REAL-TIME LOW-LATENCY MUSIC SOURCE SEPARATION USING HYBRID SPECTROGRAM-TASNET

Satvik Venkatesh, Arthur Benilov, Philip Coleman, Frederic Roskam

L-Acoustics, 67 Southwood Lane, London N65EG.

ABSTRACT

There have been significant advances in deep learning for music demixing in recent years. However, there has been little attention given to how these neural networks can be adapted for real-time low-latency applications, which could be helpful for hearing aids, remixing audio streams and live shows. In this paper, we investigate the various challenges involved in adapting current demixing models in the literature for this use case. Subsequently, inspired by the Hybrid Demucs architecture, we propose the Hybrid Spectrogram Time-domain Audio Separation Network (HS-TasNet), which utilises the advantages of spectral and waveform domains. For a latency of 23 ms, the HS-TasNet obtains an overall signal-to-distortion ratio (SDR) of 4.65 on the MusDB test set, and increases to 5.55 with additional training data. These results demonstrate the potential of efficient demixing for real-time low-latency music applications.

Index Terms— Real-time, audio source separation, hybrid features, TasNet, demix

1. INTRODUCTION

Audio source separation is the process of separating a mixture of audio into its individual components. This is useful for various applications such as speech enhancement [1], speech separation [2], dereverberation [3], and as a pre-processing step for sound event detection [4]. Music source separation (MSS), also popularly known as demixing or unmixing, enables us to remix the balance within the music, perhaps to make the vocals louder or to suppress an unwanted sound, change the spatial location of a musical instrument [5], create loops from individual instruments [6], object-based mixing [7] or even upmix a 2-channel stereo recording to a multi-channel surround sound system. In recent years, there has been tremendous interest in using deep learning for MSS, leading to significant improvements in separation quality. These studies have modelled MSS as a supervised learning problem, where the constituting components are known beforehand. For example, *vocals*, *drums*, *bass*, and *other*. However, the computational cost and latency of these models are generally high [8, 9], making them suitable for offline tasks and studio productions, but not for real-time low-latency scenarios.

Before the onset of deep learning, [10] presented Azimuth Discrimination and Resynthesis for real-time MSS. The window and hop sizes for the short-time Fourier transform (STFT) were set to 93 ms and 23 ms respectively. Therefore, considering the overlap-add operation of the inverse-STFT, the latency of the algorithm is 93 ms. In [11], a multi-layer perceptron (MLP) with an algorithmic latency of 23 ms performed singing voice separation. Their goal was to remix music for cochlear implant users. Unlike MSS, researchers have widely explored deep learning for real-time speech separation [2] and enhancement [1]. For STFT-based speech enhancement algorithms, typically a latency of 32 ms is observed, due to the required window size [12, 13]. End-to-end deep learning directly on raw audio for speech separation has obtained impressive results with latency as low as 2 ms [12]. However, low-latency MSS is more challenging due to higher sampling rates, high dependence on future context, and potentially larger models.

Neural network architectures for MSS can be broadly classified into spectrogram-based, waveform-based, and hybrid approaches. Initially, spectrogram-based models only used the magnitude and discarded the phase [14, 15, 16]. The network predicted the magnitude mask and used the phase of the input mixture for the final output. The waveform-based models that directly separate raw audio have the advantage of not discarding phase and extracting potentially useful deep features [15, 9]. Furthermore, studies have explored ways to predict phase [17] and also predict real and imaginary parts of the spectrogram [18] to obtain state-of-the-art results. Recently, studies have also proposed hybrid frameworks that harness the advantages of both waveform and spectral domains [19, 20].

In this paper, we investigate real-time MSS with a low-latency such as 23 ms (frame size of 1024 samples at 44.1kHz). To our knowledge, this is the first paper that explores deep learning to separate *vocals*, *drums*, *bass*, and *other* in real-time with low-latency. Using 23 ms was a good starting point because it is large enough for the Fourier spectrum to capture meaningful spectral features, but small enough for real-time applications such as cochlear implants [11] and stereo upmixing [10, 21]. First, we adapt the spectrogram-based X-UMX [16] and the waveform-based TasNet [2, 22] for low-latency MSS. When adapting these models for real-time processing, we observe artifacts and a considerable drop in separation quality. Therefore, we propose a novel hybrid architecture called the Hybrid Spectrogram Time-domain Audio Separation Network (HS-TasNet) that outperforms both models and demonstrates the potential for efficient real-time demixing. We perform objective and subjective evaluations to compare this real-time framework to models designed for offline processing and obtain competitive results on the MusDB dataset.

2. STATE-OF-THE-ART ARCHITECTURES

Many state-of-the-art models adopt a U-Net-style architecture [9, 14, 17, 23], where the network encodes a large context of input audio into a latent space representation, and subsequently uses a decoder to reconstruct the audio. In addition, architectures such as Wave-U-Net [15] and Demucs [9] utilise past and future contexts of a given mixture to improve separation performance at the edges. The input receptive field for such U-Nets is in the range of 2 s to 30 s [15, 20], which makes it challenging to adapt to low-latency scenarios.

The Open-Unmix (UMX) model [24] is a simple bidirectional LSTM-based framework that obtained state-of-the-art results on the MusDB dataset in 2018. It only adopts stacked LSTM layers and hence, does not depend on a latent space representation like U-Nets. CrossNet-Open-Unmix (X-UMX) [16] improved the UMX by bridging these independent networks and proposing a novel combinational and multi-domain loss. In order to develop a real-time implementation of such a model, we can replace the bidirectional LSTM with a causal unidirectional LSTM.

The time-domain audio separation network (TasNet) [2] designed for real-time speech separation in the waveform domain obtained state-of-the-art results and surpassed spectrogram-based approaches for causal and non-causal implementations. The model adopts an encoder-masker-decoder structure, where the encoder is a learned 1D convolution that creates a non-negative representation of the input. The advantage of TasNet is that it is not limited by the spectral resolution of the STFT and hence, obtains a latency as low as 5 ms. The non-causal TasNet was adopted by [22] for MSS.

In Conv-TasNet [12], the LSTM layers are replaced by temporal convolutions. This significantly speeds up the training speed and allows the algorithm to have lower latency due to potentially smaller kernel sizes [12]. Multiple studies have adopted the Conv-TasNet for MSS [9, 25] showing promising results. However, for real-time separation, the disadvantage of Conv-TasNet is that the performance of the algorithm is sensitive to the input receptive field. As explained by [12], for 2-speaker separation, the performance of the algorithm dropped when the receptive field was reduced from 1.5 s to 0.5 s. Although a large receptive field does not reduce the algorithmic latency, it hinders computational efficiency because a larger audio block needs to be processed at every time step.

3. PROPOSED MODELS

3.1. Low-Latency Constraints

There are multiple considerations that need to be made when developing a real-time demixing system. The first factor is algorithmic latency, which is the amount of context required to output one audio frame. This could be caused by overlap-add [13] or look-ahead windows [26]. The second factor is computational efficiency, which is the time taken to process a frame of audio. For instance, if the algorithm uses too many look-behind windows or the network has many blocks that need to be sequentially processed, then the algorithm is said to have poor computational efficiency.

To ensure feasibility during real-time deployment we make certain considerations when designing models. For a given hop size, the average inference time on 4 CPU cores should be less than 50% of the hop size, to allow reasonable room for updating audio blocks. Similar real-time constraints were imposed by the Deep Noise Suppression Challenge at ICASSP 2023 [27]. We are aware that this may change with increasing computational power in future research, which is beyond the scope of this study. More details on the hardware used can be found in section 5.2.

3.2. Low-Latency Adaptations

X-UMX [16] is a spectrogram-based model whose output goes through a Wiener filter [28], to improve separation quality and minimise interference between sources. For our causal implementation, we changed the bidirectional LSTMs to unidirectional LSTMs, reduced the window size from 4096 to 1024, and the hop size from 1024 to 512. We were unable to use the Wiener filter for real-time processing because it adopts an expectation maximization algorithm, which requires look-ahead windows for good performance [28].

TasNet in [22] used a window size of 220 samples (5 ms at 44.1kHz sample rate). Although studies state that smaller window sizes lead to better performance in the TasNet [2, 12], we did not observe the same trend in this causal setup. Initial experiments of hyperparameter tuning showed that a window size of 1024 performed better than smaller sizes. This is expected because the neural network has access to more context of audio at each time step. Therefore, we adopted window and hop sizes of 1024 and 512 respectively. Similar to [2, 3], we set the number of hidden units in the unidirectional LSTMs to 1000 and added an identity skip connection between every two LSTM layers. The number of basis signals N in the convolutional layer was set to 1500.

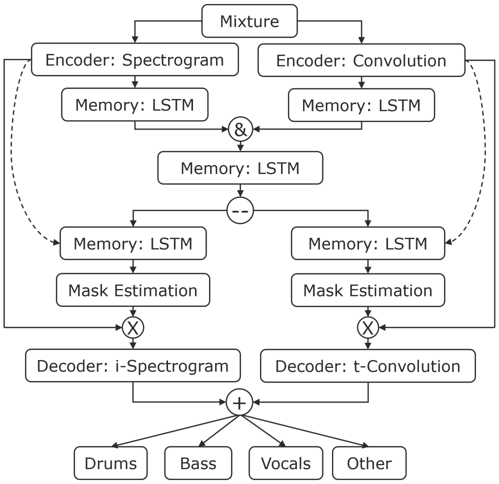

Fig. 1. Structure of the HS-TasNet. ‘&’, ‘--’, ‘+’, ‘i-’, ‘t-’, and dotted lines stand for concatenate, split, sum, inverse, transpose and skip connections respectively.

The final layer in the TasNet is a transposed convolution comprising learned filters. To avoid discontinuities in the audio output, we applied a Hanning window to the filters when performing the transposed convolution. This leads to a seamless overlap-add mechanism during audio reconstruction. Such an approach has also been adopted by studies that use analytic filters [29] and sinc filters [30].

3.3. Hybrid Spectrogram TasNet (HS-TasNet)

Our proposed model HS-TasNet is inspired by the hybrid Demucs architecture [20] that comprises a temporal branch, a spectral branch, and shared layers. Waveform-based models are well suited to *drums* and *bass*, while spectrogram-based models can be better for *vocals* and *other* [9]. The hybrid framework harnesses the advantages of both the spectrogram and waveform domains.

Figure 1 illustrates the architecture of the HS-TasNet. The spectrogram encoder generates 513 features as the window size is 1024. The learned convolution encoder generates 1024 basis signals. Each ‘Memory: LSTM’ block in figure 1 consists of two LSTMs with an identity skip connection added to the output of the second LSTM. The number of hidden units for the LSTMs in the spectrogram and learned convolution branches is 500. The combined branch has 1000 hidden units because the features from the individual branches are concatenated. In the ‘Memory: LSTM’ block after the ‘split’ layer, the encoded representation is added as a skip connection, instead of an identity one. Linear fully-connected layers were used to address any dimension mismatch between layers.

We also propose a computationally cheaper version which we refer to as *HS-TasNet-Small*. Here, we replace the ‘concatenate’ layer with a ‘summation’. This way, the number of the LSTM units in the combined branch is only 500, instead of 1000. Moreover, in each 'Memory: LSTM' block, we use one LSTM instead of two. This reduces the number of parameters in the model from 42 M to 16 M.

4. EXPERIMENTAL SETUP

Dataset: This study adopts the MusDB-HQ dataset [31] comprising 86 tracks for training, 14 tracks for validation, and 50 tracks for testing. All audio files are stereo at 44.1 kHz sampling rate. We also trained the models using an internal dataset of 150 songs.

Training Strategy: The implementation of X-UMX to train on MusDB is openly available in this repository[1]. We used the same training pipeline and hyperparameters as the original implementation, except for changing the window size, hop size, and LSTM from bidirectional to unidirectional. Please refer to [16, 32] for more details on the implementation. For the TasNet model, we adopted the training pipeline by DemucsV2 [9], which is openly available in this repository[2], as it was more optimised to train waveform-based models. It was optimised using the L1 loss with data augmentations — channel swapping, random gain, shuffling sources, and pitch/tempo shifting [24, 9]. The initial learning rate was set to 3×10^{-4} and was decayed by 0.5 if the validation loss did not improve for 3 epochs. Training was stopped if the validation loss did not improve for 10 epochs. We used the same set of parameters to train the HS-TasNet.

Evaluation: To compare our models with the state-of-the-art, we use signal-to-distortion ratio (SDR), which is popularly adopted in the literature [9, 15, 23]. The models trained with additional data were also subjectively evaluated through listening tests, using the Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) framework [33]. 16 participants[3] were asked to rate 8 pages of 7 examples each (5 models, a reference and an anchor) — 4 pages were focused on *quality* (timbral fidelity and absence of artefacts) and 4 pages were focused on the rejection of *interference* from remaining sources. We adapted webMUSHRA [33] to randomly sample examples from the MusDB test set. The *quality* anchor was a distorted version of the reference, using the algorithm presented in [34]. The *interference* anchor was the original mixture of audio. ANOVA and post-hoc Tukey tests were used to conduct statistical analysis.

5. RESULTS

5.1. Evaluation

X-UMX: As presented in table 1, there is a noticeable drop in the SDR when we exclude Wiener filtering and change the LSTM from bidirectional to unidirectional. For a real-time demixer with a latency of 93 ms, the overall SDR is 4.57 which is significantly lower than the original 5.79. For a latency of 23 ms, it further drops to 3.93. This demonstrates the importance of future context and higher window sizes in MSS, which is unavailable during real-time processing.

Table 1. The X-UMX model evaluated on MusDB with and without Wiener filtering (WF). The window size (win.) of the STFT is equal to the latency of the algorithm. Uni. denotes whether the LSTM is unidirectional and RT specifies if the algorithm can run in real-time.

WF?	Uni?	RT?	Win. (ms)	All	Voc.	Dru.	Bass	Oth.
✓	✗	✗	93	5.79	6.61	6.47	5.43	4.64
✗	✗	✗	93	5.13	6.02	5.62	4.58	4.29
✓	✓	✗	93	5.05	5.73	5.53	4.86	4.09
✗	✓	✓	93	4.57	5.49	4.69	4.13	3.96
✓	✓	✗	23	4.08	4.87	4.66	3.76	3.03
✗	✓	✓	23	3.93	4.65	4.36	3.79	2.92

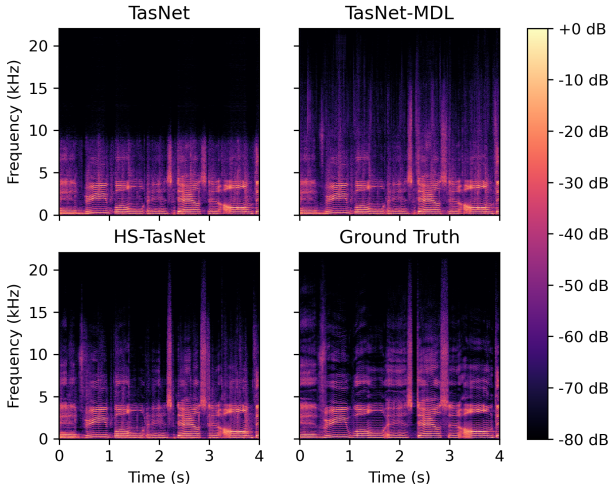

Fig. 2. Comparison of spectra produced by TasNet, TasNet with multi-domain loss, HS-TasNet, and the Ground Truth.

TasNet: As presented in Table 2, TasNet obtained a better SDR than the X-UMX for a latency of 23 ms. However, as shown in Figure 2, we observed that the model was unable to effectively generate higher frequencies in the spectrum. Interestingly, we found that this is associated with the loss function because when we use L1 directly in the waveform domain, there is a risk of overweighting the importance of lower frequencies [35, 36]. Studies for speech separation have demonstrated the superior performance of scale-invariant source-to-noise ratio (SI-SNR) and scale-dependent SDR over L1/L2 for TasNets [3, 37]. However, in our experiments with MSS, SI-SNR and SD-SDR lead to poorer performance and slower convergence. We believe this might be due to the data augmentation that we have employed (which was not performed in [22] during training with SI-SNR) or potentially higher correlation between sources ([3] found that MSE was better than SI-SNR for dereverberation due to higher correlation between sources). We also explored the multi-domain loss [16], which calculates the loss on frequency and time domains. As shown in Figure 2, this enabled the network to generate higher frequencies, however, the output had higher leakage between stems and did not lead to an overall improvement from using L1.

HS-TasNet obtained an overall SDR of 4.65 and surpassed the TasNet's performance for all four sources. As shown in Figure 2, there were also no issues with generating higher frequencies. The smaller version of the HS-TasNet obtains an overall SDR of 4.44. In Table 2, we present the SDR scores of X-UMX, TasNet, and HS-TasNet compared to the state-of-the-art models. The HS-TasNet scales better to additional training data. The overall SDR is 5.55 when trained on additional data, versus 4.92 for the TasNet and 4.10 for the X-UMX. The HS-TasNet-Small obtains an overall SDR of 5.01, which is still higher than TasNet and X-UMX.

Listening test results in table 3 show that DemucsV4 significantly outperforms other models for *quality* and *interference* ($p < 0.001$). HS-TasNet was significantly better than TasNet ($p < 0.005$). There were no significant differences between HS-TasNet, DemucsV2, and

[1] https://github.com/asteroid-team/asteroid

[2] https://github.com/facebookresearch/demucs/tree/v2

[3] Participants were from L-Acoustics and an informed consent was obtained before the test.

Table 2. Comparing our low-latency implementations of X-UMX, TasNet, and HS-TasNet to the state-of-the-art models. 'Extra' stands for how much training data in addition to MusDB is used. 1-core, 4-core, and GPU mention the time taken by the network to process 23 ms of audio using the respective hardware averaged across 1000 iterations. † indicates from current work.

	Architecture	All	Vocals	Drums	Bass	Other	Extra?	Latency (ms)	Param.	1-core (ms)	4-core (ms)	GPU (ms)
Non-causal models	Conv-TasNet [9]	5.73	6.43	6.02	6.20	4.27	✗	2,000	9 M	**52.29**	**46.50**	**16.77**
	X-UMX [16]	5.79	6.61	6.47	5.43	4.64	✗	8,000	36 M	-	-	-
	DemucsV2 [9]	6.28	6.84	6.86	7.01	4.42	✗	8,000	288 M	-	-	-
	DemucsV4 [23]	**7.52**	**7.93**	**7.94**	**8.48**	**5.72**	✗	7,800	41 M	-	-	-
Non-causal models with extra data	Conv-TasNet [9]	6.32	6.74	7.11	7.00	4.44	150	2,000	9 M	52.29	46.50	16.77
	X-UMX [32]	6.52	7.57	7.39	6.28	4.83	1505	8,000	36 M	-	-	-
	TasNet [22]	6.52	7.34	7.68	7.04	4.04	2500	15,000	29 M	**11.75**	**4.20**	**2.38**
	DemucsV2 [9]	6.79	7.29	7.58	7.60	4.69	150	8,000	288 M	-	-	-
	DemucsV4 [23]	**9.20**	**9.37**	**10.83**	**10.47**	**6.41**	800	7,800	41 M	-	-	-
Real-time Low-Latency models	X-UMX†	3.93	4.65	4.36	3.79	2.92	✗	23	31 M	9.85	4.06	**1.80**
	TasNet†	4.40	5.02	4.38	**4.73**	3.48	✗	23	51 M	9.81	4.45	1.90
	HS-TasNet†	**4.65**	5.13	**5.22**	4.59	**3.64**	✗	23	42 M	9.10	4.26	3.90
	HS-TasNet-Small†	4.48	**5.21**	4.89	4.42	3.42	✗	23	16 M	**3.98**	**1.83**	2.10
Real-time Low-Latency models with extra data	X-UMX†	4.10	4.74	4.62	3.76	3.28	150	23	31 M	9.85	4.06	**1.80**
	TasNet†	4.92	5.22	5.54	5.13	3.78	150	23	51 M	9.81	4.45	1.90
	HS-TasNet†	**5.55**	**5.97**	**6.34**	**5.62**	**4.28**	150	23	42 M	9.10	4.26	3.90
	HS-TasNet-Small†	5.01	5.51	5.75	4.93	3.86	150	23	16 M	**3.98**	**1.83**	2.10

Conv-TasNet ($p > 0.1$). For *quality*, the HS-TasNet is slightly behind DemucsV2 and equal to Conv-TasNet. HS-TasNet surpasses DemucsV2 and Conv-TasNet for the rejection of *interference* from remaining sources. Audio examples are available here[4].

5.2. Inference Times

In Table 2, we also present the time taken by the algorithm to run on an Intel i7-12850HX CPU on 1-core and 4-cores, and NVIDIA RTX 3080Ti GPU. The offline versions of TasNet [22] and Conv-TasNet [9] cannot run in real-time, but we still present the time taken to process 1024 samples. Although the Conv-TasNet has only 9 M parameters, it has many blocks that require sequential processing, which makes it computationally inefficient. The other offline models like Demucs cannot process a block as small as 1024 due to the large number of layers. The inference time of all our real-time models is around 4 ms when running on 4-cores, which gives it enough time to update within the hop size of 11 ms. The HS-TasNet-Small can run comfortably even on a single CPU core.

Table 3. The mean (± standard deviation) from the listening test. The upper half is for *quality* and the bottom half is for rejection of *interference*. The bold values indicate the highest value excluding the reference and DemucsV4. † indicates from current work.

Model	All	Vocals	Drums	Bass	Other
Reference	99$_{\pm 18}$	99$_{\pm 2}$	95$_{\pm 18}$	100$_{\pm 0}$	100$_{\pm 0}$
DemucsV4 [23]	66$_{\pm 26}$	70$_{\pm 23}$	67$_{\pm 24}$	49$_{\pm 29}$	78$_{\pm 26}$
DemucsV2 [9]	**39$_{\pm 23}$**	34$_{\pm 22}$	**48$_{\pm 27}$**	**34$_{\pm 24}$**	40$_{\pm 23}$
HS-TasNet†	37$_{\pm 23}$	**44$_{\pm 23}$**	39$_{\pm 21}$	29$_{\pm 21}$	38$_{\pm 23}$
Conv-TasNet [9]	37$_{\pm 31}$	39$_{\pm 21}$	44$_{\pm 26}$	21$_{\pm 23}$	**43$_{\pm 31}$**
TasNet†	27$_{\pm 23}$	26$_{\pm 13}$	27$_{\pm 18}$	25$_{\pm 25}$	30$_{\pm 23}$
Anchor (Quality)	5$_{\pm 9}$	4$_{\pm 8}$	1$_{\pm 3}$	10$_{\pm 16}$	5$_{\pm 9}$
Reference	99$_{\pm 4}$	99$_{\pm 5}$	100$_{\pm 0}$	100$_{\pm 1}$	99$_{\pm 4}$
DemucsV4 [23]	78$_{\pm 12}$	79$_{\pm 15}$	86$_{\pm 12}$	64$_{\pm 29}$	83$_{\pm 12}$
DemucsV2 [9]	52$_{\pm 22}$	42$_{\pm 20}$	57$_{\pm 18}$	61$_{\pm 21}$	49$_{\pm 22}$
HS-TasNet†	**58$_{\pm 23}$**	**54$_{\pm 22}$**	**60$_{\pm 20}$**	**62$_{\pm 23}$**	55$_{\pm 23}$
Conv-TasNet [9]	48$_{\pm 26}$	36$_{\pm 18}$	53$_{\pm 25}$	46$_{\pm 22}$	**56$_{\pm 26}$**
TasNet†	47$_{\pm 22}$	40$_{\pm 19}$	45$_{\pm 20}$	61$_{\pm 25}$	43$_{\pm 22}$
Anchor (Interference)	3$_{\pm 25}$	0$_{\pm 1}$	1$_{\pm 3}$	2$_{\pm 5}$	9$_{\pm 25}$

6. CONCLUSION

This paper explored real-time low-latency demixing for vocals, drums, bass, and other. A few challenges associated with this task include the non-availability of future context and the poor computational efficiency of U-Nets. Initially, we adapted the X-UMX and TasNet models for low-latency demixing. We observed that the performance of the spectrogram-based model drops with reducing the window size due to the spectral resolution of the STFT. Although the TasNet performed better than the X-UMX, we observed artifacts such as missing higher frequencies. Therefore, we proposed a novel model called the HS-TasNet that harnesses the advantages of both spectral and waveform domains. It obtained an overall SDR of 4.65 when trained only on MusDB and increased to 5.55 with additional training data. Listening tests showed that the HS-TasNet significantly outperformed the TasNet and it was comparable to DemucsV2, which was the state-of-the-art model two years ago [9]. These results demonstrate the potential of efficient demixing for real-time low-latency music applications.

This study considered 23 ms to be the required latency for the demixing models. Future work can evaluate the feasibility of further reducing the latency, which would make it applicable to a variety of live music scenarios. Moreover, training strategies such as decomposing/reorganizing a Bi-RNN layer [38] and using non-causal models to train causal models [39] can mitigate the performance degradation when changing non-causal models to their causal counterparts.

7. ACKNOWLEDGMENTS

The authors thank Emmanouil Benetos from Queen Mary University of London and Selim Sheta, Thomas Fouchard, Nicolas Epain, Etienne Corteel, and Guillaume Le Nost from L-Acoustics for their feedback and fruitful discussions.

[4] https://l-acoustics.github.io/demix.github.io/

8. REFERENCES

[1] Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi, "Real time speech enhancement in the waveform domain," *arXiv preprint arXiv:2006.12847*, 2020.

[2] Yi Luo and Nima Mesgarani, "TasNet: time-domain audio separation network for real-time, single-channel speech separation," in *Proc. IEEE ICASSP*, 2018, pp. 696–700.

[3] Yi Luo and Nima Mesgarani, "Real-time single-channel dereverberation and separation with time-domain audio separation network.," in *Interspeech*, 2018, pp. 342–346.

[4] Nicolas Turpault, Romain Serizel, Scott Wisdom, et al., "Sound event detection and separation: a benchmark on desed synthetic soundscapes," in *Proc. IEEE ICASSP*, 2021, pp. 840–844.

[5] Estefania Cano, Derry FitzGerald, Antoine Liutkus, Mark D Plumbley, and Fabian-Robert Stöter, "Musical source separation: An introduction," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 31–40, 2018.

[6] Jordan BL Smith and Masataka Goto, "Nonnegative tensor factorization for source separation of loops in audio," in *Proc. IEEE ICASSP*, 2018, pp. 171–175.

[7] Philip Coleman, Andreas Franck, Jon Francombe, et al., "An audio-visual system for object-based audio: from recording to listening," *IEEE Trans. Multimedia*, vol. 20, no. 8, pp. 1919–1931, 2018.

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

[9] Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[10] Dan Barry, Eugene Coyle, and Bob Lawlor, "Real-time sound source separation: Azimuth discrimination and resynthesis," in *117th Convention of the Audio Engineering Society*, 2004.

[11] Sina Tahmasebi, Tom Gajęcki, and Waldo Nogueira, "Design and evaluation of a real-time audio source separation algorithm to remix music for cochlear implant users," *Front. Neurosci.*, vol. 14, pp. 434, 2020.

[12] Yi Luo and Nima 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.

[13] Zhong-Qiu Wang, Gordon Wichern, Shinji Watanabe, and Jonathan Le Roux, "STFT-domain neural speech enhancement with very low algorithmic latency," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 397–410, 2022.

[14] A Jansson, E Humphrey, N Montecchio, R Bittner, A Kumar, and T Weyde, "Singing voice separation with deep U-Net convolutional networks," in *Proc. ISMIR*, 2017, pp. 23–27.

[15] 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.

[16] Ryosuke Sawata, Stefan Uhlich, Shusuke Takahashi, and Yuki Mitsufuji, "All for one and one for all: Improving music separation by bridging networks," in *Proc. IEEE ICASSP*, 2021, pp. 51–55.

[17] Qiuqiang Kong, Yin Cao, Haohe Liu, Keunwoo Choi, and Yuxuan Wang, "Decoupling magnitude and phase estimation with deep resunet for music source separation," *arXiv preprint arXiv:2109.05418*, 2021.

[18] Yi Luo and Jianwei Yu, "Music source separation with band-split RNN," *arXiv preprint arXiv:2209.15174*, 2022.

[19] Minseok Kim, Woosung Choi, Jaehwa Chung, Daewon Lee, and Soonyoung Jung, "Kuielab-mdx-net: A two-stream neural network for music demixing," *arXiv preprint arXiv:2111.12203*, 2021.

[20] Alexandre Défossez, "Hybrid spectrogram and waveform source separation," *arXiv preprint arXiv:2111.03600*, 2021.

[21] Dan Barry, "Real-time sound source separation for music applications," *Doctoral thesis, Technological University Dublin*, 2019.

[22] Emery Pierson Lancaster and Nathan Souviraà-Labastie, "A frugal approach to music source separation," *hal-02986241*, 2020.

[23] Simon Rouard, Francisco Massa, and Alexandre Défossez, "Hybrid transformers for music source separation," in *Proc. IEEE ICASSP*, 2023, pp. 1–5.

[24] Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji, "Open-unmix-a reference implementation for music source separation," *J. Open Source Softw.*, vol. 4, no. 41, pp. 1667, 2019.

[25] David Samuel, Aditya Ganeshan, and Jason Naradowsky, "Meta-learning extractors for music source separation," in *Proc. IEEE ICASSP*, 2020, pp. 816–820.

[26] Yanxin Hu, Yun Liu, Shubo Lv, et al., "DCCRN: Deep complex convolution recurrent network for phase-aware speech enhancement," *arXiv preprint arXiv:2008.00264*, 2020.

[27] Harishchandra Dubey, Ashkan Aazami, Vishak Gopal, et al., "ICASSP 2023 deep speech enhancement challenge," *arXiv preprint arXiv:2303.11510*, 2023.

[28] Antoine Liutkus and Fabian-Robert Stöter, "Norbert: Multichanne-wiener filtering," Sept. 2019.

[29] Manuel Pariente, Samuele Cornell, Antoine Deleforge, and Emmanuel Vincent, "Filterbank design for end-to-end speech separation," in *Proc. IEEE ICASSP*, 2020, pp. 6364–6368.

[30] Mirco Ravanelli and Yoshua Bengio, "Speaker recognition from raw waveform with sincnet," in *Proc. IEEE SLT*, 2018, pp. 1021–1028.

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

[32] Ryosuke Sawata, Naoya Takahashi, Stefan Uhlich, Shusuke Takahashi, and Yuki Mitsufuji, "The whole is greater than the sum of its parts: Improving DNN-based music source separation," *arXiv preprint arXiv:2305.07855*, 2023.

[33] Michael Schoeffler, Sarah Bartoschek, Fabian-Robert Stöter, et al., "webMUSHRA—a comprehensive framework for web-based listening tests," *J. Open Res. Softw.*, vol. 6, no. 1, pp. 8, 2018.

[34] Dominic Ward, Hagen Wierstorf, Russell D Mason, Emad M Grais, and Mark D Plumbley, "BSS eval or PEASS? predicting the perception of singing-voice separation," in *Proc. IEEE ICASSP*, 2018, pp. 596–600.

[35] Brian CJ Moore, *An introduction to the psychology of hearing*, Brill, 2012.

[36] Alec Wright and Vesa Välimäki, "Perceptual loss function for neural modeling of audio systems," in *Proc. IEEE ICASSP*, 2020, pp. 251–255.

[37] Jens Heitkaemper, Darius Jakobeit, Christoph Boeddeker, Lukas Drude, and Reinhold Haeb-Umbach, "Demystifying TasNet: A dissecting approach," in *Proc. IEEE ICASSP*, 2020, pp. 6359–6363.

[38] Kai Li and Yi Luo, "On the design and training strategies for rnn-based online neural speech separation systems," in *Proc. IEEE ICASSP*, 2023, pp. 1–5.

[39] Konstantinos Drossos, Stylianos Ioannis Mimilakis, Dmitriy Serdyuk, Gerald Schuller, Tuomas Virtanen, and Yoshua Bengio, "MaD TwinNet: Masker-denoiser architecture with twin networks for monaural sound source separation," in *Proc. IEEE IJCNN*, 2018, pp. 1–8.