ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096956

HYBRID TRANSFORMERS FOR MUSIC SOURCE SEPARATION

Simon Rouard, Francisco Massa, Alexandre Défossez

Meta AI

ABSTRACT

A natural question arising in Music Source Separation (MSS) is whether long range contextual information is useful, or whether local acoustic features are sufficient. In other fields, attention based Transformers [1] have shown their ability to integrate information over long sequences. In this work, we introduce Hybrid Transformer Demucs (HT Demucs), an hybrid temporal/spectral bi-U-Net based on Hybrid Demucs [2], where the innermost layers are replaced by a cross-domain Transformer Encoder, using self-attention within one domain, and cross-attention across domains. While it performs poorly when trained only on MUSDB [3], we show that it outperforms Hybrid Demucs (trained on the same data) by 0.45 dB of SDR when using 800 extra training songs. Using sparse attention kernels to extend its receptive field, and per source fine-tuning, we achieve state-of-the-art results on MUSDB with extra training data, with 9.20 dB of SDR.

Index Terms— Music Source Separation, Transformers

1. INTRODUCTION

Since the 2015 Signal Separation Evaluation Campaign (SiSEC) [4], the community of MSS has mostly focused on the task of training supervised models to separate songs into 4 stems: drums, bass, vocals and other (all the other instruments). The reference dataset that is used to benchmark MSS is MUSDB18 [3, 5] which is made of 150 songs in two versions (HQ and non-HQ). Its training set is composed of 87 songs, a relatively small corpus compared with other deep learning based tasks, where Transformer [1] based architectures have seen widespread success and adoption, such as vision [6, 7] or natural language tasks [8]. Source separation is a task where having a short context or a long context as input both make sense. Conv-Tasnet [9] uses about one second of context to perform the separation, using only local acoustic features. On the other hand, Demucs [10] can use up to 10 seconds of context, which can help to resolve ambiguities in the input. In the present work, we aim at studying how Transformer architectures can help leverage this context, and what amount of data is required to train them.

We first present in Section 3 a novel architecture, *Hybrid Transformer Demucs* (HT Demucs), which replaces the innermost layers of the original Hybrid Demucs architecture [2] with Transformer layers, applied both in the time and spectral representation, using self-attention within one domain, and cross-attention across domains. As Transformers are usually data hungry, we leverage an internal dataset composed of 800 songs on top of the MUSDB dataset, described in Section 4.

Our second contribution is to evaluate extensively this new architecture in Section 5, with various settings (depth, number of channels, context length, augmentations etc.). We show in particular that it improves over the baseline Hybrid Demucs architecture (retrained on the same data) by 0.35 dB.

Finally, we experiment with increasing the context duration using sparse kernels based with Locally Sensitive Hashing to overcome memory issues during training, and fine-tuning procedure, thus achieving a final SDR of 9.20 dB on the test set of MUSDB.

We release the training code, pre-trained models, and samples on our github facebookresearch/demucs.

2. RELATED WORK

A traditional split for MSS methods is between spectrogram based and waveform based models. The former includes models like Open-Unmix [11], a biLSTM with fully connected that predicts a mask on the input spectrogram or D3Net [12] which uses dilated convolutional blocks with dense connections. More recently, using complex-spectrogram as input and output was favored [13] as it provides a richer representation and removes the topline given by the Ideal-Ratio-Mask. The latest spectrogram model, Band-Split RNN [14], combines this idea, along with multiple dual-path RNNs [15], each acting in carefully crafted frequency band. It currently achieves the state-of-the-art on MUSDB with 8.9 dB. Waveform based models started with Wave-U-Net [16], which served as the basis for Demucs [10], a time domain U-Net with a bi-LSTM between the encoder and decoder. Around the same time, Conv-TasNet showed competitive results [9, 10] using residual dilated convolution blocks to predict a mask over a learnt representation. Finally, a recent trend has been to use both temporal and spectral domains, either through model blending, like KUIELAB-MDX-Net [17], or using a bi-U-Net structure with a shared backbone as Hybrid Demucs [2]. Hybrid Demucs was the first ranked architecture at the latest MDX MSS Competition [18], although it is now surpassed by Band-Split RNN.

Table 1: Comparison with baselines on the test set of MUSDB HQ (methods with a * are reported on the non HQ version). “Extra?” indicates the number of extra songs used at train time, † indicates that only mixes are used. “fine tuned” indices per source fine-tuning.

		Test SDR in dB				
Architecture	**Extra?**	`All`	`Drums`	`Bass`	`Other`	`Vocals`
IRM oracle	N/A	8.22	8.45	7.12	7.85	9.43
KUIELAB-MDX-Net [17]	✗	7.54	7.33	7.86	5.95	9.00
Hybrid Demucs [2]	✗	7.64	8.12	8.43	5.65	8.35
Band-Split RNN [14]	✗	**8.24**	9.01	7.22	6.70	10.01
HT Demucs	✗	7.52	7.94	8.48	5.72	7.93
Spleeter* [19]	25k	5.91	6.71	5.51	4.55	6.86
D3Net* [12]	1.5k	6.68	7.36	6.20	5.37	7.80
Demucs v2* [10]	150	6.79	7.58	7.60	4.69	7.29
Hybrid Demucs [2]	800	8.34	9.31	9.13	6.18	8.75
Band-Split RNN [14]	1750†	8.97	10.15	8.16	**7.08**	**10.47**
HT Demucs	150	8.49	9.51	9.76	6.13	8.56
HT Demucs	800	8.80	10.05	9.78	6.42	8.93
HT Demucs (fine tuned)	800	9.00	10.08	10.39	6.32	9.20
Sparse HT Demucs (fine tuned)	800	**9.20**	**10.83**	**10.47**	6.41	9.37

Using large datasets has been shown to be beneficial to the task of MSS. Spleeter [19] is a spectrogram masking U-Net architecture trained on 25,000 songs extracts of 30 seconds, and was at the time of its release, the best model available. Both D3Net and Demucs highly benefited from using extra training data, while still offering strong performance on MUSDB only. Band-Split RNN introduced a novel unsupervised augmentation technique requiring only mixes to improve its performance by 0.7 dB of SDR.

Transformers have been used for speech source separation with SepFormer [20], which is similar to Dual-Path RNN: short range attention layers are interleaved with long range ones. However, its requires almost 11GB of memory for the forward pass for 5 seconds of audio at 8 kHz, and thus is not adequate for studying longer inputs at 44.1 kHz.

3. ARCHITECTURE

We introduce the Hybrid Transformer Demucs model, based on Hybrid Demucs [2]. The original Hybrid Demucs model is made of two U-Nets, one in the time domain (with temporal convolutions) and one in the spectrogram domain (with convolutions over the frequency axis). Each U-Net is made of 5 encoder layers, and 5 decoder layers. After the 5-th encoder layer, both representation have the same shape, and they are summed before going into a shared 6-th layer. Similarly, the first decoder layer is shared, and its output is sent both the temporal and spectral branch. The output of the spectral branch is transformed to a waveform using the iSTFT, before being summed with the output of the temporal branch, giving the actual prediction of the model.

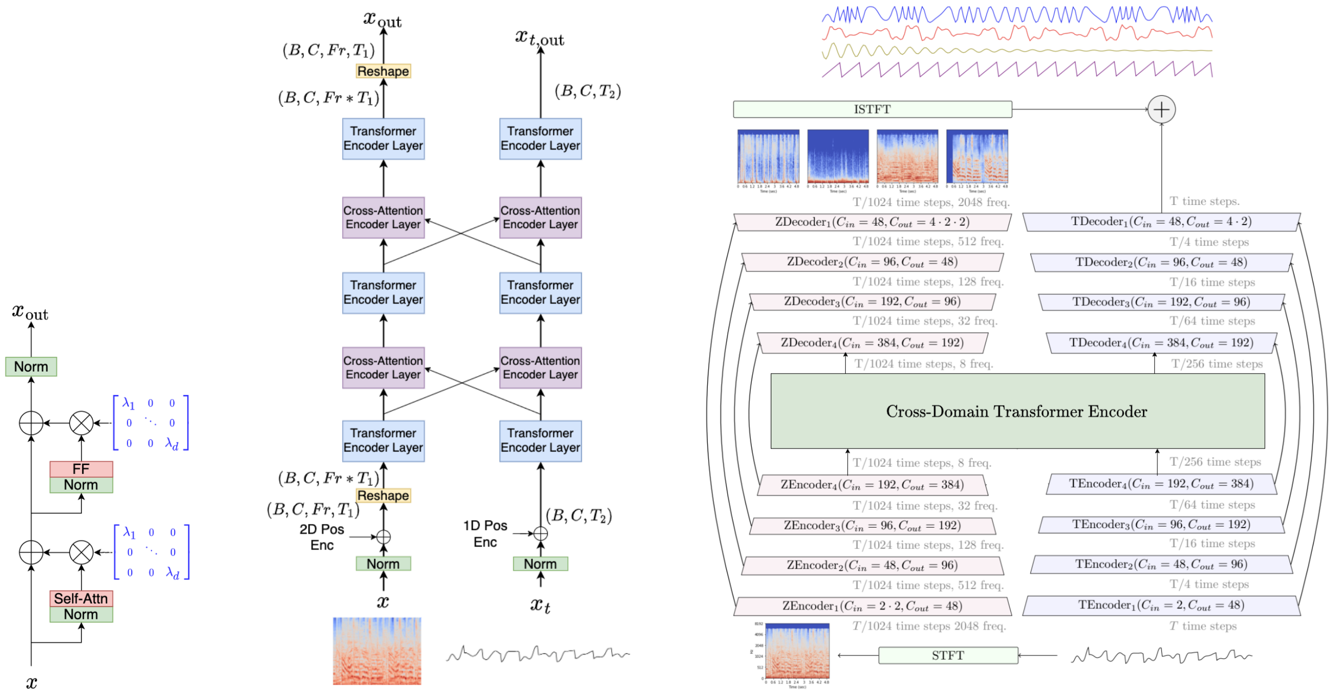
Hybrid Transformer Demucs keeps the outermost 4 layers as is from the original architecture, and replaces the 2 innermost layers in the encoder and the decoder, including local attention and bi-LSTM, with a cross-domain Transformer Encoder. For the first encoder, we tried $C_{\text{out}} = 32, 48, 64$ and retained 48. It treats in parallel the 2D signal from the spectral branch and the 1D signal from the waveform branch. Unlike the original Hybrid Demucs which required careful tuning of the model parameters (STFT window and hop length, stride, paddding, etc.) to align the time and spectral representation, the cross-domain Transformer Encoder can work with heterogeneous data shape, making it a more flexible architecture.

The architecture of Hybrid Transformer Demucs is depicted on Fig.1. On the left, we show a single self-attention Encoder layer of the Transformer [1] with normalizations before the Self-Attention and Feed-Forward operations, it is combined with Layer Scale [6] initialized to $\epsilon{=}10^{-4}$ in order to stabilize the training. The two first normalizations are layer normalizations (each token is independently normalized) and the third one is a time layer normalization (all the tokens are normalized together). The input/output dimension of the Transformer is 384, and linear layers are used to convert to the internal dimension of the Transformer when required. The attention mechanism has 8 heads and the hidden state size of the feed forward network is equal to 4 times the dimension of the transformer. The cross-attention Encoder layer is the same but using cross-attention with the other domain representation. In the middle, a cross-domain Transformer Encoder of depth 5 is depicted. It is the interleaving of self-attention Encoder layers and cross-attention Encoder layers in the spectral and waveform domain. 1D [1] and 2D [21] sinusoidal encodings are added to the scaled inputs and reshaping is applied to the spectral representation in order to treat it as a sequence. On Fig. 1 (c), we give a representation of the entire architecture, along with the double U-Net encoder/decoder structure.

Memory consumption and the speed of attention quickly deteriorates with an increase of the sequence lengths. To further scale, we leverage sparse attention kernels introduced in the `xformer` package [22], along with a Locally Sensitive Hashing (LSH) scheme to determine dynamically the sparsity pattern. We use a sparsity level of 90% (defined as the proportion of elements removed in the softmax), which is determined by performing 32 rounds of LSH with 4 buckets each. We select the elements that match at least k times over all 32 rounds of LSH, with k such that the sparsity level is 90%. We refer to this variant as Sparse HT Demucs.

4. DATASET

We curated an internal dataset composed of 3500 songs with the stems from 200 artists with diverse music genres. Each stem is assigned to one of the 4 sources according to the name given by the music producer (for instance ”vocals2”, ”fx”, ”sub” etc...). This labeling is noisy because these names are subjective and sometime ambiguous. For 150 of those tracks,



x_{out}
Norm
λ_1 0 0
0 ⋱ 0
0 0 λ_d
FF
Norm
Self-Attn
Norm
x
x_{out}
(B, C, Fr, T_1)
Reshape
$(B, C, Fr * T_1)$
$x_{t,\text{out}}$
(B, C, T_2)
Transformer Encoder Layer
Cross-Attention Encoder Layer
$(B, C, Fr * T_1)$
Reshape
(B, C, Fr, T_1)
2D Pos Enc
1D Pos Enc
(B, C, T_2)
Norm
x
x_t
ISTFT
T/1024 time steps, 2048 freq.
T time steps.
$\text{ZDecoder}_1(C_{in} = 48, C_{out} = 4 \cdot 2 \cdot 2)$
$\text{TDecoder}_1(C_{in} = 48, C_{out} = 4 \cdot 2)$
T/1024 time steps, 512 freq.
T/4 time steps
$\text{ZDecoder}_2(C_{in} = 96, C_{out} = 48)$
$\text{TDecoder}_2(C_{in} = 96, C_{out} = 48)$
T/1024 time steps, 128 freq.
T/16 time steps
$\text{ZDecoder}_3(C_{in} = 192, C_{out} = 96)$
$\text{TDecoder}_3(C_{in} = 192, C_{out} = 96)$
T/1024 time steps, 32 freq.
T/64 time steps
$\text{ZDecoder}_4(C_{in} = 384, C_{out} = 192)$
$\text{TDecoder}_4(C_{in} = 384, C_{out} = 192)$
T/1024 time steps, 8 freq.
T/256 time steps
Cross-Domain Transformer Encoder
T/1024 time steps, 8 freq.
T/256 time steps
$\text{ZEncoder}_4(C_{in} = 192, C_{out} = 384)$
$\text{TEncoder}_4(C_{in} = 192, C_{out} = 384)$
T/1024 time steps, 32 freq.
T/64 time steps
$\text{ZEncoder}_3(C_{in} = 96, C_{out} = 192)$
$\text{TEncoder}_3(C_{in} = 96, C_{out} = 192)$
T/1024 time steps, 128 freq.
T/16 time steps
$\text{ZEncoder}_2(C_{in} = 48, C_{out} = 96)$
$\text{TEncoder}_2(C_{in} = 48, C_{out} = 96)$
T/1024 time steps, 512 freq.
T/4 time steps
$\text{ZEncoder}_1(C_{in} = 2 \cdot 2, C_{out} = 48)$
$\text{TEncoder}_1(C_{in} = 2, C_{out} = 48)$
T/1024 time steps 2048 freq.
T time steps
STFT

(a) Transformer Encoder Layer

(b) Cross-domain Transformer Encoder of depth 5

(c) Hybrid Transformer Demucs

Fig. 1: Details of the Hybrid Transformer Demucs architecture. (a): the Transformer Encoder layer with self-attention and Layer Scale [6]. (b): The Cross-domain Transformer Encoder treats spectral and temporal signals with interleaved Transformer Encoder layers and cross-attention Encoder layers. (c): Hybrid Transformer Demucs keeps the outermost 4 encoder and decoder layers of Hybrid Demucs with the addition of a cross-domain Transformer Encoder between them.

we manually verified that the automated labeling was correct, and discarded ambiguous stems. We trained a first Hybrid Demucs model on MUSDB and those 150 tracks. We preprocess the dataset according to several rules. First, we keep only the stems for which all four sources are non silent at least 30% of the time. For each 1 second segment, we define it as silent if its volume is less than -40dB. Second, for a song x of our dataset, noting x_i, with $i \in \{\text{drums}, \text{bass}, \text{other}, \text{vocals}\}$, each stem and f the Hybrid Demucs model previously mentioned, we define $y_{i,j}{=}f(x_i)_j$ i.e. the output j when separating the stem i. Theoretically, if all the stems were perfectly labeled and if f were a perfect source separation model we would have $y_{i,j} = x_i \delta_{i,j}$ with $\delta_{i,j}$ being the Kronecker delta. For a waveform z, let us define the volume in dB measured over 1 second segments:

$$V(z) = 10 \cdot \log_{10} \left(\text{AveragePool}(z^2, 1\,\text{sec})\right). \qquad (1)$$

For each pair of sources i, j, we take the segments of 1 second where the stem is present (regarding the first criteria) and define $P_{i,j}$ as the proportion of these segments where $V(y_{i,j}) - V(x_i) > -10\,\text{dB}$. We obtain a square matrix $P \in [0, 1]^{4\times4}$, and we notice that in perfect condition, we should have $P = \text{Id}$. We thus keep only the songs for which for all sources i, $P_{i,i} > 70\%$, and pairs of sources $i \neq j$, $P_{i,j} < 30\%$. This procedure selects 800 songs.

5. EXPERIMENTS AND RESULTS

5.1. Experimental Setup

All our experiments are done on 8 Nvidia V100 GPUs with 32GB of memory, using fp32 precision. We use the L1 loss on the waveforms, optimized with Adam [23] without weight decay, a learning rate of $3 \cdot 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a batch size of 32 unless stated otherwise. We train for 1200 epochs of 800 batches each over the MUSDB18-HQ dataset, completed with the 800 curated songs dataset presented in Section 4, sampled at 44.1kHz and stereophonic. We use exponential moving average as described in [2], and select the best model over the valid set, composed of the valid set of MUSDB18-HQ and another 8 songs. We use the same data augmentation as described in [2], including repitching/tempo stretch and remixing of the stems within one batch.

Using one model per source can be beneficial [14], although its adds overhead both at train time and evaluation. In order to limit the impact at train time, we propose a procedure where one copy of the multi-target model is fine-tuned on a single target task for 50 epochs, with a learning rate of 10^{-4}, no remixing, repitching, nor rescaling applied as data augmentation. Having noticed some instability towards the end of the training of the main model, we use for the fine tuning a gradient clipping (maximum L2 norm of 5. for the gradient), and a weight decay of 0.05.

At test time, we split the audio into chunks having the same duration as the one used for training, with an overlap of 25% and a linear transition from one chunk to the next. We report is the Signal-to-Distortion-Ratio (SDR) as defined by the SiSEC18 [24] which is the median across the median SDR over all 1 second chunks in each song. On Tab. 2 we also report the Real Time Factor (RTF) computed on a single core of an Intel Xeon CPU at 2.20GHz. It is defined as the time to process some fixed audio input (we use 40 seconds of gaussian noise) divided by the input duration.

5.2. Comparison with the baselines

On Tab. 1, we compare to several state-of-the-art baselines. For reference, we also provide baselines that are trained without any extra training data. Comparing to the original Hybrid Demucs architecture, we notice that the improved sequence modeling capabilities of transformers increased the SDR by 0.45 dB for the simple version, and up to almost 0.9 dB when using the sparse variant of our model along with fine tuning. The most competitive baseline is Band-Split RNN [14], which achieves a better SDR on both the other and vocals sources, despite using only MUSDB18HQ as a supervised training set, and using 1750 unsupervised tracks as extra training data.

5.3. Impact of the architecture hyper-parameters

We first study the influence of three architectural hyper-parameters in Tab. 2: the duration in seconds of the excerpts used to train the model, the depth of the transformer encoder and its dimension. For short training excerpts (3.4 seconds), we notice that augmenting the depth from 5 to 7 increases the test SDR and that augmenting the transformer dimension from 384 to 512 lowers it slightly by 0.05 dB. With longer segments (7.8 seconds), we observe an increase of almost 0.6 dB when using a depth of 5 and 384 dimensions. Given this observation, we also tried to increase the duration or the depth but this led to Out Of Memory (OOM) errors. Finally, augmenting the dimension to 512 when training over 7.8 seconds led to an improvement of 0.1 dB.

5.4. Impact of the data augmentation

On Tab. 3, we study the impact of disabling some of the data augmentation, as we were hoping that using more training data would reduce the need for such augmentations. However, we observe a constant deterioration of the final SDR as we disable those. While the repitching augmentation has a limited impact, the remixing augmentation remain highly important to train our model, with a loss of 0.7 dB without it.

Table 2: Impact of segment duration, transformer depth and transformer dimension. OOM means Out of Memory. Results are commented in Sec. 5.3.

dur. (sec)	depth	dim.	nb. param.	RTF (cpu)	SDR (All)
3.4	5	384	26.9M	1.02	8.17
3.4	7	384	34.0M	1.23	8.26
3.4	5	512	41.4M	1.30	8.12
7.8	5	384	26.9M	1.49	8.70
7.8	7	384	34.0M	1.68	OOM
7.8	5	512	41.4M	1.77	**8.80**
12.2	5	384	26.9M	2.04	OOM

Table 3: Impact of data augmentation. The model has a depth of 5, and a dimension of 384. See Sec. 5.4 for details.

dur. (sec)	depth	remixing	repitching	SDR (All)
7.8	5	✓	✓	8.70
7.8	5	✓	✗	8.65
7.8	5	✗	✓	8.00

5.5. Impact of using sparse kernels and fine tuning

We test the sparse kernels described in Section 3 to increase the depth to 7 and the train segment duration to 12.2 seconds, with a dimension of 512. This simple change yields an extra 0.14 dB of SDR (8.94 dB). The fine-tuning per source improves the SDR by 0.25 dB, to 9.20 dB, despite requiring only 50 epochs to train. We tried further extending the receptive field of the Transformer Encoder to 15 seconds during the fine tuning stage by reducing the batch size, however, this led to the same SDR of 9.20 dB, although training from scratch with such a context might lead to a different result.

Conclusion

We introduced Hybrid Transformer Demucs, a Transformer based variant of Hybrid Demucs that replaces the innermost convolutional layers by a Cross-domain Transformer Encoder, using self-attention and cross-attention to process spectral and temporal informations. This architecture benefits from our large training dataset and outperforms Hybrid Demucs by 0.45 dB. Thanks to sparse attention techniques, we scaled our model to an input length up to 12.2 seconds during training which led to a supplementary gain of 0.4 dB. Finally, we could explore splitting the spectrogram into subbands in order to process them differently as it is done in [14].

6. REFERENCES

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017.

[2] Alexandre Défossez, "Hybrid spectrogram and waveform source separation," in *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.

[3] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner, "The musdb18 corpus for music separation," 2017.

[4] Nobutaka Ono, Zafar Rafii, Daichi Kitamura, Nobutaka Ito, and Antoine Liutkus, "The 2015 Signal Separation Evaluation Campaign," in *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Aug. 2015.

[5] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner, "Musdb18-hq - an uncompressed version of musdb18," Aug. 2019.

[6] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou, "Going deeper with image transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[7] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10684–10695.

[8] Tom B. Brown et al., "Language models are few-shot learners," 2020.

[9] Yi Luo and Nima Mesgarani, "Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.

[10] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach, "Music source separation in the waveform domain," 2019.

[11] 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.

[12] Naoya Takahashi and Yuki Mitsufuji, "D3net: Densely connected multidilated densenet for music source separation," 2020.

[13] Woosung Choi, Minseok Kim, Jaehwa Chung, and Soonyoung Jung, "Lasaft: Latent source attentive frequency transformation for conditioned source separation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[14] Yi Luo and Jianwei Yu, "Music source separation with band-split rnn," 2022.

[15] 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.

[16] Daniel Stoller, Sebastian Ewert, and Simon Dixon, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," *arXiv preprint arXiv:1806.03185*, 2018.

[17] Minseok Kim, Woosung Choi, Jaehwa Chung, Daewon Lee, and Soonyoung Jung, "Kuielab-mdx-net: A two-stream neural network for music demixing," 2021.

[18] Yuki Mitsufuji, Giorgio Fabbro, Stefan Uhlich, Fabian-Robert Stöter, Alexandre Défossez, Minseok Kim, Woosung Choi, Chin-Yun Yu, and Kin-Wai Cheuk, "Music demixing challenge 2021," *Frontiers in Signal Processing*, vol. 1, jan 2022.

[19] 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*, 2020.

[20] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong, "Attention is all you need in speech separation," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 21–25.

[21] Zelun Wang and Jyh-Charn Liu, "Translating math formula images to latex sequences using deep neural networks with sequence-level training," 2019.

[22] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, and Susan Zhang, "xformers: A modular and hackable transformer modelling library," https://github.com/facebookresearch/xformers, 2021.

[23] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2014.

[24] Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito, "The 2018 signal separation evaluation campaign," 2018.