A frugal approach to music source separation

Emery Pierson Lancaster, Nathan Souviraà-Labastie

A FRUGAL APPROACH TO MUSIC SOURCE SEPARATION

Emery Pierson Lancaster[*]

IMT Lille Douai, Univ. Lille, CNRS,
UMR 9189 CRIStAL, Lille, France

Nathan Souviraà-Labastie[‡]

A-Volute
Lille, France

ABSTRACT

During the past years, deep learning brought a big step in performance of music source separation algorithms. A lot has been done on the architecture optimisation, but training data remains an important bias for model comparison. In this work, we choose to work with the frugal and well-known original TasNet neural network and to focus on simple methods to exploit a relatively important dataset. Our results on the MUSDB test set outperform all previous state of the art approaches with extra data on the following source categories: vocals, accompaniment, drums, bass and in average. We believe that our results on how to shape a training set can apply to any type of architecture.

Index Terms— Audio separation, audio deep learning, audio database, music source separation, benchmark

1. INTRODUCTION

Deep learning for audio source separation is a quite recent trend. In music, neural networks techniques brought significant improvements through the two last SiSEC challenge [1, 2] and many approaches have been proposed in the past few years [3–12]. From the first neural network architectures [3] until recently, almost all proposed neural networks were using the results of the Fourier transform as input representation (and expected output). A significant gap has been made with the release of TasNet [6, 10] which was firstly designed for speech separation. Conversely to frequency based approaches, TasNet is an end-to-end architecture, also called temporal or time-domain approach. It takes as input the raw audio frames and its input representation is learned by the first layers (encoder) instead of being deterministic like the Fourier transform. This TasNet or encoder-separator-decoder paradigm has gather a lot of attention recently [11–14]. However in [14], the authors shows that we should not be categorical on using a learned encoder for speech separation.

For music separation, the reference benchmark stays today the MUS task of the SiSEC challenge based on the MUSDB18 dataset [15]. Unfortunately, this dataset contains only 150 songs (with only 100 songs in the training part) for a total of approximately 10 hours of data. This size is quite limiting for deep learning approaches whose strength is to make the most of large training sets, and more and more approaches are using extra data (in addition to the MUSDB18 corpus) for the training. As a consequence, there is in fact two different tasks to address: building (music) source separation model with scarce training set and building the best model possible. This is well highlighted by the paperswithcode website[1], although it does not list paper without code like MMDenseLSTM [5].

MMDenseLSTM [5], Spleeter [8], Demucs [9] and Conv-TasNet [10] (benchmarked in [9]) yield the best results so far, respectively using in the order of 8, 200, 2 and 2 times more songs[2] than the MUSDB18 training set. One can remember two points : except Spleeter, all these approaches are highly demanding in term of hardware learning resources ; over all approaches, the performance difference for a given architecture between with and without extra data is relatively small (between 0.3 and 0.7 db SDR improvement in average).

In this work, we explore how to make the most of a large collection of extra data but with a not so demanding model in terms of hardware. We start by presenting the experimental setup along with the state of the art TasNet-LSTM architectures, *i.e.*, original TasNet, we then show the different corpus configurations used as training set. After discussing on the obtained results, we conclude with a series of recommendation on how to exploit large datasets.

2. EXPERIMENTAL SETUP

In this section, we present the chosen architecture for this paper, and the common configuration for training.

2.1. Original TasNet

The Encoder-Separator-Decoder architectures approach has been popularised by TasNet [6]. The authors first intention was to explore if the STFT representation was necessary in source separation problems, and if one can reach better results with a learned representation. It has shown significant

[*]Work done during master internship at A-Volute

[‡]Contact: nathan.souviraa-labastie@a-volute.com

[1]https://paperswithcode.com/sota/music-source-separation-on-musdb18

[2]we did not take into account data augmentation on purpose

results for either speech separation, dereverberation and denoising. Even if it has yet not been used for music source separation, its simple architecture allows a large use with low GPU RAM usage, and can be used as a simple baseline.

In the original TasNet [6], the STFT is replaced by 1D convolutional layers. The results is then provided to the separator which is powered by the well known LSTM temporal network in the case of the original TasNet, here used in bidirectional mode. More information can be found in Section 2.4 or in [6, 16].

2.2. Discarded architectures

Even if we have selected a specific type of architecture for our paper, many recent architecture could have been included in our work. However, they would have come with some downside. Approaches using DenseNets [5], even if they bring greater performances were discarded due to computation costs. Conv-TasNet [10] were also ignored due to the high numbers of residual connection in the TCN. Demucs [9] doesn't have those defaults, but too much computation is required to "encode" the data with the 6 convolutional layers. The computational costs are not a problem for either U-Net [17] architectures nor Spleeter [8], but we choose for a more up-to-date and modular approach such as TasNet.

The best option would have be to use the Dual Path RNN (DPRNN) [12], a more recent improvement of the RNN layer, *i.e.* the separator, of TasNet. Unfortunately, our experiments were already running and were taking months due to poor GPU.

2.3. Data

2.3.1. MUSDB18

MUSDB18 [15] is a database composed of 150 songs (totalling around 10 hours of mixture). It is provided under the stems format which makes available the ground truth for `vocals`, `bass`, `drums` and `other`. The `accompaniment` source is the counterpart of the `vocals` source. It was released for the SiSEC18 challenge MUS task and is an improvement of an earlier dataset.

2.3.2. Extra-data

We collected online individual instrument recordings which we semi-automatically assigned to each source (firstly vocals and accompaniment and secondly vocals, drums, bass and other) in order to obtain a total of approximately 300 hours of supervised data. If we consider that a song is in average 4 minutes long, we estimate that our full dataset is 20 times bigger than the data used to train Demucs (150+84 songs [9]), 5 times bigger than Sony dataset (800 songs [5]) but 5 times smaller than the Bean dataset from Deezer used to train Spleeter (24097 songs [8]).

2.3.3. No data augmentation

Usually, with relatively low database sizes, various augmentation techniques are used, in particular overlapping excerpts [9] but many other possibilities exist [18]. Although, some are relevant, we believe that many of them are only efficient when there is a lack of data. For instance, switching channels doesn't represent efficiently real music, it is for the network a harder problem to solve to not be able to use correlations between channels. Also, overlapping excerpts can cause overfitting, and would imply too much training time as the used algorithm and framework is already reaching convergence after more than a month. We decide then to restrict data to non overlapping music excerpts, and to not test any data augmentation in this paper.

2.4. Setup

The used computer has a single NVIDIA Titan X GTX GPU (12 GB VRAM). The code was implemented in Pytorch [19]. Adam algorithm [20] is used to optimize the loss function presented in the next section. During training, the learning rate is halved if the validation loss does not improve after three consecutive epochs. An early stopping mechanism is also used if no improvement is observed for more than ten consecutive epochs. The hyperparameter values are fixed for all experiments and are as follow : Learning Rate : 0.001 ; Gradient Clipping : 5 ; LSTM layers : 4 ; LSTM cells per layer : 500 ; Number of filters in the encoder : 500 ; Frame size : 220 samples ; Sampling rate : 44100 Hz.

2.5. Loss function and metrics

The training metric is the now widely used SI-SNR [6, 10–12, 21]. Given the estimated time-domain source $\hat{s}$ and the clean source s, it is computed as follow:

$$s_{target} = \frac{\langle s, \hat{s} \rangle}{||s||^2} s$$

$$e_{noise} = \hat{s} - s_{target}$$

$$SI - SNR(s, \hat{s}) = log_{10} \left(\frac{||s_{target}||^2}{||e_{noise}||^2} \right)$$

It has shown good performance either for speech separation or music separation. It is also used for evaluation on our test set (a small subset of data described in Section 2.3.2).

For evaluation on the MUSDB18 test set (see Section 2.3.1), we use the Signal-to-Distortion-Ratio (SDR) from the BSS-Eval toolbox [22] The SDR is the reference metric for evaluating the impact of the overall distortion caused by source separation algorithm. It can be separated in 2 metrics: SIR and SAR, that mesure the presence of 2 types of distortion: Interferences and Artifacts.

3. EXPERIMENT

In this section, we show our results along with different variations of our training sets. Section 3.1 to 3.4 first focus on vocals versus accompaniment separation while 3.5 presents results on the MUSDB test set for four types of sources. Two experiments, denoted * and † will be present several time in the result tables.

3.1. Preliminary experiments

In order to adjust the learning hyper-parameters, a small subset of data (10% of the total size) and relatively small audio excerpts (15 seconds per song) are used. This choice allows us to reach learning convergence in less than a day on our GPU, hence enabling us to iterate until finding the values given in Section 2.4.

The same preliminary series of experiments leads us to observe that adding to the training set a second audio excerpt from the same song is not leading to improvement in the learning. The loss after the same number of epoch is better but each epoch last twice longer and in fact the losses after the same amount of learning time are worse when adding a second audio excerpt from the same song. Conversely, adding audio excerpts from other songs leads to small improvement in the learning for the same amount of learning time.

3.2. Effect of additional data (*i.e.*, the data variety)

Based on this first observation, we carry a series of training with variable database size. The audio excerpt length are 20 seconds long for these experiments. The SI-SNR on the test set after 125 epochs are displayed in Table 1. Without surprise, the more data fed to the network the better the separation performance are.

Database (%)	`Vocals`	`Accomp.`	Mean
2	-3.19	-7.93	-5.51
10	-4.61	-10.52	-7.57
50	-6.90	-12.65	-9.77
100 *	-10.33	-13.40	-10.78

Table 1. SI-SNR measured on the test set for different size of the training set.

However, one can notice that while the learning speed is rather similar at the beginning of the learning (as first observed in Section 3.1), experiments with smaller training set reach stopping criteria earlier than with the full training set. Hence, when training is done until stopping criteria for all these experiments, this leads to differences in separation performances even bigger than those displayed in Table 1.

3.3. Effect of the audio excerpt length (*i.e.*, the context)

We run a second set of experiments on the full training set with varying audio excerpt length. With in mind to reduce the bias of having a variable total amount of data seen during the learning, excerpt from the same songs are taken. Finally, With the constrain of a relatively small GPU-VRAM size, we limited the maximum of 30 seconds for the audio excerpt as longer excerpts would lead to reduce the batch size and compromise the learning stability. The results presented in Table 2 are obtain after 50 epochs on the full training set. As one can expect, the longer the excerpt lengths are the better the separation performances are.

Excerpt length (sec)	`Vocals`	`Accomp.`	Mean
5	-3.29	-13.54	-8.42
10	-4.95	-12.72	-8.83
20 *	-6.18	-12.36	-9.27
30 †	-8.11	-12.44	-10.27

Table 2. SI-SNR measured on the test set for different length of audio excerpt in the training set.

3.4. Data variety versus context

In a more formal experiments than observations from Section 3.1, different training sets are shaped as followed. Only one excerpt is taken from each song. Each training sets use different number of songs. The audio excerpt lengths presented to the network are adapted so that the different training sets have equivalent total amounts of audio. The sizes of the batches are also adapted so that the amount of audio per batch is the same from one training set to another. Tested audio excerpt lengths for this experiment are 5, 10, 15, 20, 30 and 40 seconds.

Regarding accompaniment, all experiments lead to similar performance separations. Regarding the vocals, performance separation for higher than 15 seconds excerpts are equivalent while for inferior excerpt lengths we observe clearly worst separation results. This source specific behavior is also observed in experiments described in Section 3.3.

While we can not conclude on whether the variety is more important than the context, we can keep in mind that excerpt length should be at least 20 seconds for efficient learning. As the vocals could be absent from the mixture for longer than the accompaniment, we can formulate the hypothesis that this might cause instability during the training for batches where vocals are greatly absent.

3.5. A new state of the art on the MUSDB18 test set with extra data

While previous experiments were only targeting the separation of `vocals` versus the `accompaniment`, we here train

Approaches	Extra data (ratio to MUSDB)	Vocals	Accomp.	Drums	Bass	Other	Mean
Conv-TasNet [9, 10]	2	6.74		7.11	7.00	4.44	6.32
Spleeter [8]	200	6.86	12.54	6.7	5.51	4.02	5.77
Demucs [9]	2	7.05		7.08	6.70	4.47	6.32
DenseNet (TAK2) [5]	8	7.16	13.73	6.81	5.40	**4.80**	6.05
D3Net [23]	1 (no extra data)	7.24	13.52	7.01	5.25	4.53	6.01
TasNet * [6] {1}	30	**7.34**	**13.76**	**7.68**	**7.04**	4.04	**6.52**
TasNet * [6] {4}	30	**7.39**		6.90	**7.33**	4.04	**6.42**
TasNet † [6] {1}	30	**7.43**	13.66				

Table 3. SDR measured on MUSDB18 test set (SiSEC challenge data). The results presented in this paper are noted * and †.

two additional networks to complete the * experiment, one to separate the drums source from the rest and one to separate the bass source from the rest. While vocals models (* and †) are trained until performances reach a plateau (around 300 epochs, nearly two months), bass and drums models were only trained for 30 epochs due to a lack of time and computational power.

As a final comparison, we evaluate all those models on the MUSDB18 test set. The accompaniment source is obtain as output of the same model that predict the vocals. Conversely, the other source is obtain by subtracting the temporal estimates of the vocals, bass and drums sources to the mixture, which is not ideal but enables us to quickly compute a mean SDR and compare with the state of the art. The four sources were evaluated in two different ways, firstly sources by sources denoted $\{1\}$ and secondly all four sources together denoted $\{4\}$. The results for MMDenseLSTM [5], Demucs [9] and D3Net [23] are directly taken from the corresponding papers. For Spleeter [8], we used the pretrained model publicly available and run the evaluation on the MUSDB18 test set. All results are gathered in the Table 3. Scores higher than the state of the art are displayed in bold. Except for the other source, the TasNet models learned on our dataset outperform all previous state of the art on all following categories : vocals, accompaniment, bass, drums and average SDR.

3.6. Subjective assessment

We did not proceed to any formal subjective evaluation of the results, but at least three audio experts listened to the audio results of the different experiments (experiments of Section 3.2, 3.3) and 3.5 without noticing any behavior drifting from the objective scores. As the same model is used along the all experiments, this is not surprising. Audio results and MUSDB json files are available upon request.

However, one can noticed that Spleeter results are slightly different from all other approaches regarding the balance between artefact and interference: it introduces more artefacts as usual but less interferences. This was easy to hear and also well depicted by the SAR and SIR scores.

4. DISCUSSION

Even if our results reaches state of the art, reader must remind that Table 3 stays a comparison of full training pipelines - including extra data - rather than a deep learning architectures comparison. Our work confirms that finding a tradeoff between performances with a scarce training set, scalability on a massive training set and frugal model (in order to scan the all training set) is of importance. While there is still an unbalanced research effort between model and data, we think that the training data bias reduction initiated in [7, 15, 23] is part of the right path. Prelimary experiments on the same corpus with other architectures [16, 24] also lead us to comparable performance, *i.e.*, over all previous state of the art approaches. This strengthen our assumption that for now the training dataset is at least as important as the model complexity.

Regarding instruments separation (drum and bass), most approaches are usually struggling. Architectures such as Demucs [9] propose a new encoder with a larger receptive field targeting these sources but without significant improvement. While it seems that training on our database greatly improve the performances, we also believe that architectures like TasNet do not favorize extraction of such sounds, essentially due to the small size of the encoder (hence the receptive field of the network). One interesting path could be changing the learned encoder for a fixed one (as in [14] (fixed filterbank) or in [25] (STFT)) adapted to music instruments. Using a FiLM layer [26] is also an interesting idea.

5. CONCLUSION

In this paper on music separation, we first depict the importance of audio excerpt length as well as the use of extra data during the training. We also show that a really simple model like the original TasNet outperforms all previous proposed models like MMDenseLSTM, Spleeter or Demucs (trained with extra data). On the MUSDB18 test set (SiSEC challenge data), reported SDR show improvement for most source types and in average. This indicates that for the music source separation problem data is for now at least as important as well designed neural network architectures. This also indicates that future model comparison should definitely avoid the bias of extra and/or different training data.

6. REFERENCES

[1] Antoine Liutkus, Fabian-Robert Stöter, Zafar Rafii, Daichi Kitamura, Bertrand Rivet, Nobutaka Ito, Nobutaka Ono, and Julie Fontecave, "The 2016 signal separation evaluation campaign," in *International conference on latent variable analysis and signal separation*. Springer, 2017, pp. 323–332.

[2] Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito, "The 2018 signal separation evaluation campaign," in *Latent Variable Analysis and Signal Separation*, 2018, pp. 293–30.

[3] Aditya Arie Nugraha, Antoine Liutkus, and Emmanuel Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1652–1664, 2016.

[4] Daniel Stoller, Sebastian Ewert, and Simon Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," *CoRR*, vol. abs/1806.03185, 2018.

[5] Naoya Takahashi, Nabarun Goswami, and Yuki Mitsufuji, "Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation," 05 2018.

[6] Yi Luo and Nima Mesgarani, "Tasnet: time-domain audio separation network for real-time, single-channel speech separation," *CoRR*, vol. abs/1711.00541, 2017.

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

[8] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, pp. 2154, 2020, Deezer Research.

[9] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach, "Music Source Separation in the Waveform Domain," working paper or preprint, Nov. 2019.

[10] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[11] David Samuel, Aditya Ganeshan, and Jason Naradowsky, "Meta-learning extractors for music source separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 816–820.

[12] Yi Luo, Zhuo Chen, and Takuya Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," 2020.

[13] M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "Filterbank design for end-to-end speech separation," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6364–6368.

[14] D. Ditter and T. Gerkmann, "A multi-phase gammatone filterbank for speech separation via tasnet," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 36–40.

[15] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner, "The MUSDB18 corpus for music separation," Dec. 2017.

[16] Ismail Alaoui Abdellaoui and Nathan Souviraa-Labastie, "Blending the attention mechanism in tasnet," 2020.

[17] A. Jansson, R. M. Bittner, S. Ewert, and T. Weyde, "Joint singing voice separation and f0 estimation with deep u-net architectures," in *2019 27th European Signal Processing Conference (EUSIPCO)*, 2019, pp. 1–5.

[18] L. Prétet, R. Hennequin, J. Royo-Letelier, and A. Vaglio, "Singing voice separation: A study on training data," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 506–510.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.

[20] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] Manuel Pariente, Samuele Cornell, Joris Cosentino, Sunit Sivasankaran, Efthymios Tzinis, Jens Heitkaemper, Michel Olvera, Fabian-Robert Stöter, Mathieu Hu, Juan M. Martín-Doñas, David Ditter, Ariel Frank, Antoine Deleforge, and Emmanuel Vincent, "Asteroid: the PyTorch-based audio source separation toolkit for researchers," *arXiv preprint arXiv:2005.04132*, 2020.

[22] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[23] Naoya Takahashi and Yuki Mitsufuji, "D3net: Densely connected multidilated densenet for music source separation," *arXiv preprint arXiv:2010.01733*, 2020.

[24] Yi Luo, Zhuo Chen, and Takuya Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 46–50.

[25] Ilya Kavalerov, Scott Wisdom, Hakan Erdogan, Brian Patton, Kevin Wilson, Jonathan Le Roux, and John R Hershey, "Universal sound separation," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 175–179.

[26] Gabriel Meseguer-Brocal and Geoffroy Peeters, "Conditioned-u-net: Introducing a control mechanism in the u-net for multiple source separations," *arXiv preprint arXiv:1907.01277*, 2019.