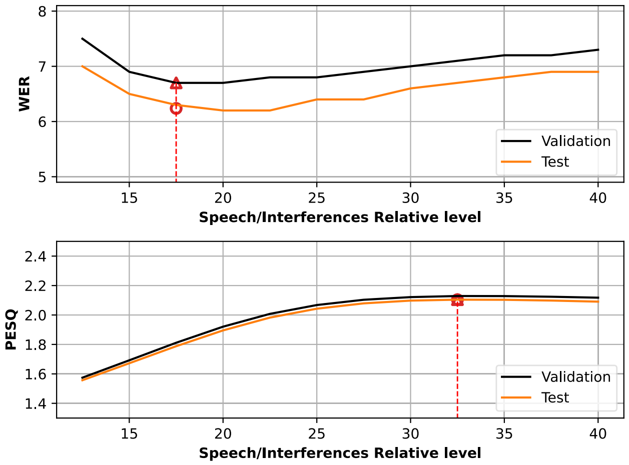


Fig. 6. WER (%) and PESQ performances on speech as a function of speech SNR against interfering sources (Music and SFX). Once again the symbols along the red dashed lines denote the points where the best validation set performance is observed (triangle) as well as the test-set performance associated with it (circle).

2) Automatic Speech Recognition: Table VII presents ASR results using ESPnet for three segmentation methods: using the oracle speech utterance onsets/offsets ("Oracle Boundaries"), using the onsets/offsets detected by Φ_{EI} ("VAD Boundaries"), and no segmentation ("No Boundaries"). We first note that in the "No Boundaries" case, ASR performance degrades dramatically, indicating that our pre-trained ASR model cannot handle 60 s long files containing multiple utterances. We tried training the ASR model from scratch on the DnR dataset speech files, but the performance still significantly lagged that of inputting segmented utterances into the pre-trained ASR model. We also note from Table VII that performance on the original LibriSpeech test set ("Libri. Test-Clean") and the clean DnR speech submix ("DnR Speech GT") is comparable in the case of oracle boundary segmentation, which is to be expected since they are identical utterances with slightly different levels added during the DnR data creation process. Performance degrades dramatically when the noisy DnR mixture is used for ASR, which is also to be expected since our model is pre-trained using clean speech. When using the separated speech stem from MRX-C, we obtain a 10.5% absolute (or 60% relative) reduction in WER compared to the noisy mixture when using VAD boundaries.

In the case of ASR, artifacts introduced by the separation process can degrade performance, and only partially separating the speech has been shown to improve performance [33], [34], [35]. Fig. 6 displays the performance in terms of WER and PESQ on the validation and test sets obtained for a grid search over the relative level between the remixed separated speech signal and the interference signals. Note that due to the large computational expense of speech decoding, we perform a grid search only for the combined (SFX+music) interference signal as described in (4). For WER, a minimum is observed at 17.5 dB, which is selected as the gain for the MRX-C Remix system. In the "MRX-C Remix" row of Table VII, we observe that by remixing the separated interference signal back with the estimated speech, WER is reduced from 7.0% down to 6.3% using VAD boundaries.

In the case of wideband PESQ from Fig. 6, the best score is achieved when the speech and interference signals are remixed at a relative level of 32.5 dB, achieving a score of 2.10, compared to 2.01 using the enhanced speech output in Table III. Source separation errors can both over-suppress true speech and fail to remove interference signals, and both types of errors can negatively impact perceptually inspired objective measures such as PESQ. However, the slight improvement from remixing suggests over-suppression might be a bigger issue for the MRX-C separation model. We also note that this over-suppression is more pronounced at high frequencies, as we noticed no improvements from remixing in terms of narrowband PESQ, which analyzes frequencies between 300-3100 Hz, as opposed to 50-7000 Hz in the wideband case [62] shown in Fig. 6.

VI. Conclusions and Discussions

In this work, we extended our previous work on the cocktail fork problem by tackling transcription for each of the three sources involved: audio tagging for music and sound effects, and ASR for speech. We proposed an activity detection system for the three parent classes and showcased its benefits towards both the separation and transcriptions tasks. We first demonstrated how the system could help improve separation by using the activity labels as conditioning information. Secondly, we described how the integration of an activity detection mechanism was essential in order to tackle real-world soundtrack transcription tasks. Lastly, we led an extensive investigation to show how source-remixing strategies could benefit transcription. Due to the imperfect nature of our separator output, we demonstrated that mixing back the interfering signals with the isolated source estimates could help improve performance on their associated downstream transcription task.

Moving forward, we aim to explore source-remixing strategies that could minimize perceptual artifacts for the separator output and enhance the listening experience. While we approached this work from a fully supervised angle, for both separation and transcription, taking advantage of the large amount of "real-world" unlabeled data available is an important topic for our future work. Finally, many real-world soundtracks contain a complex chain of post-processing audio effects, including non-linear processes such as dynamic range compression and lossy encoding that may complicate the separation process. Exploring techniques similar to [69] to quantify and mitigate issues caused by this non-linear processing will also be necessary to increase robustness and further improve separation quality.

References

[1] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 10, pp. 1702–1726, Oct. 2018.

[2] C. K. Reddy et al., "The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results," in *Proc. Interspeech*, 2020, pp. 2492–2496.

[3] J. R. Hershey, Z. Chen, and J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2016, pp. 31–35.

[4] L. Drude, J. Heitkaemper, C. Boeddeker, and R. Haeb-Umbach, "SMS-WSJ: Database, performance measures, and baseline recipe for multi-channel source separation and recognition," 2019, *arXiv:1910.13934*.

[5] G. Wichern et al., "WHAM!: Extending speech separation to noisy environments," in *Proc. Interspeech*, 2019, pp. 1368–1372.

[6] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1117372

[7] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-unmix - a reference implementation for music source separation," *J. Open Source Softw.*, vol. 4, no. 41, 2019, Art. no. 1667.

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

[9] I. Kavalerov et al., "Universal sound separation," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2019, pp. 175–179.

[10] E. Tzinis, S. Wisdom, J. R. Hershey, A. Jansen, and D. P. W. Ellis, "Improving universal sound separation using sound classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 96–100.

[11] F. Pishdadian, G. Wichern, and J. Le Roux, "Finding strength in weakness: Learning to separate sounds with weak supervision," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2386–2399, 2020.

[12] T. Ochiai, M. Delcroix, Y. Koizumi, H. Ito, K. Kinoshita, and S. Araki, "Listen to what you want: Neural network-based universal sound selector," in *Proc. Interspeech*, 2020, pp. 1441–1445.

[13] D. Petermann, G. Wichern, Z.-Q. Wang, and J. Le Roux, "The cocktail fork problem: Three-stem audio separation for real-world soundtracks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 526–530.

[14] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR–half-baked or well done?," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 626–630.

[15] J. F. Gemmeke et al., "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 776–780.

[16] C. Li et al., "ESPnet-SE: End-to-end speech enhancement and separation toolkit designed for ASR integration," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 785–792.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 5206–5210.

[18] A. W. Rix, J.G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2001, pp. 749–752.

[19] T. Kastner and J. Herre, "An efficient model for estimating subjective quality of separated audio source signals," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2019, pp. 95–99.

[20] E. Manilow, P. Seetharaman, and B. Pardo, "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 771–775.

[21] J. F. Woodruff, B. Pardo, and R. B. Dannenberg, "Remixing stereo music with score-informed source separation," in *Proc. Int. Soc. Music Inf. Retrieval*, 2006, pp. 314–319.

[22] H. Yang, K. Zhen, S. Beack, and M. Kim, "Source-aware neural speech coding for noisy speech compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 706–710.

[23] A. S. Subramanian et al., "Speech enhancement using end-to-end speech recognition objectives," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2019, pp. 229–233.

[24] X. Chang, T. Maekaku, Y. Fujita, and S. Watanabe, "End-to-end integration of speech recognition, speech enhancement, and self-supervised learning representation," in *Proc. Interspeech*, 9, 2022, pp. 3819–3823.

[25] N. Moritz, G. Wichern, T. Hori, and J. Le Roux, "All-in-one transformer: Unifying speech recognition, audio tagging, and event detection.," in *Proc. Interspeech*, 2020, pp. 3112–3116.

[26] O. Slizovskaia, L. Kim, G. Haro, and E. Gomez, "End-to-end sound source separation conditioned on instrument labels," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 306–310.

[27] Y.-N. Hung and A. Lerch, "Multitask learning for instrument activation aware music source separation," in *Proc. ISMIR*, Oct. 2020, pp. 748–755.

[28] E. Verteletskaya and K. Sakhnov, "Voice activity detection for speech enhancement applications," *Acta Polytechnica*, vol. 50, pp. 100–105, Jan. 2010.

[29] X. Tan and X.-L. Zhang, "Speech enhancement aided end-to-end multi-task learning for voice activity detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 6823–6827.

[30] N. L. Westhausen, R. Huber, H. Baumgartner, R. Sinha, J. Rennies, and B. T. Meyer, "Reduction of subjective listening effort for TV broadcast signals with recurrent neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3541–3550, 2021.

[31] M. Torcoli, J. Paulus, T. Kastner, and C. Uhle, "Controlling the remixing of separated dialogue with a non-intrusive quality estimate," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2021, pp. 91–95.

[32] O. Kalinli, M. L. Seltzer, J. Droppo, and A. Acero, "Noise adaptive training for robust automatic speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, pp. 1889–1901, Nov. 2010.

[33] Y. Koizumi, S. Karita, A. Narayanan, S. Panchapagesan, and M. Bacchiani, "SNRi target training for joint speech enhancement and recognition," in *Proc. Interspeech*, H. Ko and J. H. L. Hansen, Eds., ISCA, 2022, pp. 1173–1177.

[34] H. Sato, T. Ochiai, M. Delcroix, K. Kinoshita, N. Kamo, and T. Moriya, "Learning to enhance or not: Neural network-based switching of enhanced and observed signals for overlapping speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 6287–6291.

[35] K. Iwamoto et al., "How bad are artifacts?: Analyzing the impact of speech enhancement errors on ASR," in *Proc. Interspeech*, Sep. 2022, pp. 5418–5422.

[36] N. Turpault et al., "Improving sound event detection in domestic environments using sound separation," in *Proc. DCASE*, 2020, pp. 205–209.

[37] H. Yang, S. Firodiya, N. J. Bryan, and M. Kim, "Don't separate, learn to remix: End-to-end neural remixing with joint optimization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 116–120.

[38] C. Gupta, E. Yılmaz, and H. Li, "Automatic lyrics alignment and transcription in polyphonic music: Does background music help?," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 496–500.

[39] C.-Y. Chiu, J. Ching, W.-Y. Hsiao, Y.-H. Chen, A. W.-Y. Su, and Y.-H. Yang, "Source separation-based data augmentation for improved joint beat and downbeat tracking," in *Proc. IEEE 29th Eur. Signal Process. Conf.*, 2021, pp. 391–395.

[40] L. Lin, Q. Kong, J. Jiang, and G. Xia, "A unified model for zero-shot music source separation, transcription and synthesis," in *Proc. ISMIR*, Nov. 2021, pp. 381–388.

[41] P. Chandna, H. Cuesta, D. Petermann, and E. Gómez, "A deep-learning based framework for source separation, analysis, and synthesis of choral ensembles," *Front. Signal Process.*, vol. 2, pp. 1–15, 2022.

[42] H. Wierstorf, D. Ward, R. Mason, E. M. Grais, C. Hummersone, and M. Plumbley, "Perceptual evaluation of source separation for remixing music," in *Proc. Audio Eng. Soc. Conv.*, Oct. 2017, p. 9880.

[43] R. Sawata, S. Uhlich, S. Takahashi, and Y. Mitsufuji, "All for one and one for all: Improving music separation by bridging networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 51–55.

[44] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[45] J. Baumann, P. Meyer, T. Lohrenz, A. Roy, M. Papendieck, and T. Fingscheidt, "A new DCASE 2017 rare sound event detection benchmark under equal training data: CRNN with multi-width kernels," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 865–869.

[46] D. De Benito-Gorrón, D. Ramos, and D. T. Toledano, "A multi-resolution CRNN-based approach for semi-supervised sound event detection in DCASE 2020 challenge," *IEEE Access*, vol. 9, pp. 89029–89042, 2021.

[47] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2018, pp. 3942–3951.

[48] A. G. Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[49] J. F. Gemmeke et al., "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 776–780.

[50] A. Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020 pp. 5036–5040.

[51] W.-N. Hsu, B. Bolte, Y. -H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3451–3460, 2021.

[52] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *Proc. ISMIR*, Oct. 2017, pp. 316–323.

[53] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: An open dataset of human-labeled sound events," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 829–852, 2022.

[54] L. Zhang, C. Li, F. Deng, and X. Wang, "Multi-task audio source separation," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 671–678.

[55] T. Menne, I. Sklyar, R. Schlüter, and H. Ney, "Analysis of deep clustering as preprocessing for automatic speech recognition of sparsely overlapping speech," in *Proc. Interspeech*, 2019, pp. 2638–2642.

[56] S. Chaudhuri et al., "AVA-speech: A densely labeled dataset of speech activity in movies," in *Proc. Interspeech*, 2018, pp. 1239–1243.

[57] E. M. Grimm, R. Van Everdingen, and M. J. L. C. Schöpping, "Toward a recommendation for a European standard of peak and LKFS loudness levels," *SMPTE Mot. Imag. J.*, vol. 119, no. 3, pp. 28–34, Apr. 2010.

[58] H. Erdogan, J. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 708–712.

[59] M. Torcoli, T. Kastner, and J. Herre, "Objective measures of perceptual audio quality reviewed: An evaluation of their application domain dependence," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 1530–1541, 2021.

[60] C. Colomes, C. Schmidmer, T. Thiede, and W. C. Treurniet, "Perceptual quality assessment for digital audio: PEAQ-the new ITU standard for objective measurement of the perceived audio quality," in *Proc. AES 17th Int. Conf. High Qual Audio Coding*, Sep. 1999, pp. 337–351.

[61] P. Kabal, "An examination and interpretation of ITU-R BS. 1387: Perceptual evaluation of audio quality," Dept. Electr. Comput. Eng., McGill University, Montreal, QC, Canada,Tech. Rep., 2002.

[62] "P862.2: Wideband extension to recommendation P.862 for the assessment of wideband telephone networks and speech codecs," 2007. [Online]. Available: https://www.itu.int/rec/T-REC-P.862.2-200711-I/en

[63] D. Stowell and D. Clayton, "Acoustic event detection for multiple overlapping similar sources," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2015, pp. 1–5.

[64] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP J. Audio Speech Music Process.*, vol. 2013, pp. 1–13, 2013.

[65] C. Bilen, G. Ferroni, J. Azcarreta, and S. Krstulovic, "A framework for the robust evaluation of sound event detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 61–65.

[66] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Appl. Sci.*, vol. 6, no. 6, 2016, Art. no. 162.

[67] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman, "Vggsound: A large-scale audio-visual dataset," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 721–725.

[68] S. Hershey et al., "CNN architectures for large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 131–135.

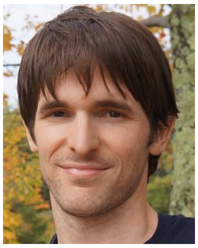
[69] C.-B. Jeon and K. Lee, "Towards robust music source separation on loud commercial music," in *Proc. ISMIR*, 2022, pp. 575–582.

Darius Petermann (Graduate Student Member, IEEE) received the bachelor's degree from the Berklee College of Music, Boston, MA, USA, and the M.Sc. degree from Music Technology Group, Universitat Pompeu-Fabra, Barcelona, Spain. He is currently working toward the Ph.D. degree in intelligent systems engineering with Signals and AI Group in Engineering, Indiana University, Bloomington, IN, USA. He previously interned with Mitsubishi Electrical Research Laboratories, Cambridge, MA, USA. He is a Student Researcher with Google, working on source separation problems.

Gordon Wichern (Member, IEEE) received the B.Sc. and M.Sc. degrees from Colorado State University, Fort Collins, CO, USA, in electrical engineering and the Ph.D. degree from Arizona State University, Tempe, AZ, USA, in electrical engineering with concentration in arts, media and engineering, where he was supported by National Science Foundation, Integrative Graduate Education, and Research Traineeship for his work on environmental sound recognition. He is currently a Senior Principal Research Scientist with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. He was a Member of research team with iZotope, inc, Cambridge, where he focused on applying novel signal processing and machine learning techniques to music and post production software. He is also a Member of the Technical Staff with MIT Lincoln Laboratory, Lexington, MA, where he worked on radar signal processing. He is the Chair of the AES Technical Committee on Machine Learning and Artificial Intelligence and a Member of IEEE Audio and Acoustic Signal Processing Technical Committee.

Aswin Shanmugam Subramanian received the M.S.E. and Ph.D. degrees from Johns Hopkins University, Baltimore, MD, USA, in 2017 and 2022, respectively. From 2021 to 2022, he was a Research Scientist with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. He is currently a Sr. Applied Scientist with Microsoft Corporation, Redmond, WA, USA. His research interests include automatic speech recognition, microphone array processing, source separation, and speaker diarization.

Zhong-Qiu Wang received the B.E. degree in computer science and technology from the Harbin Institute of Technology, Harbin, China, in 2013, and the M.S. and Ph.D. degrees in computer science and engineering from The Ohio State University, Columbus, OH, USA, in 2017 and 2020, respectively. He is currently a Postdoctoral Research Associate with Carnegie Mellon University, PA, USA. From 2020 to 2021, he was a Visiting Research Scientist with Mitsubishi Electrical Research Laboratories, Cambridge, MA, USA. He is a Committee Member of the IEEE Audio and Acoustic Signal Processing Technical Committee, during 2023–2025. His research interests include computer audition, machine hearing, microphone array processing, speech separation, robust automatic speech recognition, machine learning, and deep learning.



Jonathan Le Roux (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in mathematics from Ecole Normale Supérieure, Paris, France and the Ph.D. degree from the University of Tokyo, Tokyo, Japan, and the Université Pierre et Marie Curie, Paris. He is a Distinguished Research Scientist and Speech and Audio Senior Team Leader with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. From 2009 to 2011, he was a Postdoctoral Researcher with NTT's Communication Science Laboratories, Kyoto, Japan. He has contributed to more than 140 peer-reviewed papers and 30 granted patents in his research areas which inlude signal processing and machine learning applied to speech and audio. He is the Founder and Chair of Speech and Audio in the Northeast series of workshops.