MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Transcription Is All You Need: Learning to Separate Musical Mixtures with Score as Supervision

Hung, Yun-Ning; Wichern, Gordon; Le Roux, Jonathan

TR2021-069 June 08, 2021

Abstract

Most music source separation systems require large collections of isolated sources for training, which can be difficult to obtain. In this work, we use musical scores, which are comparatively easy to obtain, as a weak label for training a source separation system. In contrast with previous score-informed separation approaches, our system does not require isolated sources, and score is used only as a training target, not required for inference. Our model consists of a separator that outputs a time-frequency mask for each instrument, and a transcriptor that acts as a critic, providing both temporal and frequency supervision to guide the learning of the separator. A harmonic mask constraint is introduced as another way of leveraging score information during training, and we propose two novel adversarial losses for additional fine-tuning of both the transcriptor and the separator. Results demonstrate that using score information outperforms temporal weak-labels, and adversarial structures lead to further improvements in both separation and transcription performance.

IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

TRANSCRIPTION IS ALL YOU NEED: LEARNING TO SEPARATE MUSICAL MIXTURES WITH SCORE AS SUPERVISION

Yun-Ning Hung[1,2], *Gordon Wichern*[1], *Jonathan Le Roux,*[1]

[1]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA
[2]Center for Music Technology, Georgia Institute of Technology, Atlanta, GA, USA
amyhung@gatech.edu, {wichern,leroux}@merl.com

ABSTRACT

Most music source separation systems require large collections of isolated sources for training, which can be difficult to obtain. In this work, we use musical scores, which are comparatively easy to obtain, as a weak label for training a source separation system. In contrast with previous score-informed separation approaches, our system does not require isolated sources, and score is used only as a training target, not required for inference. Our model consists of a separator that outputs a time-frequency mask for each instrument, and a transcriptor that acts as a critic, providing both temporal and frequency supervision to guide the learning of the separator. A harmonic mask constraint is introduced as another way of leveraging score information during training, and we propose two novel adversarial losses for additional fine-tuning of both the transcriptor and the separator. Results demonstrate that using score information outperforms temporal weak-labels, and adversarial structures lead to further improvements in both separation and transcription performance.

Index Terms— audio source separation, weakly-supervised separation, weakly-labeled data, music transcription

1. INTRODUCTION

Music source separation has long been an important task for music information retrieval, and recent advances in deep neural networks, have led to dramatic performance improvements. Most current methods rely on supervised learning, which requires a dataset including separated tracks, or stems for each instrument. However, copyright issues prevent wide availability of stems for most commercial music, and open-source datasets suffer from various drawbacks. For example, MUSDB [1] has a limited number of instruments, MIR-1K [2] only contains short clips of music, and MedleyDB [3] features an unbalanced amount of instrument categories. Most importantly, compared to large-scale datasets such as AudioSet [4] and The Million Song dataset [5], which are used in audio or music classification, source separation datasets are relatively small.

In contrast, it is more practical to obtain a musical score for a track than its isolated stems. Trained musicians are able to transcribe either part of the instruments or the whole song, which contributes to a large number of accessible scores. For example, the Musescore [6] and Lakh MIDI [7] datasets were both collected from online forums, while the SIMSSA project [8] provides the ability to generate digital scores from existing classical sheet music.

Based on this data advantage, in this work, we investigate whether score information alone can be used as a weak label to train a music source separation system without access to isolated stems. Many previous works have studied incorporating score information into music source separation, either as an additional training target to learn musically relevant source separation models [9], or as part of "score-informed" source separation approaches [10–14], where scores are used as an additional input feature or conditioning mechanism. However, all these methods are still trained in a supervised manner, and most need the score during inference as well. Different from these works, our proposed approach does not need to use isolated sources during training. The model directly learns from musical scores, and only needs the music mixture during inference.

In addition to musical score, other types of weak labels have recently been used for sound event separation. For example, [15, 16] introduce model structures that only need class labels for training a source separation system, rather than labels on each time-frequency bin. Instead of using ground truth class labels, [17, 18] propose to combine sound event detection systems with source separation systems to achieve large-scale sound separation. Visual information is leveraged to separate sound in [19, 20], where video features are available at both training and inference time. Generative adversarial network architectures have also been proposed to learn the distribution of isolated sources [21, 22], without requiring isolated sources associated with an input mixture. Without observing the target source in isolation, the models in [23, 24] learn to separate from corrupted target observations and regions of isolated interference.

Our problem definition is similar to that of Pishdadian et al [16], in that only weak labels are available as training targets, not the strong labels obtained from isolated sources. However, the weak labels we consider here are musical scores, rather than temporal labels of source activity. While it may sound more restrictive, this is a realistic assumption for music separation. Furthermore, using scores allows us to cope with several drawbacks that may occur when extending the approach in [16], which relies on supervision by a classifier (pre-trained on audio mixtures and corresponding source activity labels), to the music domain. First, since only temporal information regarding source presence is provided to the classifier, harmonic sounds (e.g., sirens or instruments) are not well separated, as temporal information alone is not enough to isolate upper harmonics. Second, the model does not learn to separate well when two sounds consistently appear together, a scenario that often occurs in music. For example, bass and drums often consistently play through entire songs. To solve the above problems, we propose using a transcriptor (pre-trained on mixtures and their musical scores) to provide both temporal and frequency information when training the separator. Moreover, we incorporate harmonic masks generated form the score to better model harmonic structure, and present two novel adversarial fine-tuning strategies to improve the ability of the separator to learn from weak labels.

This work was performed while Y. Hung was an intern at MERL.

2. TRAINING WITH TRANSCRIPTION AS SUPERVISION

We adopt a common source separation pipeline. The input mixture is first transformed into a time-frequency (TF) representation $X = (X_{f,t})$ (e.g., STFT magnitude), where f and t denote time and frequency indices. A TF mask $M_i = (M_{i,f,t})$ for the i-th source (i.e., instrument) is then estimated by the source separation model from X. The separated TF representation $\hat{S}_i = (\hat{S}_{i,f,t})$ is obtained by multiplying $M_{i,f,t}$ with $X_{f,t}$ at each TF bin (f,t). After combining each $\hat{S}_i$ with the mixture phase, an inverse transform (e.g., iSTFT) is applied to produce the estimated sources.

During training, we assume only the music mixture and its score are available. Note that we here assume the audio and musical scores are well aligned, in practice using audio synthesized from MIDI. But as we only require scores during training (not at inference time), computationally intensive offline alignment algorithms such as those based on dynamic time warping [25] could be incorporated as a preprocessing step. We leave the alignment problem as future work.

Our proposed model is composed of two components, a *transcriptor* and a *separator*. The model is trained using a three-step training approach. In the first step, we train a transcriptor to transcribe a mixture of music. Once the transcriptor is well trained, in the second step, we use the transcriptor as a critic to guide the learning of the separator. After this isolated training, the transcriptor and the separator are fine-tuned together in a final step.

2.1. Step 1: Transcriptor Training

We first train the transcriptor to estimate the musical score given a mixture spectrogram, as illustrated in the blue region of Fig. 1. We represent the target score $Y = (Y_{i,n,t})$ as a multi-instrument piano roll, a tensor with instruments indexed by i, the 88 MIDI notes by n, and time by t. Following [6], two marginalized representations, multi-pitch labels and instrument activations, are also derived from the piano roll as additional training targets to improve music transcription performance. The binary cross entropy loss $H(y,\hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$ is used to evaluate the error between an estimated output probability $\hat{y}$ and a ground truth label y. The transcriptor is updated based on three loss terms, score estimation, instrument estimation, and multi-pitch estimation:

$$\mathcal{L}_{\text{score}}(X,Y) = \sum_{i,n,t} H(Y_{i,n,t}, p_{i,n,t}(X)), \tag{1}$$

$$\mathcal{L}_{\text{inst}}(X,Y) = \sum_{n,t} H(\max_i(Y_{i,n,t}), \max_i(p_{i,n,t}(X))), \tag{2}$$

$$\mathcal{L}_{\text{pitch}}(X,Y) = \sum_{i,t} H(\max_n(Y_{i,n,t}), \max_n(p_{i,n,t}(X))), \tag{3}$$

where $p_{i,n,t}(X)$ denotes the estimated transcriptions for instrument i at note n and time t. The main transcription loss $\mathcal{L}_{\text{TL}}$ is

$$\mathcal{L}_{\text{TL}} = \mathcal{L}_{\text{score}} + \alpha_1 \mathcal{L}_{\text{inst}} + \beta_1 \mathcal{L}_{\text{pitch}} \tag{4}$$

where α_1 and β_1 are weights used to balance the contribution of the marginalized objectives.

2.2. Step 2: Separator Training

We next use the transcriptor pre-trained in the first step with fixed parameters as a critic to train the separator, as described in the red region of Fig. 1. The separator has to output separated sources that are good enough for the transcriptor to transcribe into the correct score. The separator is given a mixture spectrogram and has to estimate separated spectrograms $\hat{S}_i$, which are then given to the transcriptor as input for computing score estimates $p_{i,n,t}(\hat{S}_i)$. Since each separated source should ideally only contain information from one instrument, the score estimated from the separated source should only have one active instrument while all others should be zero. The separator is updated via the sum of the transcription losses on each separated source $\hat{S}_i$:

$$\mathcal{L}_{\text{score}}(\hat{S}_i, Y^{(i)}) = \textstyle\sum_{n,t} \left(H(Y_i, p_{i,n,t}(\hat{S}_i)) + \sum_{j \neq i} H(0, p_{j,n,t}(\hat{S}_i)) \right), \tag{5}$$

where $Y^{(i)}$ are the labels obtained from Y by setting to 0 all sources other than i, i.e., $Y^{(i)}_{j,n,t} = \delta_{i,j} Y_{j,n,t}$.

A mixture loss is introduced in [16] to prevent the output masks from solely focusing on the discriminating components for transcription, encouraging the sum of active components to be close to the input mixture, and the sum of inactive components to be close to zero. Depending on whether ground truth activity is available at the clip or frame level, this leads to so-called clip-level and frame-level mixture losses. As instrument activity tends to be consistent over the short clips (4 seconds) that we use for training, we here only consider the clip-level loss (frame-level loss lead to similar results):

$$\mathcal{L}_{\text{c-mix}} = \sum_{f,t} |X_{f,t} - \sum_{i \in \mathcal{A}} \hat{S}_{i,f,t}| + \sum_{f,t} \sum_{i \notin \mathcal{A}} |\hat{S}_{i,f,t}|, \tag{6}$$

where $\mathcal{A}$ is the index set of active sources in a training clip.

To further leverage score information, we here introduce a harmonic mask mixture loss $\mathcal{L}_{\text{h-mix}}$ to strengthen the harmonic structure of the separated sources. That is, we devise a within-frame notion of activity based on the harmonic structure of each instrument in each frame [11]. The harmonic structure can be calculated from the piano roll representation. Assuming the tuning frequency of all songs is f_q and its corresponding MIDI note number is n_q, we can compute the fundamental frequency for note n as $f_0 = f_q \cdot 2^{\frac{1}{12}\cdot(n-n_q)}$. With the fundamental frequency, we can further calculate the harmonic frequencies $f_l = f_0 * (l+1)$ where $l = 1,\ldots,L$ indexes the harmonics. We then create the set $\mathcal{A}_{f,t}$ of active sources expected to have energy in a given bin of the TF representation based on information from the score. This leads to the following loss:

$$\mathcal{L}_{\text{h-mix}} = \sum_{f,t} |X_{f,t} - \sum_{i \in \mathcal{A}_{f,t}} \hat{S}_{i,f,t}| + \sum_{f,t} \sum_{i \notin \mathcal{A}_{f,t}} |\hat{S}_{i,f,t}|. \tag{7}$$

With this strong constraint, the model is able to focus on the harmonic structure. However, since the harmonic energy may not always concentrate at a single TF bin, and since fluctuations in f_0 may occur, frequencies within ± 1 bin of an expected active harmonic f_l are also considered active. Defining scalar weights α_2 and β_2, the overall objective of separator training is

$$\mathcal{L}_{\text{sep}} = \sum_i \mathcal{L}_{\text{score}}(\hat{S}_i, Y^{(i)}) + \alpha_2 \mathcal{L}_{\text{c-mix}} + \beta_2 \mathcal{L}_{\text{h-mix}}. \tag{8}$$

2.3. Step 3: Joint Training

We now fine-tune the previously trained transcriptor and separator by training them together. Without further loss terms, training of the transcriptor is not influenced by that of the separator, but we found that fine-tuning the separator against a slowly evolving transcriptor resulted in better performance than against a converged and fixed transcriptor. However, training the transcriptor to better transcribe

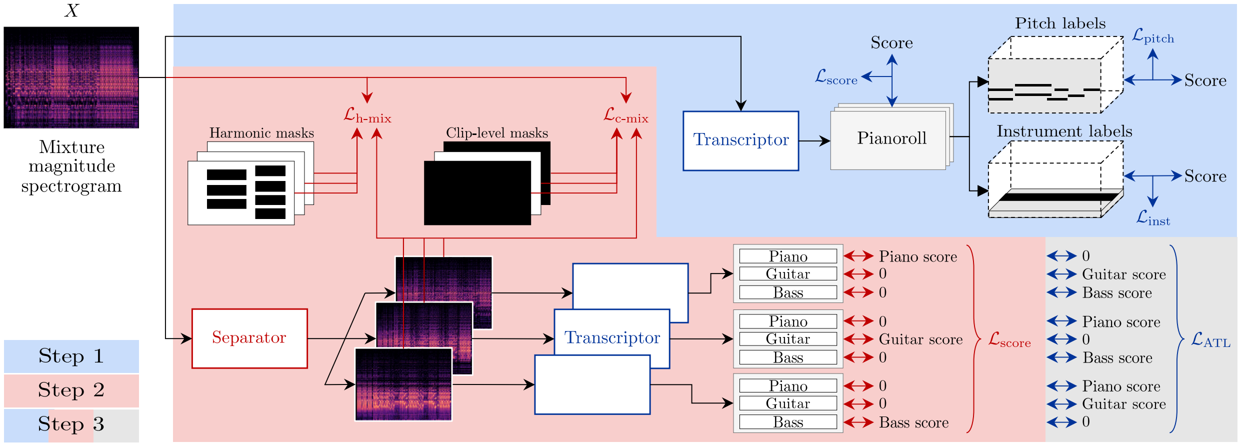

Fig. 1. Our proposed training strategy. Region colors indicate the associated training steps. Red and blue loss terms are respectively used to update the separator and the transcriptor. The grey region shows the adversarial transcription loss described in Section 2.3.

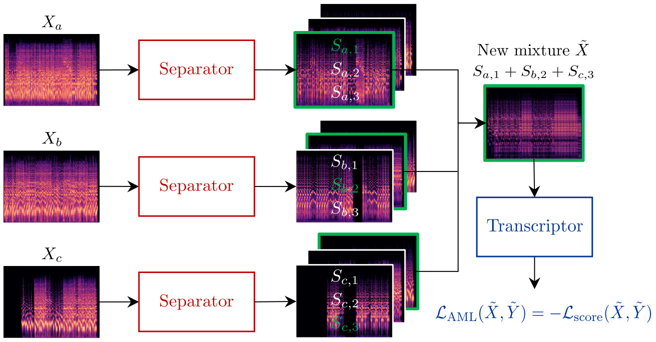

Fig. 2. Diagram of the adversarial mixture loss. The blue loss term is only used to update the transcriptor.

the separated outputs might cause it to co-adapt to the separator's mistakes, as shown in [16]. To solve this problem we introduce adversarial loss terms to make the transcriptor more sensitive to errors or incongruities in the separator output.

We first consider an adversarial mixture loss $\mathcal{L}_{\text{AML}}$, illustrated in Fig. 2. Denoting by $\hat{S}_{a,i}$ the i-th separated spectrogram for the a-th sample, we randomly pick one separated track for each instrument i from a different sample a_i to create a new remixed mixture $\tilde{X} = \sum_i \hat{S}_{a_i,i}$, and corresponding remixed labels $\tilde{Y}$. Since the separated sources likely contain errors, we penalize the transcriptor for correctly transcribing these imperfect remixes via a negative loss:

$$\mathcal{L}_{\text{AML}}(\tilde{X},\tilde{Y}) = -\mathcal{L}_{\text{score}}(\tilde{X},\tilde{Y}) \tag{9}$$

We also consider an adversarial transcription loss $\mathcal{L}_{\text{ATL}}$ as a counterpart to (5), based on the assumption that the imperfect $\hat{S}_{i,f,t}$ contains incomplete information from the target instrument i and residual information from other instruments. We thus ask the transcriptor not to recognize the target instrument, estimating zero as the target score, while transcribing non-target instruments to their correct scores, leading to a loss function $\mathcal{L}_{\text{ATL}}(\hat{S}_i,\overline{Y^{(i)}}) = \mathcal{L}_{\text{score}}(\hat{S}_i,\overline{Y^{(i)}})$, where $\overline{Y^{(i)}} = Y - Y^{(i)}$ is the complement of $Y^{(i)}$:

$$\mathcal{L}_{\text{ATL}}(\hat{S}_i,\overline{Y^{(i)}}) = \textstyle\sum_{n,t}\big(H(0,p_{i,n,t}(\hat{S}_i)) + \sum_{j\neq i} H(Y_j,p_{j,n,t}(\hat{S}_i)\big). \tag{10}$$

During step 3 training, the separator is still updated as in (8), while the adversarial terms are added with weights α_3 and β_3 to the original transcriptor loss (4), resulting in

$$\mathcal{L}_{\text{TL-3}} = \mathcal{L}_{\text{TL}} + \alpha_3\mathcal{L}_{\text{AML}} + \beta_3\mathcal{L}_{\text{ATL}}. \tag{11}$$

3. EXPERIMENT

3.1. Dataset

Although many datasets have been proposed for music transcription, they have some limitations. For example, Bach10 [26] and Su [27] contain a limited amount of songs; MusicNet [28] only includes classical music; MedleyDB [3] only has score annotation for monophonic instruments, not polyphonic instruments. As a result, we leverage the recently released synthesized dataset Slakh2100 [29] for training and evaluating our system. Slakh is synthesized from the Lakh MIDI Dataset [7] using professional-grade instruments. It contains aligned mono audio and MIDI score with 145 hours of data and 34 instrument categories. We use the *Slakh2100-split2* training, validation, and test splits, and down-sample the audio to 16 kHz. We use MIDI as a proxy for scores, for simplicity, and only consider mixtures of acoustic piano, distorted electric guitar, and electric bass in our current experiments, as we found certain synthesized instrument classes in Slakh2100 to be perceptually indistinguishable from one another (e.g., certain electric piano and clean electric guitar patches). While vocals are unavailable and there is not a consistent interpretation of MIDI drum scores in our synthesized dataset, if such data becomes available there would be no obstacles preventing the inclusion of vocals and drums in the proposed framework. Harmonic masks could be replaced by percussive masks in the case of drums.

3.2. System Details

We use the same temporal convolutional network (TCN) model for both our transcriptor and separator architectures since it has been shown to be efficient in source separation [30,31]. Our TCN contains three repeated layers with each layer including eight dilated residual blocks. All convolution layers have kernel size of 3. The output layers of the transcriptor and the separator are fully connected layers with sigmoid activation, the former producing a piano roll for each instrument, and the latter producing one TF mask per instrument. The inputs to both transcriptor and separator are magnitude STFT. The STFT is computed using a square root Hann window of size

Table 1. Separation performance in terms of SI-SDR (dB), where 'isolated' and 'fine-tune' respectively indicate the training setup of step 2 and step 3.

Training	$\mathcal{L}_{\text{c-mix}}$	$\mathcal{L}_{\text{h-mix}}$	$\mathcal{L}_{\text{AML}}$	$\mathcal{L}_{\text{ATL}}$	Bass	Guitar	Piano	Avg
Supervised					11.1	5.7	7.7	8.2
isolated	✓				7.5	1.2	4.2	4.3
isolated		✓			7.8	0.4	4.1	4.1
isolated	✓	✓			**8.4**	**1.6**	**5.0**	**5.0**
fine-tune	✓	✓			9.0	2.7	5.3	5.6
fine-tune	✓	✓	✓		**9.1**	2.8	5.4	**5.8**
fine-tune	✓	✓		✓	9.0	2.5	**5.7**	5.7
fine-tune	✓	✓	✓	✓	9.0	**2.9**	5.4	**5.8**
Input mixture					1.2	−5.8	−2.3	−2.3
Baseline [16]					7.3	0.5	3.5	3.8

Table 2. Transcription performance in terms of note accuracy, where 'pre-train' and 'fine-tune' respectively indicate the training setup of step 1 and step 3, and 'mixture' and 'iso tracks' respectively represent mixture and isolated ground truth audio.

Training	Evaluated on	$\mathcal{L}_{\text{AML}}$	$\mathcal{L}_{\text{ATL}}$	Bass	Guitar	Piano
pre-train	mixture			0.85	0.44	0.58
fine-tune	mixture			0.84	0.42	0.54
fine-tune	mixture	✓		0.86	0.51	0.61
fine-tune	mixture		✓	0.85	0.50	0.60
pre-train	iso tracks			0.91	0.52	0.66
fine-tune	iso tracks			0.90	0.53	0.63
fine-tune	iso tracks	✓		0.91	0.58	0.68
fine-tune	iso tracks		✓	0.91	0.57	0.66

2048 samples and a hop size of 500. The output of the transcriptor is a three-dimensional piano-roll representation extracted using `pretty_midi` [32]. As described in [16], instrument dependent weights calculated based on the rate of frame-wise activity in the training set are applied to the transcription loss to compensate for class imbalance. The activity rate is 0.41 for piano, 0.14 for guitar and 0.67 for bass. All the loss functions are given a weight to balance the loss during training process, with the weights manually tuned on development data and set to $\alpha_1 = 0.03$, $\beta_1 = 0.1$, $\alpha_2 = 1$, $\beta_2 = 0.1$, $\alpha_3 = 0.2$, and $\beta_3 = 0.05$. The various setups use subsets of the mixture and adversarial losses, and the weights of the unused terms are then set to 0. Systems are trained with the Adam optimizer and a learning rate 0.001, which decays by a factor of 0.5 if the loss on the validation set does not improve for two consecutive epochs. Training stops if the validation loss does not improve for 10 epochs.

3.3. Evaluation

We compare our approach with the baseline method proposed in [16], which uses a classifier to perform weakly supervised source separation, with the difference that the classifier is designed to have the same TCN structure as the transcriptor for comparison. Only the last linear layer in the model is modified to output instrument activation. We report the scale-invariant signal-to-distortion (SI-SDR) ratio [33] of each instrument in various settings in Table 1. We observe that using a transcriptor leads to better separation performance than using a classifier. This shows that providing weak frequency information can benefit separating harmonic sounds, such as musical instruments. Although using a harmonic mask via $\mathcal{L}_{\text{h-mix}}$ alone has slightly lower performance, adding $\mathcal{L}_{\text{h-mix}}$ with $\mathcal{L}_{\text{c-mix}}$ improves average SI-SDR by 0.7 dB. This is likely because the harmonic mask is too strict in constraining the separated spectrogram, in particular forcing positions that are not exactly harmonic to be zero is not realistic and might influence separation quality. However, when combined with coarse clip-level activity constraints, $\mathcal{L}_{\text{h-mix}}$ can provide some information on harmonic structure which is lacking in $\mathcal{L}_{\text{c-mix}}$.

The bottom part of Table 1 shows the result of jointly training the transcriptor and separator (fine-tune), starting from the best setup in step 2. We see that joint training without any adversarial loss improves separation performance, especially on guitar. As mentioned in Section 2.3, this may be because, when fine-tuning the transcriptor and separator together, the slowly evolving transcriptor acts as a regularizer, preventing the separator from easily exploiting loopholes in the transcriptor. Moreover, by using an adversarial training strategy, we can further improve average SI-SDR by 0.2 dB, especially benefitting piano and guitar. We also include the supervised training result obtained with the same data split and separator architecture, where the separator is trained to minimize an L_1 loss function between $\hat{S}_i$ and the ground truth target spectrogram S_i. Other training settings are unchanged. Although the best weakly-supervised result is 2.4 dB worse than the supervised scenario, our approach still closes a significant part of the gap from the input SI-SDR (using mixture as input).

Table 2 compares the transcription results when first pre-training on mixtures alone (step 1) and when fine-tuning (step 3). Fine-tuning without adversarial loss simply means longer training, which may overfit. The note accuracy is calculated by `mir_eval` [34]. The top section is evaluated using mixtures as the input. We observe that using adversarial training provides improvement in transcription accuracy, especially for guitar. Furthermore, we calculate the false positive rate of detected notes for all step 3 models in Table 2 and find that the model trained without adversarial losses has a higher false positive rate (0.37) than models trained with $\mathcal{L}_{\text{AML}}$ or $\mathcal{L}_{\text{ATL}}$ (0.17 and 0.19 respectively). Because guitar and piano have similar note ranges, adversarial training appears to help the model distinguish between these two instruments. The lower section of Table 2 presents the results when evaluating on isolated ground truth audio. We observe a similar pattern as evaluating on the mixture, with guitar improving the most. This demonstrates that although our transcriptor is only trained on mixtures, it can generalize to isolated tracks.

Example separation results are available at:
https://biboamy.github.io/Transcription_separation_ICASSP21/

4. CONCLUSION

In this work, we proposed a three-step training method to train a weakly-supervised music source separation model with musical scores. The results demonstrate that using a transcriptor to provide both temporal and frequency information as supervision outperforms the baseline method which only includes temporal information. We showed that using harmonic masks derived from the musical score, as well as adversarial losses on the transcriptor, leads to improved separation and transcription performance. Future work includes tackling the alignment problem to expand the range of data that can be used for training, and considering separation of non-harmonic instruments such as drums within our framework. We also plan to explore combining our approach with fully-supervised algorithms, either as an additional objective in multitask learning or in a semi-supervised setting, where a small training set is augmented with large amounts of mixtures and the corresponding musical scores.

5. REFERENCES

[1] 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

[2] C.-L. Hsu and J.-S. R. Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, pp. 310–319, 2010.

[3] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research." in *Proc. ISMIR*, vol. 14, 2014, pp. 155–160.

[4] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. ICASSP*, Mar. 2017, pp. 776–780.

[5] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere, "The million song dataset." in *Proc. ISMIR*, Oct. 2011, pp. 591–596.

[6] Y.-N. Hung, Y.-A. Chen, and Y.-H. Yang, "Multitask learning for frame-level instrument recognition," in *Proc. ICASSP*, May 2019, pp. 381–385.

[7] C. Raffel, "Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching," Ph.D. dissertation, Columbia University, 2016.

[8] J. Calvo-Zaragoza, F. J. Castellanos, G. Vigliensoni, and I. Fujinaga, "Deep neural networks for document processing of music score images," *Applied Sciences*, vol. 8, no. 5, p. 654, 2018.

[9] E. Manilow, P. Seetharaman, and B. Pardo, "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments," in *Proc. ICASSP*, May 2020, pp. 771–775.

[10] M. Miron, J. J. Carabias-Orti, J. J. Bosch, E. Gómez, and J. Janer, "Score-informed source separation for multichannel orchestral recordings," *J. Electr. Comput. Eng.*, 2016.

[11] M. Miron, J. Janer, and E. Gómez, "Monaural score-informed source separation for classical music using convolutional neural networks," in *Proc. ISMIR*, vol. 18, 2017, pp. 569–576.

[12] S. Ewert and M. B. Sandler, "Structured dropout for weak label and multi-instance learning and its application to score-informed source separation," in *Proc. ICASSP*, Mar. 2017, pp. 2277–2281.

[13] O. Slizovskaia, L. Kim, G. Haro, and E. Gomez, "End-to-end sound source separation conditioned on instrument labels," in *Proc. ICASSP*, May 2019, pp. 306–310.

[14] K. Schulze-Forster, C. Doire, G. Richard, and R. Badeau, "Weakly informed audio source separation," in *Proc. WASPAA*, Oct. 2019, pp. 273–277.

[15] E. Karamatli, A. T. Cemgil, and S. Kirbiz, "Audio source separation using variational autoencoders and weak class supervision," *IEEE Signal Process. Lett.*, vol. 26, no. 9, pp. 1349–1353, 2019.

[16] 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.

[17] E. Tzinis, S. Wisdom, J. R. Hershey, A. Jansen, and D. P. W. Ellis, "Improving universal sound separation using sound classification," in *Proc. ICASSP*, May 2020.

[18] Q. Kong, Y. Wang, X. Song, Y. Cao, W. Wang, and M. D. Plumbley, "Source separation with weakly labelled data: An approach to computational auditory scene analysis," in *Proc. ICASSP*, May 2020, pp. 101–105.

[19] R. Gao and K. Grauman, "Co-separating sounds of visual objects," in *Proc. ICCV*, Oct. 2019, pp. 3879–3888.

[20] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba, "The sound of pixels," in *Proc. ECCV*, 2018, pp. 570–586.

[21] N. Zhang, J. Yan, and Y. Zhou, "Weakly supervised audio source separation via spectrum energy preserved Wasserstein learning," in *Proc. IJCAI*, 2018, pp. 4574–4580.

[22] D. Stoller, S. Ewert, and S. Dixon, "Adversarial semi-supervised audio source separation applied to singing voice extraction," in *Proc. ICASSP*, Apr. 2018, pp. 2391–2395.

[23] D. Stowell and R. E. Turner, "Denoising without access to clean data using a partitioned autoencoder," *arXiv preprint arXiv:1509.05982*, 2015.

[24] M. Michelashvili, S. Benaim, and L. Wolf, "Semi-supervised monaural singing voice separation with a masking network trained on synthetic mixtures," in *Proc. ICASSP*, May 2019.

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

[26] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, pp. 2121–2133, 2010.

[27] L. Su and Y.-H. Yang, "Escaping from the abyss of manual annotation: New methodology of building polyphonic datasets for automatic music transcription," in *Proc. CMMR*, 2015, pp. 309–321.

[28] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Learning features of music from scratch," in *Proc. ICLR*, 2017.

[29] 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. WASPAA*, Oct. 2019.

[30] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[31] 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, 2019.

[32] C. Raffel and D. P. Ellis, "Intuitive analysis, creation and manipulation of midi data with pretty midi," in *ISMIR Late Breaking and Demo Papers*, 2014, pp. 84–93.

[33] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR–half-baked or well done?" in *Proc. ICASSP*, May 2019, pp. 626–630.

[34] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "mir_eval: A transparent implementation of common mir metrics," in *Proc. ISMIR*, vol. 15, Oct. 2014, pp. 367–372.