USER-GUIDED ONE-SHOT DEEP MODEL ADAPTATION FOR MUSIC SOURCE SEPARATION

Giorgia Cantisani,[1,2*] *Alexey Ozerov,*[2] *Slim Essid,*[1] *Gaël Richard*[1]

[1]LTCI, Télécom Paris, Institut Polytechnique de Paris, 91120, Palaiseau, France
firstname.lastname@telecom-paristech.fr
[2]InterDigital R&D France, 35510, Cesson Sevigne, France
firstname.lastname@interdigital.com

ABSTRACT

Music source separation is the task of isolating individual instruments which are mixed in a musical piece. This task is particularly challenging, and even state-of-the-art models can hardly generalize to unseen test data. Nevertheless, prior knowledge about individual sources can be used to better adapt a generic source separation model to the observed signal. In this work, we propose to exploit a temporal segmentation provided by the user, that indicates when each instrument is active, in order to fine-tune a pre-trained deep model for source separation and adapt it to one specific mixture. This paradigm can be referred to as *user-guided one-shot deep model adaptation for music source separation*, as the adaptation acts on the target song instance only. Our results are promising and show that state-of-the-art source separation models have large margins of improvement especially for those instruments which are underrepresented in the training data.

Index Terms— Music Source Separation, One-shot Domain Adaptation, User-guided Source Separation

1. INTRODUCTION

The ultimate goal of source separation is to isolate individual sound sources in a mixture of multiple sounds. In the case of music, this translates into isolating individual instruments such as voice, bass, drums, and any other accompaniments which are mixed in a musical piece. Mathematically, one can assume that the mixture signal y_n at sample n is a linear mixture of I sources $s_{i,n}$ such as: $y_n = \sum_{i=1}^{I} s_{i,n}$. Given only y_n, the goal of a source separation system is to recover one or more sources $s_{i,n}$, where $i \in \{1, ..., I\}$. Usually, a song is not a linear sum of sources because there is a mastering step, which may include the application of multiple non-linear transformations and audio effects. Another factor that makes music separation a challenging problem is the fact that musical sources are highly correlated, both in frequency and time.

Most current state-of-the-art approaches are based on deep neural networks trained in a fully supervised fashion [1–4]. Such approaches require large datasets of mixtures and corresponding isolated sources for training and it is hard for them to generalize to unseen test data with significant timbre variation compared to the training data.

To mitigate this issue, one can inform the separation process with any prior knowledge one may have about the sources and the mixing process [5]. For instance, there are works which use additional information such as the score [6], pitch [7], lyrics [8], the motion of the sound sources and visual cues [9]. One of the most underrated and powerful additional modalities is the user feedback which may leverage significant human expertise [10–14]. Particularly prolific was the use of time annotations provided by the user to learn source separation systems based on non-negative matrix factorization (NMF) or non-negative tensor factorization (NTF) [10–13]. In deep learning-based systems, time activations have already been used in multi-task learning paradigms where the source separation and the instrument activity detection tasks are jointly optimized [15, 16]. Often, the time activations are relaxed to weak class labels, indicating a given instrument in a specific time interval, and are used as an input conditioning for the separation system [17–20].

In all these works, the model is learned using both the activations and the audio material (mixtures) to be separated. One may want, instead, to choose a powerful deep model which was trained for the source separation task only and adapt it to a specific mixture using the time activations provided by the user. This is the case, for instance, of a sound engineer who can provide many priors about a mixture of interest and use them to optimize the separation system.

Within this work, we propose a *user-guided one-shot deep model adaptation for music source separation*, where the user's temporal segmentation is used to adapt a pre-trained deep source separation model to one specific test mixture. The adaptation is made possible thanks to a proposed loss function which aims to minimize the energy of the silent sources while at the same time forcing the perfect reconstruction of the mixture. We underline that the adaptation is one-shot, as it acts on the target song instance only and not on a new dataset as most fine-tuning strategies do.

Our approach is particularly beneficial for those instruments which are under-represented in the training data. Most state-of-the-art supervised music source separation models are built to separate four classes of instruments: bass, vocals, drums and "other" [1, 4]. The class "other" contains one or more instruments in the mixture that are not bass, vocals or drums (piano, strings, brass or even electronic sounds). This class has a much broader variability in timbre and pitch range than the three other single-instrument classes. We show that our approach has the most considerable improvement over this class, for which a non-adapted model struggles to find a common representation of such heterogeneous sounds.

The source code and audio examples are available online.[1]

[*] This work was founded by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 765068 (MIP-frontiers) and by the European Union's Horizon 2020 research and innovation program under the grant No. 951911 (AI4Media).

[1] https://adasp.telecom-paris.fr/resources/2021-06-01-ugosa-paper/

2. RELATED WORK

The idea of using time annotations directly provided by the user to inform a source separation system was already explored in many previous works [11–13]. Some of them rely on dedicated graphical user interfaces, while others are interactive, where the user can iteratively improve and correct the separation [21, 22]. Time annotations were also extended to more general time-frequency annotations [23–26]. There are also some interesting works where the user can hum [27], sing or play [28] the source he/she wants to enhance as an example to the source separation system. In the work from El Badawy *et al.* [29], the user may listen to an audio mixture and type some keywords (e.g., "dog barking", "wind") describing the sound sources to be separated. These keywords are then used as text queries to search for audio examples from the internet to guide the separation process. The user can also provide the fundamental frequency or manually correct it [14, 30]. Some other works use the neural activity of the listener to inform a source separation model [31, 32]. Most of these approaches are based on NMF or NTF. Only the work of Nakano et al. [14] is deep learning-based and is specific for music source separation. Their proposed model jointly estimates separated singing voice and its fundamental frequency F0, and the user is asked to provide a manual correction of the F0 trajectory based on which the model is adapted.

Within this work, we explore if adaptation is beneficial for deep learning-based source separation models, as nowadays, most state-of-the-art models are based on a fully data-driven approach without adaptation [1–4]. In the work of Nakano at al. [14], the model was initially trained for both singing voice separation and F0 estimation and then is adapted using the F0 loss only. In our case, instead, we are interested in a more general framework, where the deep model is trained on the source separation task only, and the activations are used solely for the adaptation. This paradigm is general since allows for adapting any deep-learning-based source separation model, using the activations of the target song instance only.

3. METHODS

The goal is to adapt a pre-trained deep model for source separation to a particular music piece using the time annotations provided by the user. To this aim, we chose a state-of-the-art music source separation model whose pre-trained weights were made available, and we study fine-tuning strategies using a new loss function we propose which makes use of the annotation provided by the user.

3.1. Model

The source separation model chosen for our experiment is ConvTasnet. This architecture was proposed for single-channel speech separation [2] and extended to multi-channel music separation in [1]. It achieves state-of-the-art results in both tasks, and this is why we have chosen this model for our experiments. ConvTasnet works in the waveform domain and is structured as three main blocks: an encoder, a separation subnetwork and a decoder (see Figure1 for further details). The encoder transforms a mixture's segments into a non-negative representation in an intermediate feature space; this representation is then used to estimate a mask for each source at each time step in the separation subnetwork; the isolated waveforms are finally reconstructed transforming the masked encoder features using the decoder.

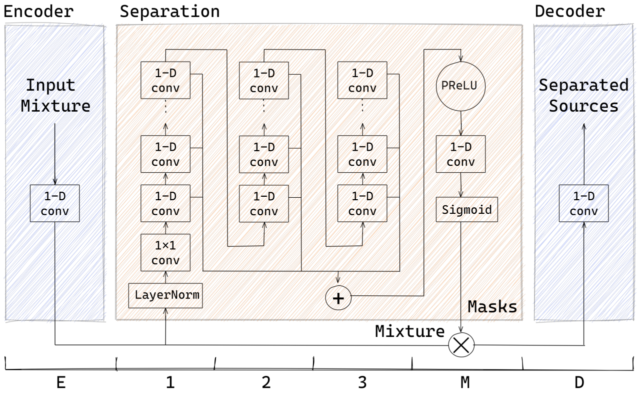

Figure 1: ConvTasnet architecture. Layer names are given below to understand the different fine-tuning strategies.

3.2. Proposed adaptation loss

In supervised training of a source separation model, the mixture is provided as input; the model outputs the estimated sources which are then compared to the original sources used to create the mixture. The difference between the estimated and the original sources is used to update the model parameters during training. Typically, an ℓ_1 or ℓ_2 loss is adopted, which respectively represents the average absolute error or average mean square error between waveforms.

In our case, during adaptation, we do not have access to the isolated sources anymore but only to their binary temporal activations. To adapt the weights of the model to the test mixture, we introduce a new loss function based on the binary activations $h_{i,n}$ (active: $h_{i,n} = 1$ / non-active: $h_{i,n} = 0$) of each instrument i at sample n. When one instrument is absent, the loss minimizes the ℓ_1-norm of its estimate while at the same time, it forces the perfect reconstruction of the mixture. Given the binary activations $h_{i,n}$ of each instrument i at time frame n, this formulation can be implemented as follows:

$$L = \frac{1}{N}\sum_{n=1}^{N}\left[\left| \sum_{i=1}^{I}(h_{i,n} \cdot \hat{s}_{i,n}) - y_n \right| + \lambda \sum_{i=1}^{I} \left|(1 - h_{i,n}) \cdot \hat{s}_{i,n}\right| \right]; \qquad (1)$$

where the total cost is composed by two terms: the first one concerns the perfect reconstruction of the mixture while the second one the energy minimization of the silent sources. If the instrument is active in a given frame n, then $h_{i,n} = 1$ and the energy minimization term is 0. On the contrary, if $h_{i,n} = 0$, then the energy of $\hat{s}_{i,n}$ is minimized. Only if the instrument is active, it will concur to the mixture reconstruction loss. λ is a hyper-parameter that weights the contribution of the energy minimization term in the total loss.

4. DATA

We use the popular MUSDB18 dataset, which consists of 150 full-length music stereo tracks of various genres sampled at 44.1 kHz. For each track, it provides a linear mixture (identical to the sum of the sources) along with the isolated tracks for the four categories: drums, bass, vocals, and others. The "others" category contains all other sources in the mix that are not the drums, bass, or vocals. In

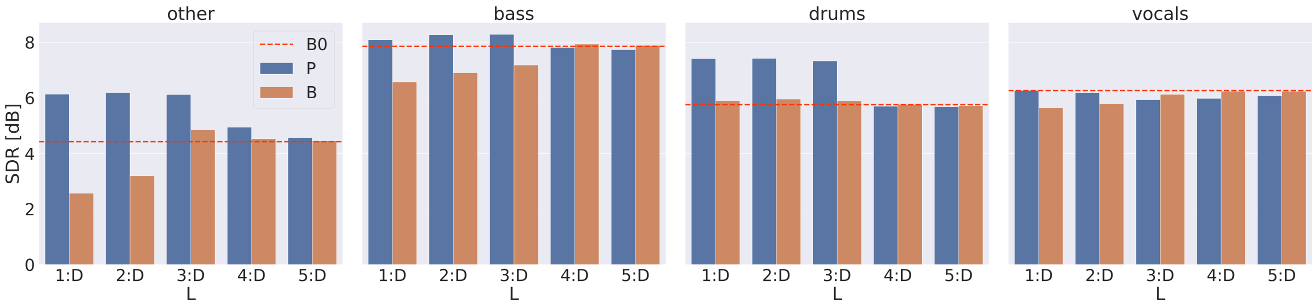

Figure 2: Median over all tracks of the median SDR (expressed in dB) over each track for different fine-tuning strategies and different instruments in the dataset. Blue bars correspond to models adapted with the proposed loss while Orange ones correspond to models adapted using a reconstruction loss only. The horizontal red line represents the B0 baseline, *i.e.*, the original ConvTasnet before adaptation.

our experiments, we use the first ten songs of the test set together with the binary temporal activations of each instrument.

To validate the proposed loss function, we decided to work in a controlled scenario: we manually set to zero each source composing a mixture for one-quarter of the song so as to have at least 25% of silence for each instrument. This procedure belongs to a data preparation step before computing the frame-wise activations. For each test mixture, the procedure is as follows:

1. segment the mixture into four segments of equal length,
2. assign each segment to one source,
3. set each source to zero in the assigned segment.

The source to segment assignment (see step 2. above) is performed randomly to avoid systematic bias. The sources are set to zero in the short-time Fourier transform (STFT) domain, so to have smooth transitions in time between silent and non-silent segments thanks to the STFT windowing.

Then, the time annotations were obtained using the same procedure and hyper-parameters used to annotate the MedleyDB dataset [33], a music dataset which provides the temporal activations of each instrument. The amplitude envelopes were generated for each source $s_{i,n}$ using a standard *envelope following* technique, consisting of half-wave rectification, compression, smoothing, and downsampling. The resulting envelope $a_{i,n}$ is then normalized to account for overall signal energy and the total number of sources in the mixture. Finally, the confidence $c_{i,n}$ of the activations $a_{i,n}$ of instrument i at time frame n can be approximated via a logistic function:

$$c_{i,n} = 1 - \frac{1}{1 + e^{\gamma(a_{i,n} - \theta)}}, \tag{2}$$

where $\gamma = 20$ controls the slope of the function, and $\theta = 0.15$ controls the threshold of activation. If $c(i,n) \geq 0.5$, then instrument i is considered active ($h_{i,n} = 1$) at time frame n. Otherwise, if $c(i,n) < 0.5$, it is considered silent ($h_{i,n} = 0$).

5. EXPERIMENTS

In this work, we considered the implementation of ConvTasnet for multi-channel music separation provided by [1]. The weights of the model pre-trained on the MUSDB18 training set were downloaded from the author's Github page.[2] For further details about the model implementation, please refer to this page. To adapt the network to each test mixture we fine-tuned it for 10 epochs on 4-second-long segments extracted from the mixture. The initial learning rate was set to 10^{-5}, batch size to 1 and Ranger was used as the optimizers.[3] Specifically, Ranger combines RAdam [34] and LookAhead [35] optimiser together. Our source code is publicly available at [4].

As mentioned above, the evaluation was performed on the first 10 test mixtures of the MUSDB18 dataset. For a fair comparison, the binary activations were applied to the outputs of all models including the baselines. The models are evaluated using standard metrics in music source separation, *i.e.* Signal-to-Distortion Ratio (SDR), Signal-to-Interference Ratio (SIR), Signal-to-Artifacts Ratio (SAR) and Interference-to-Signal Ratio (ISR) expressed in dB and computed using the BSSEval v4 [36]. As the SDR is not defined for silent frames, the evaluation is done only where the sources are non-silent. Each tested configuration is evaluated in terms of the median over all tracks of the median SDR, SIR, SAR, and ISR over each track, as done in the SiSEC Mus evaluation campaign [4].

5.1. Adaptation strategies

When adapting a deep learning model for a new task, it is often useless and counterproductive to fine-tune all the network parameters as, for example, the first layers extract some general features which might be useful also for the new task. In our case, the adaptation is not performed over a new task but over a specific instance of the test set. Thus, the task remains the same as the one for which the network was trained. Moreover, the data on which to perform the adaptation is extremely limited (just one mixture), increasing the risk of overfitting. Those factors make the choice of parameters to fine-tune critical and will largely influence the performance.

Let "P" stand for proposed while "B" stand for baseline. "Lx:y" indicates the layers that are fine-tuned (e.g., P-L2:D means that the network is fine-tuned from the second block to the last one using the proposed loss). Please refer to Figure 1 for the layer's names. We consider as the main baseline the original ConvTasnet trained on the MUSDB18 training set (B0). Moreover, for each of the proposed fine-tuning strategies, we obtain a specific baseline B-Lx:y where the model is adapted in an unsupervised manner using the mixture reconstruction loss only and ignoring the activations.

[2] https://github.com/facebookresearch/demucs

[3] https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer

[4] https://github.com/giorgiacantisani/ugosa

		other				bass				drums				vocals			
	#TP	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR
P-L1:D	8.2M	6.1	9.3	6.7	12.4	8.1	15.3	7.6	12.2	7.4	14.6	7.5	14.3	6.3	15.9	7.3	13.2
P-L2:D	5.6M	6.2	9.5	6.5	12.1	8.3	15.3	7.6	12.0	7.4	14.5	7.6	14.4	6.2	15.7	7.1	13.7
P-L3:D	2.9M	6.1	9.5	6.5	11.6	8.3	12.3	7.0	11.4	7.3	14.2	7.3	12.6	5.9	14.3	7.3	14.1
P-L4:D	0.4M	4.9	8.9	5.6	11.0	7.8	10.4	7.3	13.1	5.7	12.7	6.1	11.6	6.0	16.5	6.9	12.4
P-L5:D	0.01M	4.6	9.1	5.1	11.1	7.7	10.9	7.3	14.1	5.7	13.7	6.0	11.4	6.1	16.8	6.7	12.1
B0	-	4.4	10.0	4.5	11.5	7.9	11.2	7.4	15.5	5.8	15.4	5.9	12.1	6.3	18.9	6.7	14.2

Table 1: SDR, SIR, SAR, ISR expressed in dB: median over frames, median over tracks for different fine-tuning strategies and different instruments in the dataset. #TP stands for the number of trainable parameters which are fine-tuned during adaptation.

5.2. Hyper-parameter sensitivity

We verified the influence of the hyper-parameter λ on the performances by testing nine different values of λ ranging from 10^{-4} to 10^{4} with a logarithmic step. Those results were obtained on the P-L3:M configuration using a window length of 10 seconds. λ expresses the weight of the term that minimizes the energy of the absent sources in the total cost function. Only the vocals performances are pretty stable with respect to this parameter with no statistically significant difference in the SDR, SAR and SIR across different values of λ. For the other classes, a higher λ leads to a higher SIR, meaning that the suppression of the interferes is more aggressive. A more aggressive separation is often counterbalanced by a significant deterioration of the SAR, meaning more artefacts.

The performances are not sensitive, instead, to the length of the input segments. The results were obtained on the P-L3:M configuration with $\lambda = 1$ for different lengths of the input segments. We tested five different lengths from 2 to 10 seconds obtaining no statistically significant differences in the SDR and SAR performances except for the class "other", where, with a window below 4 seconds, the SDR and the SAR decreases. This parameter does not significantly influence the SIR except for the vocals, where it significantly decreases below 4 seconds.

5.3. Results and discussion

In Figure 2 one can see the SDR expressed in dB for different fine-tuning strategies and instruments in the dataset. Blue bars correspond to models fine-tuned with the proposed loss while orange ones correspond to models fine-tuned using the mixture reconstruction loss only. The red line represents the B0 baseline, *i.e.*, the original ConvTasnet trained on the MUSDB18 training set and not adapted at all. We can see how the SDR changes with respect to the block from which we start fine-tuning the network. It is necessary to fine-tune at least from the third block to obtain a significant improvement over the baseline B0. We have to keep in mind that fine-tuning starting from a deeper block corresponds to millions more parameters to fine-tune. If the number of such parameters is high, it requires a proportional amount of training data, which in our case is not possible, as the "adaptation" data comes from only one mixture.

The improvement over the baseline is particularly pronounced for the category "other", for which the original baseline B0 was struggling the most. As we said before, this category does not represent a specific instrument. So, it has much more variability than the other classes which are homogeneous in terms of type of instruments, and the network struggles to find a common representation for those sounds. Adaptation is then particularly useful in this situation, where we need to adapt to a specific instrument which may be different from the ones seen in the training phase. The vocals are the only instrument where we do not improve over the baseline, indicating that probably this class was already well represented in the training data, leaving small room for improvement. In general, the deeper we fine-tune, the higher the improvement of the proposed model over the corresponding baseline, showing that the activations play an active role in the adaptation and that the improvement over B0 cannot be achieved easily in a completely unsupervised fashion.

Looking at Table 1, we can have an insight into the evolution of all the metrics. The SDR improvement is mostly due to a SAR improvement, while at the same time, the SIR drops. This means that there are fewer artefacts than before the adaptation, but at the same time, the interferences are not entirely removed. The only instrument which shows a different trend is the bass, for which the SIR and SDR increase and the SAR drops. The bass is the only instrument for which the SIR improves over B0. Separating the bass often corresponds to a low-pass filter and probably the adaptation allows for better adapting the filter to the register played by the bass in the given piece of music.

Motivated by the observation that the decoder has the general function of going back from the feature to the waveform domain, two other fine-tuning strategies were experimented: one strategy where the decoder weights are frozen during fine-tuning (P-Lx:M) and one where both the decoder and masking blocks are frozen (P-Lx:3). We experimented those variants for all the fine-tuning depths and compared them to the corresponding variants where the network is fine-tuned until the last layer (P-Lx:D). The three variants' performances are not significantly different, indicating that there is no need to fine-tune the decoder or the masking blocks and giving us an insight into the network functionality.

6. CONCLUSION

In this work we proposed a *user-guided one-shot deep model adaptation for music source separation*, where the temporal segmentation provided by the user is used to adapt a pre-trained deep source separation model to one specific test mixture. The adaptation is made possible thanks to a newly proposed loss function which aims to minimize the energy of the silent instruments while at the same time forcing the perfect reconstruction of the mixture. Our results are promising and show that state-of-the-art source separation models may be significantly improved via adaptation with a small number of epochs to the specific test mixture. We show that the improvement is particularly remarkable for those instruments which are underrepresented in the training data. We underline that the proposed approach is general and can be applied to other types of audio sources (speech, natural sounds) or other deep model architectures.

7. REFERENCES

[1] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv preprint:1911.13254*, 2019.

[2] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM TASLP*, 2019.

[3] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-unmix-a reference implementation for music source separation," *Journal of Open Source Software*, 2019.

[4] F.-R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *Int. Conf. on LVA/ICA*. Springer, 2018.

[5] A. Liutkus, J.-L. Durrieu, L. Daudet, and G. Richard, "An overview of informed audio source separation," in *IEEE WIAMIS*, 2013.

[6] S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley, "Score-informed source separation for musical audio recordings: An overview," *IEEE Signal Process. Mag.*, 2014.

[7] T. Virtanen, A. Mesaros, and M. Ryynänen, "Combining pitch-based inference and non-negative spectrogram factorization in separating vocals from polyphonic music." in *SAPA@ INTERSPEECH*, 2008.

[8] K. Schulze-Forster, C. Doire, G. Richard, and R. Badeau, "Weakly informed audio source separation," in *IEEE WASPAA*, 2019.

[9] S. Parekh, S. Essid, A. Ozerov, N. Q. Duong, P. Pérez, and G. Richard, "Guiding audio source separation by video object information," in *IEEE WASPAA*, 2017.

[10] M.-Q. Bui, V.-H. Duong, S.-P. Tseng, Z.-Z. Hong, B.-C. Chen, Z.-W. Zhong, and J.-C. Wang, "NMF/NTF-based methods applied for user-guided audio source separation: An overview," in *IEEE ICOT*, 2016.

[11] H. Laurberg, M. N. Schmidt, M. G. Christensen, and S. H. Jensen, "Structured non-negative matrix factorization with sparsity patterns," in *IEEE 42nd Asilomar Conf. on Signals, Systems and Computers*, 2008.

[12] A. Ozerov, C. Févotte, R. Blouet, and J.-L. Durrieu, "Multichannel nonnegative tensor factorization with structured constraints for user-guided audio source separation," in *IEEE ICASSP*, 2011.

[13] N. Q. Duong, A. Ozerov, and L. Chevallier, "Temporal annotation-based audio source separation using weighted nonnegative matrix factorization," in *IEEE ICCE-Berlin*, 2014.

[14] T. Nakano, Y. Koyama, M. Hamasaki, and M. Goto, "Interactive deep singing-voice separation based on human-in-the-loop adaptation," in *Proc. 25th Int. Conf. IUI*, 2020.

[15] D. Stoller, S. Ewert, and S. Dixon, "Jointly detecting and separating singing voice: A multi-task approach," in *Int. Conf. on LVA/ICA*. Springer, 2018.

[16] Y.-N. Hung and A. Lerch, "Multitask learning for instrument activation aware music source separation," *arXiv preprint:2008.00616*, 2020.

[17] R. V. Swaminathan and A. Lerch, "Improving singing voice separation using attribute-aware deep network," in *IEEE MMRP*, 2019.

[18] O. Slizovskaia, L. Kim, G. Haro, and E. Gomez, "End-to-end sound source separation conditioned on instrument labels," in *IEEE ICASSP*, 2019.

[19] P. Seetharaman, G. Wichern, S. Venkataramani, and J. Le Roux, "Class-conditional embeddings for music source separation," in *IEEE ICASSP*, 2019.

[20] E. Karamatlı, A. T. Cemgil, and S. Kırbız, "Audio source separation using variational autoencoders and weak class supervision," *IEEE Signal Process. Lett.*, 2019.

[21] N. Bryan and G. Mysore, "An efficient posterior regularized latent variable model for interactive sound source separation," in *ICML*, 2013.

[22] N. Q. Duong, A. Ozerov, L. Chevallier, and J. Sirot, "An interactive audio source separation framework based on nonnegative matrix factorization," in *IEEE ICASSP*, 2014.

[23] A. Lefevre, F. Bach, and C. Févotte, "Semi-supervised NMF with time-frequency annotations for single-channel source separation," in *ISMIR*, 2012.

[24] A. Lefèvre, F. Glineur, and P.-A. Absil, "A convex formulation for informed source separation in the single channel setting," *Neurocomputing*, 2014.

[25] I.-Y. Jeong and K. Lee, "Informed source separation from monaural music with limited binary time-frequency annotation," in *IEEE ICASSP*, 2015.

[26] Z. Rafii, A. Liutkus, and B. Pardo, "A simple user interface system for recovering patterns repeating in time and frequency in mixtures of sounds," in *IEEE ICASSP*, 2015.

[27] P. Smaragdis and G. J. Mysore, "Separation by "humming": User-guided sound extraction from monophonic mixtures," in *IEEE WASPAA*, 2009.

[28] D. FitzGerald, "User assisted separation using tensor factorisations," in *Proc. 20th EUSIPCO*, 2012.

[29] D. El Badawy, N. Q. Duong, and A. Ozerov, "On-the-fly audio source separation," in *IEEE MLSP*, 2014.

[30] J.-L. Durrieu and J.-P. Thiran, "Musical audio source separation based on user-selected f0 track," in *Int. Conf. LVA/ICA*. Springer, 2012.

[31] E. Ceolini, J. Hjortkjær, D. D. Wong, J. O'Sullivan, V. S. Raghavan, J. Herrero, A. D. Mehta, S.-C. Liu, and N. Mesgarani, "Brain-informed speech separation (BISS) for enhancement of target speaker in multitalker speech perception," *NeuroImage*, 2020.

[32] G. Cantisani, S. Essid, and G. Richard, "Neuro-steered music source separation with eeg-based auditory attention decoding and contrastive-nmf," in *IEEE ICASSP*, 2021.

[33] 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 *ISMIR*, 2014.

[34] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint:1908.03265*, 2019.

[35] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, "Lookahead optimizer: k steps forward, 1 step back," in *Advances in Neural Information Processing Systems*, 2019.

[36] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE TASLP*, 2006.