Retrieving Signals with Deep Complex Extractors

Chiheb Trabelsi, ♣♠ **Olexa Bilaniuk,** ♣ **Ousmane Dia,** ♠ **Ying Zhang,** ♠ **Mirco Ravanelli,** ♣

Jonathan Binas,♣ **Negar Rostmazadeh,**♠ **Christopher J Pal,**♣♠¶
♣ Quebec Artificial Intelligence Institute (Mila), Montreal, Quebec
♠ Element AI, Montreal, Quebec
¶ CIFAR Fellow

Abstract

Recent advances have made it possible to create deep complex-valued neural networks. Despite this progress, many challenging learning tasks have yet to leverage the power of complex representations. Building on recent advances, we propose a new deep complex-valued method for signal retrieval and extraction in the frequency domain. As a case study, we perform audio source separation in the Fourier domain. Our new method takes advantage of the convolution theorem which states that the Fourier transform of two convolved signals is the elementwise product of their Fourier transforms. Our novel method is based on a complex-valued version of Feature-Wise Linear Modulation (FiLM) and serves as the keystone of our proposed signal extraction method. We also introduce a new and explicit amplitude and phase-aware loss, which is scale and time invariant, taking into account the complex-valued components of the spectrogram. Using the Wall Street Journal Dataset, we compared our phase-aware loss to several others that operate both in the time and frequency domains and demonstrate the effectiveness of our proposed signal extraction method and proposed loss.

1 Introduction

Complex-valued neural networks have been studied since long before the emergence of modern deep learning techniques [Georgiou and Koutsougeras, 1992, Zemel et al., 1995, Kim and Adalı, 2003, Hirose, 2003, Nitta, 2004]. Nevertheless, deep complex-valued models have only just started to gain momentum [Reichert and Serre, 2014, Arjovsky et al., 2015, Danihelka et al., 2016, Trabelsi et al., 2017, Jose et al., 2017, Wolter and Yao, 2018b, Choi et al., 2019], with the great majority of models in deep learning still relying on real-valued representations. The motivation for using complex-valued representations for deep learning is twofold: On the one hand, biological nervous systems actively make use of synchronization effects to gate signals between neurons – a mechanism that can be recreated in artificial systems by taking into account phase differences [Reichert and Serre, 2014]. On the other hand, complex-valued representations are better suited to certain types of data, particularly those that are naturally expressed in the frequency domain.

Other benefits provided by working with complex-valued inputs in the spectral (frequency) domain are computational. In particular, short-time Fourier transforms (STFTs) can be used to considerably reduce the temporal dimension of the representation for an underlying signal. This is a critical advantage, as training Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs) on long sequences remains challenging due to unstable gradients and computational requirements of backpropagation through time (BPTT) [Hochreiter, 1991, Bengio et al., 1994]. Applying the STFT on the raw signal, on the other hand, is computationally efficient, as in practice it is implemented with the Fast Fourier Transform (FFT) whose computational complexity is $\mathcal{O}(n\,log(n))$.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

The aforementioned biological, representational and computational considerations provide compelling motivations for designing learning models for tasks where the complex-valued representation of the input and output data is more desirable than their real-counterpart.

Recent work has provided building blocks for deep complex-valued neural networks [Trabelsi et al., 2017]. These building blocks have been shown, in many cases, to avoid numerical problems during training and thereby enable the use of complex-valued representations. These representations are well-suited for frequency domain signals, as they have the capacity to explicitly encode frequency magnitude and phase components. In particular, models built with these building blocks have excelled at tasks such as automatic music transcription, spectrum prediction [Trabelsi et al., 2017], speech enhancement [Choi et al., 2019], as well as MRI reconstruction [Dedmari et al., 2018].

Recently, Choi et al. [2019] have designed a deep complex U-Net for *speech enhancement*. This task consists of separating clean speech from noise when the noisy speech is given as input. A different but related task is that of separating the speech of multiple speakers into separate signals. This motivates us to build and improve upon the deep complex U-Net for the task of *speech separation*. From this work, our contributions are summarized as follows:

1. We present a new signal extraction method based on Feature-wise Linear Modulation (FiLM) [Perez et al., 2018] to create multiple separated candidates for each of the signals we aim to retrieve from a mixture of inputs. A signal averaging operation on the candidates is performed in order to increase the robustness of the signal to noise and interference. Before the averaging procedure, a dropout is implemented on the signal candidates in order to reduce the amount of interference and noise correlation existing between the different candidates. Our extraction method could be seen as one performing local ensembling. In the case of audio source separation, we aim to retrieve distinct audio signals associated with each speaker in the input mix. The candidates are averaged in order to obtain the final separated speech for each of the speakers in question. Our experiments demonstrate the efficacy of our proposed masking method, and show its regularizing effect.
2. We propose and explore a new frequency-domain loss taking **explicitly** into account the magnitude and phase of signals. A key characteristic of our loss is that it is scale- and time-invariant. Our comparative analysis (**See section 6.9 in the appendix**) related to different phase-aware losses defined in time and frequency domains demonstrates the advantage of our proposed loss.

2 Connection to Signal Processing and Motivation for FiLM and Signal Averaging

A clean signal $\mathbf{s}$ corrupted by the environment impulse response $\mathbf{r}$ and an additive noise ϵ can expressed as $\mathbf{y} = \mathbf{s} \circledast \mathbf{r} + \epsilon$, where $\circledast$ denotes the circular convolution operator. By leveraging the convolution theorem and the linearity of the Fourier transform we get :

$$\mathcal{F}(\mathbf{y}) = \mathcal{F}(\mathbf{s}) \odot \mathcal{F}(\mathbf{r}) + \mathcal{F}(\epsilon), \tag{1}$$

where $\mathcal{F}$ denotes the Fourier transform and $\odot$ the complex element-wise multiplication. If we want to retrieve the spectral information of the clean signal $\mathbf{s}$, we can express it as:

$$\mathcal{F}(\mathbf{s}) = \left[\mathcal{F}(\mathbf{y}) \odot \frac{1}{\mathcal{F}(\mathbf{r})}\right] - \frac{\mathcal{F}(\epsilon)}{\mathcal{F}(\mathbf{r})}, \tag{2}$$

where $\frac{1}{\mathcal{F}(\mathbf{r})}$ and $-\frac{\mathcal{F}(\epsilon)}{\mathcal{F}(\mathbf{r})}$ are respectively scaling and shifting representations. These representations could easily be inferred using FiLM [Perez et al., 2018] as it conditionally learns scaling $\mathbf{\Gamma}$ and shifting $\mathbf{B}$ representations. To be more rigorous, we can assume in the case of speech separation that, for each speaker, there exists an impulse response such that when it convolved with the clean speech of the speaker, it allows to reconstruct the mix. We would then have:

$$\begin{aligned} \mathbf{mix} &= \mathbf{s}_i \circledast \mathbf{r}_i + \epsilon_i \quad \forall i \in \{1, ..., \text{Nb speakers}\} \\ &\Rightarrow \mathcal{F}(\mathbf{s}_i) = \mathcal{F}(\mathbf{mix}) \odot \frac{1}{\mathcal{F}(\mathbf{r}_i)} - \frac{\mathcal{F}(\epsilon_i)}{\mathcal{F}(\mathbf{r}_i)} \\ &\Rightarrow \mathcal{F}(\mathbf{s}_i) = \mathcal{F}(\mathbf{mix}) \odot \mathbf{\Gamma}_i + \mathbf{B}_i. \end{aligned} \tag{3}$$

Now, let's assume that $\mathbf{y}$ is a stochastic process such that $\mathbf{y} = \mathbf{x} + \epsilon$, where ϵ is the noise component which mean $\mathbb{E}[\epsilon] = 0$. $\mathbf{x}$ is the clean signal that we want to estimate such that $\mathbf{x}$ is constant for all

observations and that an i^{th} observation of $\mathbf{y}$ is given by $\mathbf{y}_i = \mathbf{x} + \epsilon_i$. The signal-to-noise ratio (SNR), which is a measure of the signal quality, is defined as the ratio of the power of a clean signal to the power of noise, i.e, SNR $= \frac{\mathbb{E}[|\mathbf{x}|^2]}{\mathbb{E}[|\epsilon_i|^2]}$. Estimating the clean speech $\mathbf{x}$ by approximating $\mathbb{E}[\mathbf{y}]$ allows to discard the noise component as $\mathbb{E}[\mathbf{y}] = \mathbf{x}$. In that case $\hat{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x} + \epsilon_i) = \mathbf{x} + \frac{1}{N}\sum_{i=1}^{N}\epsilon_i$. The SNR would then be: SNR $= \frac{\mathbb{E}[|\mathbf{x}|^2]}{\mathbb{E}[|\frac{1}{N}\sum_{i=1}^{N}\epsilon_i|^2]} = \frac{\mathbb{E}[|\mathbf{x}|^2]}{\frac{1}{N^2}\mathbb{E}[|\sum_{i=1}^{N}\epsilon_i|^2]}$. If ϵ_i are uncorrelated, $\mathbb{E}[|\sum_{i=1}^{N}\epsilon_i|^2] = \sum_{i=1}^{N}\mathbb{E}[|\epsilon_i|^2] = N\,\mathbb{E}[|\epsilon_i|^2] \Rightarrow$ SNR $= N\frac{\mathbb{E}[|\mathbf{x}|^2]}{\mathbb{E}[|\epsilon_i|^2]}$. This shows that the signal averaging operation and the uncorrelated noises allows to increase the SNR by a factor of N. If we want to approximate $\mathcal{F}(\mathbf{s}_i)$ by performing signal averaging, we would then have:

$$
\begin{aligned}
\mathbb{E}[\mathcal{F}(\mathbf{s}_i)] &= \mathcal{F}(\mathbf{mix}) \odot \mathbb{E}[\mathbf{\Gamma}_i] + \mathbb{E}[\mathbf{B}_i] \\
\Rightarrow \widehat{\mathbb{E}[\mathcal{F}(\mathbf{s}_i)]} &= \mathcal{F}(\mathbf{mix}) \odot \widehat{\mathbb{E}[\mathbf{\Gamma}_i]} + \widehat{\mathbb{E}[\mathbf{B}_i]} \\
&= \mathcal{F}(\mathbf{mix}) \odot \frac{1}{N}\sum_{j=1}^{N}\mathbf{\Gamma}_{ij} + \frac{1}{N}\sum_{j=1}^{N}\mathbf{B}_{ij},
\end{aligned}
\tag{4}
$$

where $\mathcal{F}(\mathbf{mix})$ is constant. In equation 4, N is equal to the number of scaling and shifting representations generated to approximate respectively each of $\mathbb{E}[\mathbf{\Gamma}_i]$ and $\mathbb{E}[\mathbf{B}_i]$.

3 Amplitude and Phase-aware Loss

In Choi et al. [2019] a weighted version of the cosine similarity is proposed in order to maximize the signal-to-distortion ratio (SDR) proposed in Vincent et al. [2006]. Recall that cosine similarity loss is defined in the real-valued domain and it is given by the following equation:

$$
\cos_{\text{time}}(\mathbf{y}, \mathbf{x}) = \frac{-\sum_i x_i \circ y_i}{||\mathbf{x}|| \cdot ||\mathbf{y}||},
\tag{5}
$$

where $\circ$ denotes the element-wise real-valued multiplication operation. Both $\mathbf{y}$ and $\mathbf{x}$ are real-valued in the above equation as $\mathbf{y}$ is the target signal in the temporal domain and $\mathbf{x}$ is the estimated signal after performing an inverse STFT on the spectrogram. The phase is then taken **implicitly** into account as the real-valued target signal encodes inherently the phase of the spectrogram. As the task in Choi et al. [2019] is speech enhancement (which is different from ours as we are performing speech separation), the authors used a weighted version of the $\cos_{\text{time}}$ loss to weight the part of the loss corresponding to the speech signal and also the complementary part corresponding to the noise signal. This weighting is performed according to their respective target energies. In our case we are interested in extracting the clean speech signals of all the involved speakers whether each speaker signal has either high or low energy in the mixture. This is why we are not interested in penalizing the retrieved speech of each speaker by its corresponding energy.

Here, we suggest the use of a loss function which explicitly takes into account both magnitude and phase. This is accomplished by computing the inner product, between the reference signal and its estimate, in the complex plane. In fact computing the inner product in the frequency domain is equivalent to computing the cross correlation in the time domain followed by a weighted average. The inner product in the frequncy domain is then, shift invariant. The complex inner product between 2 signals is given by the following equation:

$$
\langle \mathbf{x}|\mathbf{y} \rangle = \sum_j [\Re(x_j)\Re(y_j) + \Im(x_j)\Im(y_j)] + i\,[\Re(x_j)\Im(y_j) - \Im(x_j)\Re(y_j)].
\tag{6}
$$

If $\mathbf{x}$ and $\mathbf{y}$ are identical, which is equivalent of having $||\mathbf{x}|| = ||\mathbf{y}||$ and $\angle\mathbf{x} = \angle\mathbf{y}$, then, $\langle \mathbf{x}|\mathbf{y} \rangle = ||\mathbf{y}||^2 + 0i$. If $\mathbf{x}$ and $\mathbf{y}$ are parallel, then $\frac{\langle \mathbf{x}|\mathbf{y} \rangle}{||\mathbf{x}|| \cdot ||\mathbf{y}||} = 1 + 0i = 1$. The inner product between the 2 signals normalized by the product of their amplitudes, is then scale and time invariant. We chose a loss that maximizes the real part of that normalized inner product and minimizes the square of its imaginary part. Note that each of the real and imaginary parts of the normalized inner product lies between [-1, 1]. To understand more how the complex inner product is both amplitude and phase aware, how the real part of equation (6) is responsible of the amplitude similarity between the reference and estimate signals and how the imaginary part of the same equation is responsible for the

phase matching between them, see section 6.2 in the appendix. We define the following similarity loss denoted by $\mathbb{C}\mathrm{SimLoss}$ as:

$$\mathbb{C}\mathrm{SimLoss}(\mathbf{x}, \mathbf{y}) = -\lambda_{real}\Re\left(\frac{\langle \mathbf{x}|\mathbf{y}\rangle}{||\mathbf{x}|| \cdot ||\mathbf{y}||}\right) + \lambda_{imag}\Im^2\left(\frac{\langle \mathbf{x}|\mathbf{y}\rangle}{||\mathbf{x}|| \cdot ||\mathbf{y}||}\right), \tag{7}$$

where λ_{real} and λ_{imag} are penalty constants. We fixed λ_{real} to 1 in all our experiments. We tried different values of $\lambda_{imag} \in \{10^2, 10^3, 10^4\}$. $\lambda_{imag} = 10^4$ worked the best. All the results are reported in Table 1 and Table 2 for $\mathbb{C}\mathrm{SimLoss}$ correpond to $\lambda_{imag} = 10^4$.

4 Complex Mask Generation

Important Note: **All the details about the architecture used can be found in the in section 6.1 in the appendix. The details about the conducted experiments and the empirical analysis can be found in section 6.9 in the appendix.** Featurewise Linear Modulation (FiLM) [Perez et al., 2018] techniques have yielded impressive results in visual question answering (VQA). The FiLM approach applies an affine transformation to convolutional feature maps, given the embedding of the question. In our approach, we create multiple transformations of the complex input spectrogram using FiLM. The FiLM parameters are determined from the output of our U-Net (**See Figure 1**). We then generate a complex mask for the original input spectrogram as well as for each of the FiLM-transformed spectrograms. This is accomplished by using a ResNet conditioned on the U-Net output, the spectrogram and its FiLM transformations. Each spectrogram is multiplied by its corresponding complex mask. This leads to multiple candidates for the separated speech of each speaker. The resulting outputs are averaged to produce the final estimated clean speech. This could be interpreted as a local ensembling procedure to estimate the clean speech of the different speakers. More precisely, given the output of the last upsampling block of the U-Net, we generate scaling matrices Γ_j and shift matrices B_j, $j \in [1, C]$ of the same size as the input mix spectrogram. These parameters operate on the input mix as described by the following equation:

$$\mathrm{input\_transformation}_j = \Gamma_j \otimes \mathrm{inputmix} + \mathrm{B}_j, \tag{8}$$

where Γ_j and B_j are functions of the output of the last upsampling block in the U-Net, and $\otimes$ is the elementwise complex product. In our case, we used a simple complex convolution layer with a kernel of size 3×3 to generate Γ_j and B_j. The original input mix and its C scaled and shifted transformations together form $C + 1$ representations of the input mix. Given these $C + 1$ complex representations, we generate $C + 1$ corresponding complex masks, with which the representations are then multiplied. These masks are generated by a sequence of a complex convolution layer which kernel size is 3×3 followed by two residual blocks. Once we have performed the complex multiplication of the masks with their respective inputs, $C + 1$ separated speech candidates are obtained for a given speaker. This procedure is repeated for the maximum number of speakers that could exist in an input mix. The main motivation for this process is to increase the separation capability and reduce interference between the separated speakers. Each transformation can focus on a specific pattern in the representation; Thereafter, as Danihelka et al. [2016] suggest, we can include and exclude candidates in order to keep specific patterns in the speech while removing unwanted ones. Each mask corresponding to a specific input transformation can be seen as a feature of the speaker embedding. Grouped together, the masks generated to retrieve the speech of a given speaker could be interpreted as an embedding identifying the speaker. The complex masking procedure is summarized in **Algorithm 1 in section 6.3 in the appendix**.

5 Conclusion

In this work, we introduced a new complex-valued framework for signal retrieval and signal separation in the Fourier domain. As a case sudy, we considered audio source separation. We proposed a new masking method based on a complex-valued version of the Feature-wise Linear Modulation (FiLM) model, allowing to perform local ensembling and yielding a beneficial regularization effect. We also proposed a new phase-aware loss taking, explicitly, into account the magnitude and phase of the reference and estimated signals. In our study, phase proved to be an important factor that should be taken into account in order to improve the quality of the separation in terms of SDR. The phase-aware loss improves over other frequency and time-domain losses. Our deep separator draws its power from

the compelling properties of complex-valued neural networks and the proposed masking method. Our finding might shed light on the deep complex-valued neural networks' tendency to solve challenging tasks where the data lie in the complex space and where it could be represented in the frequency domain. We view these results as an opportunity to pursue a more systematic investigation of the underpinning of complex-valued representation success. We believe that our proposed method could lead to new research directions where signal retrieval is needed.

References

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. *ICML*, 2015.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pages 1120–1128, 2016.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 1994.

Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

Zhuo Chen, Yi Luo, and Nima Mesgarani. Deep attractor network for single-microphone speaker separation. *CoRR*, abs/1611.08930, 2016.

Hyeong-Seok Choi, Janghyun Kim, Jaesung Huh, Adrian Kim, Jung-Woo Ha, and Kyogu Lee. Phase-aware speech enhancement with deep complex u-net. *arXiv preprint arXiv:1903.03107*, 2019.

Ivo Danihelka, Greg Wayne, Benigno Uria, Nal Kalchbrenner, and Alex Graves. Associative long short-term memory. *ICML*, 2016.

Muneer Ahmad Dedmari, Sailesh Conjeti, Santiago Estrada, Phillip Ehses, Tony Stöcker, and Martin Reuter. Complex fully convolutional neural networks for mr image reconstruction. In *International Workshop on Machine Learning for Medical Image Reconstruction*, pages 30–38. Springer, 2018.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. The importance of skip connections in biomedical image segmentation. In *Deep Learning and Data Labeling for Medical Applications*, pages 179–187. Springer, 2016.

J. Du, Y. Tu, Y. Xu, L. Dai, and C. Lee. Speech separation of a target speaker based on deep neural networks. In *Proc. of ICSP*, pages 473–477, 2014.

N. Q. K. Duong, E. Vincent, and R. Gribonval. Under-determined reverberant audio source separation using a full-rank spatial covariance model. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2010.

Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T. Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *CoRR*, 2018.

H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *Proc. of ICASSP*, pages 708–712, 2015.

Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the beta-divergence. *CoRR*, abs/1010.1763, 2010.

Cédric Févotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural Computation*, 2009.

Ruohan Gao, Rogério Schmidt Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. *CoRR*, 2018.

George M Georgiou and Cris Koutsougeras. Complex domain backpropagation. *IEEE transactions on Circuits and systems II: analog and digital signal processing*, 1992.

Klaus Greff, Rupesh K Srivastava, and Jürgen Schmidhuber. Highway and residual networks learn unrolled iterative estimation. *arXiv preprint arXiv:1612.07771*, 2016.

D. Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

John R. Hershey and Michael Casey. Audio-visual sound separation via hidden markov models. 2002.

John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. *CoRR*, 2015.

Akira Hirose. *Complex-valued neural networks: theories and applications*. World Scientific, 2003.

Sepp Hochreiter. *Untersuchungen zu dynamischen neuronalen Netzen*. PhD thesis, 1991.

Guoning Hu and DeLiang Wang. Monaural speech segregation based on pitch tracking and amplitude modulation. *Trans. Neur. Netw.*, 2004.

Po-Sen Huang, Kim Minje, Mark Hasegawa-Johnson, and Paris Smaragdis. Deep learning for monaural speech separation. *ICASSP*, 2014.

A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 2000.

Stanisław Jastrzebski, Devansh Arpit, Nicolas Ballas, Vikas Verma, Tong Che, and Yoshua Bengio. Residual connections encourage iterative inference. *arXiv preprint arXiv:1710.04773*, 2017.

Cijo Jose, Moustpaha Cisse, and Francois Fleuret. Kronecker recurrent units. *arXiv preprint arXiv:1705.10142*, 2017.

Taehwan Kim and Tülay Adalı. Approximation by fully complex multilayer perceptrons. *Neural computation*, 2003.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Yuan-Shan Lee, Chien-Yao Wang, Shu-Fan Wang, Jia-Ching Wang, and Chung-Hsien Wu. Fully complex deep neural network for phase-incorporating monaural source separation. In *ICASP*, 2017.

Antoine Liutkus, Derry Fitzgerald, Zafar Rafii, Bryan Pardo, and Laurent Daudet. Kernel additive models for source separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2014.

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

Yi Luo and Nima Mesgarani. Tasnet: Surpassing ideal time-frequency masking for speech separation. *arXiv preprint arXiv:1809.07454*, 2018.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). 1983.

Tohru Nitta. Orthogonality of decision boundaries in complex-valued neural networks. *Neural Computation*, 2004.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. 2018.

Tony Plate. Holographic reduced representations: Convolution algebra for compositional distributed representations. In *IJCAI*, pages 30–35, 1991.

Tony A Plate. Holographic reduced representations. *IEEE Transactions on Neural networks*, 6(3): 623–641, 1995.

David P Reichert and Thomas Serre. Neuronal synchrony in complex-valued deep networks. *ICLR*, 2014.

Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. In *Advances in neural information processing systems*, pages 2449–2457, 2015.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. pages 234–241, 2015.

Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. A probabilistic latent variable model for acoustic modeling. In *In Workshop on Advances in Models for Acoustic Processing at NIPS*, 2006.

Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. Supervised and semi-supervised separation of sounds from single-channel mixtures. In *International Conference on Independent Component Analysis and Signal Separation*, pages 414–421. Springer, 2007.

Martin Spiertz. Source-filter based clustering for monaural blind source separation. *International Conference on Digital Audio Effects*, 2009.

N. Sturmel and L. Daudet. Signal reconstruction from stft magnitude: A state of the art. In *In Proc. of the International conference on digital audio effects*, 2006.

Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. *ICLR*, 2017.

Shrikant Venkataramani and Paris Smaragdis. End-to-end networks for supervised single-channel speech separation. *CoRR*, abs/1810.02568, 2018.

Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE Trans. Audio, Speech & Language Processing*, 14(4):1462–1469, 2006.

Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *Trans. Audio, Speech and Lang. Proc.*, 2007.

Beiming Wang and Mark Plumbley. Investigating single-channel audio source separation methods based on non-negative matrix factorization. *ICA Research Network International Workshop*, 2006.

DeLiang Wang and Jitong Chen. Supervised speech separation based on deep learning: An overview. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.

Zhong-Qiu Wang, Jonathan Le Roux, DeLiang Wang, and John R. Hershey. End-to-end speech separation with unfolded iterative phase reconstruction. *CoRR*, abs/1804.10204, 2018.

Chao Weng, Dong Yu, Michael L Seltzer, and Jasha Droppo. Deep neural networks for single-channel multi-talker speech recognition. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 23(10):1670–1679, 2015.

Moritz Wolter and Angela Yao. Fourier rnns for sequence analysis and prediction. *arXiv preprint arXiv:1812.05645*, 2018a.

Moritz Wolter and Angela Yao. Gated complex recurrent neural networks. *arXiv preprint arXiv:1806.08267*, 2018b.

Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen. Permutation invariant training of deep models for speaker-independent multi-talker speech separation. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 241–245. IEEE, 2017.

Richard S Zemel, Christopher KI Williams, and Michael C Mozer. Lending direction to neural networks. *NIPS*, 1995.

Jiong Zhang, Yibo Lin, Zhao Song, and Inderjit S Dhillon. Learning long term dependencies via fourier recurrent units. *arXiv preprint arXiv:1803.06585*, 2018.

Michael Zibulevsky and Barak A. Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neural Computation*, 2001.

6 Appendix

6.1 Deep Complex Speech Separation

We detail here the deep complex architecture to perform speech separation. For this, we rely on the U-Net architecture proposed by Ronneberger et al. [2015] and the complex-valued building blocks proposed by Trabelsi et al. [2017]. In our proposed architecture, we incorporated residual connections inside the U-Net blocks and we replaced the complex batch normalization with complex layer normalization, as the model was unable to learn with the former technique and yielded instabilities during training. The details of the complex layer normalization technique and the reasons of its outperformance compared to complex batchnorm are discussed in the appendix in section 6.4. After that, we describe the steps of our novel complex masking method which is based on a complex-valued version of Feature-wise Linear Modulation (FiLM) [Perez et al., 2018], which we designed, and allows to perform local ensembling.

6.1.1 Complex Residual U-Net

identity connections [He et al., 2016] have had a significant impact on image segmentation. These architectural elements have also been combined with U-Nets [Drozdzal et al., 2016] for image segmentation. In our case, we use simple basic complex residual blocks (Figure 2) inside each of the U-Net encoding and decoding paths (Figure 1) . Figure 2 (Left) and (Middle) illustrate the basic structure of our Residual U-Net upsampling and downsampling blocks (U_i and D_i) used in Figure 1, while Figure 2 (Right) illustrates the structure of the complex residual blocks used in Figure 2 (Left) and Figure 2 (Middle). Each U-Net block begins with a downsampling block (in the encoding U-Net path) or an upsampling block (in the decoding U-Net path). It also contains a block that doubles the number of feature maps (in the encoding path), or halves them (in the decoding path). The upsampling, downsampling, doubling and halving blocks each applies successively a complex layer normalization, a $\mathbb{C}$ReLU and a complex convolution to their inputs. All complex convolutions have a kernel size of 3×3 except for the case of a downsampling block, where the convolution layer has a kernel size of 1×1 and a stride of 2×2. In the case of upsampling, we use bilinear interpolation instead of transposed convolution because we found empirically that it yielded better results. Immediately before and immediately after the doubling / halving blocks, we use $k = 1$ or $k = 2$ residual blocks. We have opted for this residual U-Net block architecture because of memory constraints and because residual connections are believed to perform inference through iterative refinement of representations [Greff et al., 2016, Jastrzebski et al., 2017].

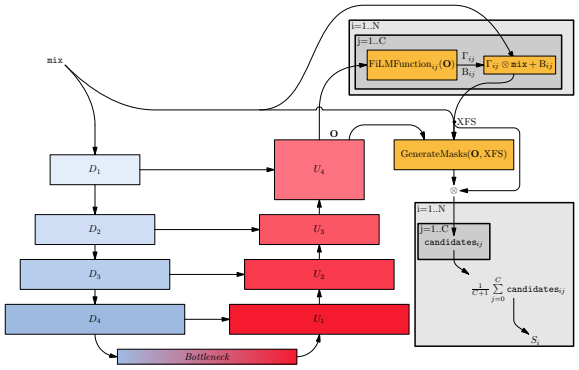

Figure 1: **The architecture of our Deep Complex Separator that we call Deep Pitcher**. It consists of a pipeline containing a U-Net and a Complex FiLMed Masking operator (see Algorithm 1). The Deep Pitcher takes as input the mixed speech signal which is fed to the U-Net. The downsampling blocks of the U-Net are denoted by D_i where $i \in \{1, 2, 3, 4\}$ and the upsampling blocks are denoted by U_i where $i \in \{1, 2, 3, 4\}$. The output of the U-Net along with the input mix are then fed to the Complex FiLMed Masking operator in order to estimate the clean speech for each of the speakers.

6.2 Details about the Amplitude and Phase-aware Loss

We show here that solving the two-equation system, assimilating the real part of the inner product, between the two signals $\mathbf{x}$ and $\mathbf{y}$, to the square of the amplitude of $\mathbf{y}$, and canceling its imaginary part, amounts to canceling the differences in amplitude and phase between $\mathbf{x}$ and $\mathbf{y}$, respectively (See equation 11). For this we will use the

following trigonometric properties:

$$\begin{cases} \cos(\theta_x)\cos(\theta_y) &= \frac{1}{2}\cos(\theta_x-\theta_y) + \frac{1}{2}\cos(\theta_x+\theta_y) \\ \sin(\theta_x)\sin(\theta_y) &= \frac{1}{2}\cos(\theta_x-\theta_y) - \frac{1}{2}\cos(\theta_x+\theta_y) \\ \cos(\theta_x)\sin(\theta_y) &= \frac{1}{2}\sin(\theta_x+\theta_y) - \frac{1}{2}\sin(\theta_x-\theta_y) \\ \sin(\theta_x)\cos(\theta_y) &= \frac{1}{2}\sin(\theta_x+\theta_y) + \frac{1}{2}\sin(\theta_x-\theta_y), \end{cases} \tag{9}$$

where $\theta_x, \theta_y \in \mathbb{R}$. For simplicity of notation, we will denote a complex-valued target scalar as y and a its estimate as x instead of $\hat{y}$:

$$\begin{aligned} y &= |y|e^{i\theta_y} = |y|\left[\cos(\theta_y) + i\sin(\theta_y)\right] \in \mathbb{C} \\ x &= |y|e^{i\theta_x} = |x|\left[\cos(\theta_x) + i\sin(\theta_x)\right] \in \mathbb{C}. \end{aligned} \tag{10}$$

θ_y and θ_x are the corresponding phases of the reference y and its complex estimate x respectively.Resolving the system of equations below is equivalent of having both magnitude and phase of the reference and estimation identical **OR** when y is 0. Recall that $\Re(\langle \mathbf{x}|\mathbf{y}\rangle) = \sum_j [\Re(x_j)\Re(y_j) + \Im(x_j)\Im(y_j)]$ and $\Im(\langle \mathbf{x}|\mathbf{y}\rangle) = \sum_j [\Re(x_j)\Im(y_j) - \Re(y_j)\Im(x_j)]$.

$$\begin{cases} \Re(x)\Re(y) + \Im(x)\Im(y) - |y|^2 = 0 \\ \Re(x)\Im(y) - \Re(y)\Im(x) = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} \Re(x)\Re(y) + \Im(x)\Im(y) = \Re(y)^2 + \Im(y)^2 \\ \Re(x)\Im(y) = \Re(y)\Im(x) \end{cases}$$

$$\Leftrightarrow \begin{cases} |x|\cos(\theta_x)\,|y|\cos(\theta_y) + |x|\sin(\theta_x)\,|y|\sin(\theta_y) = |y|^2 \\ |x|\cos(\theta_x)\,|y|\sin(\theta_y) = |y|\cos(\theta_y)\,|x|\sin(\theta_x) \end{cases}$$

$$\Leftrightarrow \begin{cases} |x|\,|y|\left[\cos(\theta_x)\cos(\theta_y) + \sin(\theta_x)\sin(\theta_y)\right] = |y|^2 \\ |x|\,|y|\left[\cos(\theta_x)\sin(\theta_y) - \cos(\theta_y)\sin(\theta_x)\right] = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} |y|\left[\,|x|\left(\cos(\theta_x)\cos(\theta_y) + \sin(\theta_x)\sin(\theta_y)\right) - |y|\,\right] = 0 \\ |x| = 0 \text{ OR } |y| = 0 \text{ OR } \left[\cos(\theta_x)\sin(\theta_y) - \cos(\theta_y)\sin(\theta_x)\right] = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} |y| = 0 \text{ OR } |x|\left(\cos(\theta_x)\cos(\theta_y) + \sin(\theta_x)\sin(\theta_y)\right) = |y| \\ |x| = 0 \text{ OR } |y| = 0 \text{ OR } \cos(\theta_x)\sin(\theta_y) = \cos(\theta_y)\sin(\theta_x) \end{cases}$$

$$\Leftrightarrow \begin{cases} |y| = 0 \text{ OR } |x|\left(\frac{1}{2}\cos(\theta_x-\theta_y) + \frac{1}{2}\cos(\theta_x+\theta_y) + \frac{1}{2}\cos(\theta_x-\theta_y) - \frac{1}{2}\cos(\theta_x+\theta_y)\right) = |y| \\ |x| = 0 \text{ OR } |y| = 0 \text{ OR } \frac{1}{2}\sin(\theta_x+\theta_y) - \frac{1}{2}\sin(\theta_x-\theta_y) = \frac{1}{2}\sin(\theta_x+\theta_y) + \frac{1}{2}\sin(\theta_x-\theta_y) \end{cases}$$

$$\Leftrightarrow \begin{cases} |y| = 0 \text{ OR } |x|\cos(\theta_x-\theta_y) = |y| \\ |x| = 0 \text{ OR } |y| = 0 \text{ OR } -\sin(\theta_x-\theta_y) = \sin(\theta_x-\theta_y) \end{cases}$$

$$\Leftrightarrow \begin{cases} |y| = 0 \text{ OR } |x|\cos(\theta_x-\theta_y) = |y| \\ |x| = 0 \text{ OR } |y| = 0 \text{ OR } \theta_x - \theta_y \equiv 0 \,(\mathrm{mod}\,\pi) \end{cases}$$

$$\Leftrightarrow \begin{cases} |y| = 0 \text{ OR } |x|\cos(\theta_x-\theta_y) = |y| \\ |x| = 0 \text{ OR } |y| = 0 \text{ OR } -\sin(\theta_x-\theta_y) = \sin(\theta_x-\theta_y) \end{cases}$$

$$\Leftrightarrow \begin{cases} |y| = 0 \text{ OR } |x|\cos(k\pi) = |y|,\ k \in \mathbb{Z} \\ |x| = 0 \text{ OR } |y| = 0 \text{ OR } \theta_x = \theta_y + k\pi,\ k \in \mathbb{Z} \end{cases}$$

$$\Leftrightarrow \begin{cases} |y| = 0 \text{ OR } |x|\cos(2k'\pi) = |y| \text{ OR } |x|\cos((2k'+1)\pi) = |y| = 0 \,(\text{because}\cos((2k'+1)\pi) = -1) \\ |x| = 0 \text{ OR } |y| = 0 \text{ OR } \theta_x = \theta_y + k'\pi,\ k' \in \mathbb{Z} \end{cases}$$

$$\Leftrightarrow \begin{cases} |y| = 0 \text{ OR } |x| = |y| \text{ OR } |x| = |y| = 0 \\ |x| = 0 \text{ OR } |y| = 0 \text{ OR } \theta_x = \theta_y + 2k'\pi,\ k' \in \mathbb{Z} \end{cases}$$

$$\Leftrightarrow \begin{cases} |y| = 0 \text{ OR } |x| = |y| \\ \theta_x = \theta_y + 2k'\pi,\ k' \in \mathbb{Z}. \end{cases} \tag{11}$$

Now, a solution corresponding to a null reference vector $\boldsymbol{y}$ could be problematic as it leads to an infinite number of choices for the estimated signal x. In fact, Choi et al. [2019] mentioned this issue and chose to work with a cosine similarity-based function in order to learn from noisy-only data. This is why it is more convenient to work with the normalized inner product loss.

6.3 Complex FiLMed Masking Algorithm

Algorithm 1 Complex FiLMed Masking

Input: *U-Net output:* O
Input: *Nb transformations (XFs):* C
Input: *Nb speakers:* N
Input: *Input Mix:* mix
Output: *Speakers separated speeches:* $S_1, ..., S_N$

function $\mathbb{C}$-FiLMed Masking(O, C, N, mix)
  for $i \leftarrow 1$ **to** N **do**
    $\Gamma_{i1} ... \Gamma_{iC}$, $\mathrm{B}_{i1} ... \mathrm{B}_{iC} \leftarrow$ FilMFunction(O)
  end for
  XFS $\leftarrow [\,]$
  for $i \leftarrow 1$ **to** N **do**
    for $j \leftarrow 1$ **to** C **do**
      $\mathrm{XF}_{ij} \leftarrow \Gamma_{ij} \otimes \mathrm{mix} + \mathrm{B}_{ij}$
      XFS_i.append(XF_{ij})
    end for
  end for
  XFS $\leftarrow$ concatenate($\mathrm{XFS}_{11}, ..., \mathrm{XFS}_{NC}$)
  masks $\leftarrow$ GenerateMasks(O, XFS)
  candidates $\leftarrow$ masks $\otimes$ XFS
  cleanspeeches $\leftarrow [\,]$
  for $i \leftarrow 1$ **to** N **do**
    $\mathrm{cleanspeech}_i \leftarrow$ average(candidates$[(C+1) \times (i-1) + 1 : (C+1) \times i]$))
    $\mathrm{cleanspeeches}$.append($\mathrm{cleanspeech}_i$)
  end for
  return cleanspeeches
end function

6.4 Complex Layer Normalization

Just as in complex batch normalization, complex layer normalization consists in whitening 2D vectors by left-multiplying the $\mathbf{0}$-centered data $\big(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}]\big)$ by the inverse square root of the 2×2 covariance matrix $\boldsymbol{V}$. $\tilde{\boldsymbol{x}} = (\boldsymbol{V})^{-\frac{1}{2}} \big(\boldsymbol{x} - \mathbb{E}[\boldsymbol{x}]\big)$, where the covariance matrix $\boldsymbol{V}$ is

$$\begin{aligned}\boldsymbol{V} &= \begin{pmatrix} V_{rr} & V_{ri} \\ V_{ir} & V_{ii} \end{pmatrix} \\ &= \begin{pmatrix} \mathrm{Cov}(\Re\{\boldsymbol{x}\}, \Re\{\boldsymbol{x}\}) & \mathrm{Cov}(\Re\{\boldsymbol{x}\}, \Im\{\boldsymbol{x}\}) \\ \mathrm{Cov}(\Im\{\boldsymbol{x}\}, \Re\{\boldsymbol{x}\}) & \mathrm{Cov}(\Im\{\boldsymbol{x}\}, \Im\{\boldsymbol{x}\}) \end{pmatrix}.\end{aligned}$$

Complex layer normalization is distinguished from complex batch normalization by its computation of the mean and covariance statistics over the *layer* features instead of the *batch* instances. This allows us, as in the real-valued version, to avoid estimating batch statistics during training. An intuition for batch normalization's inappropriateness is related to the sparsity, in both time and frequency domains, of speech. This is reflected in the spectrograms. Speech is temporally halting and restarting, and spectrally consists of at most a few simultaneously-held fundamentals and their discrete overtones. Mixing few speakers does not significantly change this property.

In the light of this observation, it stands to reason that statistics computed across a batch's multiple utterance mixtures are almost meaningless. Speakers within and across utterance mixtures are not controlled for volume, nor can their pauses be meaningfully aligned. Batch statistics will therefore be inappropriately driven by the mixture with the most simultaneous speakers, the loudest speaker(s), or the speaker(s) with the "dirtiest" spectrogram. Finally, in the absence of any speech, batch statistics will inappropriately boost background noise to a standardized magnitude.

The above motivates the use of exclusively intra-sample normalization techniques like Layer Normalization for speech data. Batch normalization is more appropriate for natural images, which are dense, both in space and frequency.

In addition to the fact that intra-sample normalization is more appropriate for speech signals, CLN ensures a more robust normalization of data when the number of feature maps is sufficiently large. In fact, according to the weak law of large numbers, as the sample size increases, the sample statistics approximate their expected values. Therefore, when the number of feature maps far exceeds the number of batch instances, we obtain more robust estimates because they converge, in probability, to the corresponding expected values.

6.5 Figures

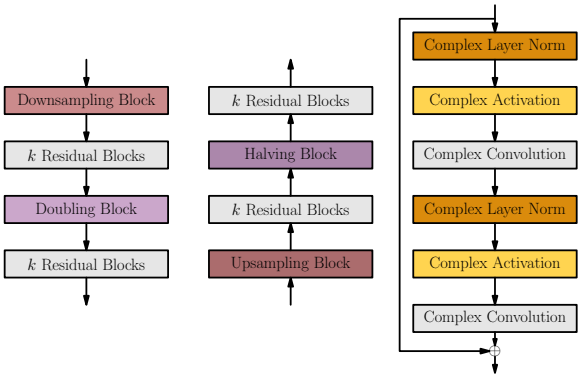

Figure 2: The basic structures of our U-Net downsampling block D_i (Left) and our U-Net upsampling block U_i (Middle) used respectively in the encoding and the decoding paths of Figure 1. The structure of a basic complex residual block (Right) in each of D_i and U_i.

6.6 Data Pre-processing and Training Details

The speech mixtures are generated using the procedure adopted in Erdogan et al. [2015], Wang et al. [2018]. More precisely, the training set consists of 30 hours of two-speaker mixtures that were generated by randomly selecting sentences (uttered by different speakers) from the Wall Street Journal WSJ0 training set called `si_tr_s`. The signals are then mixed with different amplitude factors, leading signal-to-noise ratios (SNR) ranging between 0 dB and 5 dB. Using the same method, we also generated 10 hours of validation set. The test set is composed of 5 hours that were generated similarly using utterances from the different speakers belonging to the WSJ0 development set `si_dt_05`. The data sampling rate is 8KHz. Regarding the STFT parameters, a Hann window of size 256 and a hop length equal to 128 are used.

Table 1 (see section 6.9) and Table 2 contain the results for the experiments conducted using the Wall Street Journal dataset. All models in Tables 1 (see section 6.9) and 2 were trained using the backpropagation algorithm with Stochastic Gradient Descent with Nesterov momentum [Nesterov, 1983] set at 0.9. The gradient norm was clipped to 1. We used the learning rate schedule described in Trabelsi et al. [2017]. In order to warm up the model during training, a constant learning rate of 0.01 was fixed for the first 10 epochs. From epoch 10 to 100, the learning rate was increased to 0.1. Later, an annealing of the learning rates by a factor of 10, at epochs, 120 and 150 was performed. We ended up the training at epoch 200. Models in Table 1(see section 6.9) have been trained using a batch size of 40. Models in Table 2 have been trained using a batch size of 24 to fit in the GPU memory. All the models have been trained in parallel using 8 V100 GPUs. For all the tested losses, we used the Permutation Invariant Training criterion knows as PIT [Yu et al., 2017]. The PIT criterion allows to take into account all possible assignments between the target signals and the estimated clean speeches. This is done by computing all possible permutations between the targets and the estimated clean speeches. During training, the assignment with the minimal loss is considered for backpropagation. This is due to the fact that for the synthetically mixed input, the order of the target speakers is randomly chosen and it doesn't satisfy a specific criterion. This random order in the target speakers causes the well-known label permutation problem [Hershey et al., 2015, Weng et al., 2015]. The PIT criterion allows then to reduce significantly this problem by considering the output-target assignment yielding the minimal training loss. During inference, we assume that the model has learned to produce output that does not permute speeches. (Yu et al. [2017] mention that output-to-speaker assignment may change across time frames. This would have the effect of decreasing the Signal to Noise Ratio (SNR) and the Signal to Distortion Ratio (SDR) as it causes interference of speakers speeches.

6.7 Related work on Learning Representations in the Fourier Domain

Leveraging the Convolution Theorem to retrieve information has been done decades ago in the machine learning community using Holographic Reduced Representations (HRRs) in the context of associative memories [Plate,

1991, 1995]. HRRs enables one to store key-value data. Retrieving a value in the data associated with a given key could be performed by convolving the whole data with the key or by applying an inner product between these two. By applying a Fast Fourier Transform (FFT) on the keys and the data, one could perform elementwise multiplication between their Fourier transforms and apply an inverse FFT to convert the result to the time domain. This would be equivalent to performing convolution between the key and the data in the time domain and has the advantage of being less expensive. Recently, Danihelka et al. [2016] have used associative memories to augment the capacity of LSTMs and to increase their robustness to noise and interference. For that, they applied independent permutations on the memory to create multiple copies of it. This enables one to obtain decorrelated noise in each of the permuted copies. A complex multiplication is then performed between the key and each of the copies. A signal averaging on the resulted multiplications eliminates the decorrelated noise in them and strengthens the Signal-To-Noise ratio (SNR) of the retrieved signal. Danihelka et al. [2016], however, have not relied on FFTs in order to convert the temporal signals to the frequency domain. In fact, they assumed that complex-valued multiplication between the key and the data is itself enough to perform retrieval, and they have assumed that for each input representation the first half is real and the second one is imaginary.

During this decade, interest in Fourier domain representations has started to grow in the machine learning community. Bruna et al. [2013] introduced a generalization of convolutions to graphs using the Graph Fourier Transform, which is in turn defined as the multiplication of a graph signal by the eigenvector matrix of the graph Laplacian. However, the computation of the eigenvector matrix is expensive. Recently, methods that are computationally more efficient have been introduced in Defferrard et al. [2016] and Kipf and Welling [2016] to avoid an explicit use of the Graph Fourier basis. In the context of Convolutional Neural Networks (CNNs), Rippel et al. [2015] introduced spectral pooling, which allows one to perform pooling in the frequency domain. This enables maintaining the output spatial dimensionality, and thus retaining significantly more information than other pooling approaches. Rippel et al. [2015] have also observed that the parametrization of the convolution filters in the Fourier domain induces faster convergence during training. Arjovsky et al. [2016] designed a recurrent neural network (RNN) where the transition hidden matrix is unitary. More precisely, the hidden transition matrix is constructed using the product of specific unitary transformations such as diagonal matrices, permutations, rotations, the Discrete Fourier Transform and its inverse. This allows preserving the norm of the hidden state, and as a consequence, avoids the problem of vanishing and exploding gradients. Wolter and Yao [2018a] designed an RNN where the input is converted to the frequency domain using a Short Time Fourier Transform (STFT). The output is converted back to the time domain by applying an inverse STFT. Zhang et al. [2018] proposed a Fourier Recurrent Unit (FRU) where they showed that FRU has gradient lower and upper bounds independent of the temporal dimension. They have also demonstrated the great expressivity of the sparse Fourier basis from which the FRU draws its power.

As we consider the task of speech separation as case study, we provide a related work section on speech separation in section 6.8 in the appendix.

6.8 Related Work on Speech Separation Methods

Speech separation has been the subject of extensive study within the audio processing literature for a considerable amount of time. Early attempts at disentangling different speakers in an audio source have typically involved either pure audio inputs that are isolated from any sort of disturbance or special microphone configurations for strong supervision [Duong et al., 2010]. Strong supervision in the form of isolated and clean recordings of individual sound, while effective, can be hard to achieve due to the difficulty in gathering large enough quantities of natural data. Subsequent work in this area assumed in some cases monophonic audio signals [Huang et al., 2014, Smaragdis et al., 2007, Spiertz, 2009, Virtanen, 2007, Wang and Plumbley, 2006] on which some well-known matrix decomposition techniques were applied, such as Independent Component Analysis [Hyvärinen and Oja, 2000], Sparse Decomposition [Zibulevsky and Pearlmutter, 2001], Non-negative Matrix Factorization [Févotte et al., 2009, Févotte and Idier, 2010, Liutkus et al., 2014], and Probabilistic Latent Variables models [Smaragdis et al., 2006]. Despite their adoption, matrix decomposition approaches have several limitations. One notable example is that many of these approaches operate on the frequency domain without taking into account the phase component of the signals, as the methods themselves are real-valued. Moreover, for large enough quantities of recordings, performing matrix decomposition can be computationally prohibitive and the decomposition task can be sensitive to the fixed number of spectral bases chosen to represent the signal in question.

More recently, there has been growing interest in leveraging deep learning techniques [Huang et al., 2014, Hershey et al., 2015, Gao et al., 2018, Ephrat et al., 2018] to tackle the speech separation problem. Methods that have been proposed thus far can be grouped in two categories, audio-only and audio-visual speech separation methods. As our work fits the audio-only category, we provide below a related-work section discussing these methods. A section discussing audio-visual methods can be found in section 6.8.2.

6.8.1 Audio-only Speech Separation Methods

In this section, we discuss some neural speech separation methods that rely on audio information only. Our work falls into this category of methods. To the best of our knowledge, Huang et al. [2014] were the first to explore the use of deep learning applied to monaural speech separation. Their system is based on a combination of a feed-forward and a recurrent network, that are jointly optimized with a soft masking function. A closely-related work has been concurrently proposed by Du et al. [2014], where a neural network is trained to estimate the log power spectrum of the target speakers.

Hershey et al. [2015] proposed a deep clustering approach to speech separation. The basic idea is to learn high-dimensional embeddings of the mixture signals, that is later exploited to separate the speech targets with standard clustering techniques. A recent attempt to extend deep clustering led to the deep attractor network proposed by Chen et al. [2016]. Similarly to deep clustering, high dimensional embeddings are learned, but the network also creates the so-called “attractors" to better cluster time-frequency points dominated by different speakers.

The aforementioned approaches estimate only the magnitude of the STFTs and reconstruct the time-domain signal with the Griffin-Lim algorithm [Griffin and Lim, 1984] or other similar procedures [Sturmel and Daudet, 2006]. Similarly to our work, other papers have recently proposed to integrate the phase-information within a speech separation system. The work by Erdogan et al. [2015], for instance, proposes to train a deep neural network with a phase-sensitive loss. Another noteworthy attempt has been described in Wang et al. [2018], where the neural network still estimates the magnitude of the spectrum, but the time-domain speech signals are retrieved by directly integrating the Griffin-Lim reconstruction into the neural layers. Instead of explicitly integrating phase-information, other recent work perform speech separation in the time domain directly, as described in Venkataramani and Smaragdis [2018]. Likewise, The TasNet architectures proposed in Luo and Mesgarani [2017] and Luo and Mesgarani [2018] perform speech separation using the mixed time signal as input.

The studies by [Lee et al., 2017, Hu and Wang, 2004, Huang et al., 2014] are more closely related to our work as they address the speech separation problem taking into account phase information. However, this was done without leveraging the recent advances in complex-valued deep learning. The authors in Lee et al. [2017] address the audio-source separation problem using a fully complex-valued deep neural network that learns the nonlinear relationship between an input sound and its distinct sources. Both the activations and weights of the network are complex-valued. For a more comprehensive review of most of these techniques, we refer the readers to Wang and Chen [2018].

6.8.2 Related Work on Audio-Visual Speech Separation Methods

These methods exploit visual information in video in order to perform audio source separation. Two recent works in this category include Gao et al. [2018] and Ephrat et al. [2018] where they leverage large collections of unannotated "in-the-wild" videos to train deep neural networks. The task is to separate object sounds and isolate or enhance the speech of the desired speakers. In Gao et al. [2018] specifically, the authors perform non-negative matrix factorization on the audio channel in order to discover latent sound representations for each physical object detected. Then, they train a multi-instance multi-label neural network to map the spectral audio bases to the distribution of detected visual objects. The multi-instance multi-label learning problem is performed in order to address the label permutation issue described in Hershey and Casey [2002]. One drawback with this two-step approach is that the objects in the videos are detected without guidance from the subsequent classification task, thus, some key information could be lost, as a result of overfitting. The authors in Ephrat et al. [2018] introduced a speaker-independent audio-visual model where they combine a feed-forward convolutional model and a bidirectional LSTM to jointly extract visual features of distinct speakers and their corresponding audio signal. While both Gao et al. [2018] and Ephrat et al. [2018] leverage visual and auditory signals as a means to achieve high separation quality, in our work, we rely however on audio source only.

6.9 Experiments

Table 1: Speech separation experiments on two speakers using the standard setup with the Wall Street Journal corpus.
We explore different real and complex-valued model variants and report the test SDR. k is the the number of residual blocks used inside the residual U-Net block (See Figure 1). Start Fmaps is the number of feature maps in the first layer of the encoding path in the U-Net. The Start Fmaps defines the width of each of the successive layers in the model. We respectively double and half the size of the layers in each of the successive downsampling and upsampling stages. The effective number of feature maps for a complex feature map is equal to the number of reported feature maps $\times$ 2. This is due to the fact that it has a real and an imaginary part. The number of parameters is expressed in millions. The number of input mixture transformations is also reported. Test SDR scores for different time and spectral domain losses are inserted in the last column.

MODEL	k	START FMAPS	PARAMS	TRANSFORMS	LOSS FUNCTION	TEST SDR
REAL U-NET	1	64	8.45	0	L2_{freq}	4.59
REAL U-NET	2	64	14.76	0	L2_{freq}	7.92
COMPLEX U-NET	1	32	4.29	0	L2_{freq}	9.61
COMPLEX U-NET	2	32	7.4	0	L2_{freq}	**9.70**
COMPLEX U-NET	2	40	11.55	0	L2_{freq}	**10.30**
COMPLEX U-NET	2	40	11.55	0	$\mathbb{C}$SimLoss	10.21
COMPLEX U-NET	2	40	11.55	0	L2_{time}	9.31
COMPLEX U-NET	2	40	11.55	0	cos_{time}	9.34
COMPLEX U-NET	2	40	11.57	5	L2_{freq}	10.58
COMPLEX U-NET	2	40	11.57	5	$\mathbb{C}$SimLoss	**10.87**
COMPLEX U-NET	2	40	11.57	5	L2_{time}	10.31
COMPLEX U-NET	2	40	11.57	5	cos_{time}	10.14
COMPLEX U-NET	2	40	11.61	10	L2_{freq}	10.59
COMPLEX U-NET	2	40	11.61	10	$\mathbb{C}$SimLoss	**10.90**
COMPLEX U-NET	2	40	11.61	10	L2_{time}	10.86
COMPLEX U-NET	2	40	11.61	10	cos_{time}	10.47
COMPLEX U-NET	2	40	11.67	15	L2_{freq}	**10.93**
COMPLEX U-NET	2	40	11.67	15	$\mathbb{C}$SimLoss	10.91
COMPLEX U-NET	2	40	11.67	15	L2_{time}	10.66
COMPLEX U-NET	2	40	11.67	15	cos_{time}	10.74

We tried different configurations combining unitary and standard complex initializations. All of these initializations have been proposed by Trabelsi et al. [2017]. It turned out that the best configuration is obtained when using a complex standard initialization for all layers, except for the convolutional layer, generating the FiLM parameters, and the first convolutional layer in the generating mask function which precedes the 2 residual blocks. For the above-mentioned convolutional layers a unitary initialization respecting the He criterion [He et al., 2015] was applied. This is not surprising as a unitary weight matrix $\in \mathbb{C}^{d \times d}$ constitutes a basis of $\mathbb{C}^d$. Therefore any complex-valued vector in $\mathbb{C}^d$, such as those representing the FiLM parameters or the masks, could be generated using a linear combination of the row vectors of that unitary matrix.

In Tables 1 and 2 we experiment with architectures that use different number of mixture transformations. Adding mixture transformations does not significantly increase the number of parameters compared to the size of the whole model. In the case where 15 transformations are adopted, the number of parameters is increased by less than 1% of the total number.

Since Table 1's first row contains baselines, they exclude our proposed masking method and loss. These baselines (both real and complex) are architecturally the same as the U-Net of Figure 1, without the FiLM, the GenerateMask and the averaging operation. A real counterpart of a complex model is one where the convolution and the normalization layers are real, the nonlinearity is plain ReLU and He init is used for the weights. The real and complex U-Nets output the masks which are complex multiplied with the mix in order to infer the clean speech of the speakers. All the complex models, whether they have approximately the same number of parameters ($\mathbb{R}$:8.45M $\approx$ $\mathbb{C}$:7.4M), half ($\mathbb{R}$:8.45M; $\mathbb{C}$:4.39M) or a third, with half the depth ($\mathbb{R}$:14.76M; $\mathbb{C}$:4.39M) outperformed by a large margin their real counterparts. This shows that whether the comparison is fair, or even where advantages in terms of capacity and depth are given to the real network, it doesn't perform as well as complex models when it comes to process complex input and infer complex output. Thus, we will no longer

Table 2: Experiments on two speaker speech separation using the standard setup with the Wall Street Journal corpus. We explore different numbers of input mixture transformations and different dropout rates on the latter using the training losses defined in the spectral domain. The losses in questions are $\mathrm{L2}_{\mathrm{freq}}$ and $\mathbb{C}\mathrm{SimLoss}$. The number of parameters is expressed in millions. All tested models contain 44 feature maps in the first downsampling layer of the U-Net instead of 40 in Table 1. The same number of $k = 2$ residual blocks is used inside the basic structure of the residual U-Net block. SDR scores are shown in the last column.

PARAMS	TRANSFORMS	DROPOUT	LOSS FUNCTION	TEST SDR
13.97	0	0	$\mathrm{L2}_{\mathrm{freq}}$	9.88
13.99	5	0	$\mathrm{L2}_{\mathrm{freq}}$	10.11
14.03	10	0	$\mathrm{L2}_{\mathrm{freq}}$	10.91
14.09	15	0	$\mathrm{L2}_{\mathrm{freq}}$	9.92
13.97	0	0	$\mathbb{C}\mathrm{SimLoss}$	9.87
13.99	5	0	$\mathbb{C}\mathrm{SimLoss}$	10.64
14.03	10	0	$\mathbb{C}\mathrm{SimLoss}$	**11.05**
14.09	15	0	$\mathbb{C}\mathrm{SimLoss}$	10.82
13.99	5	0.1	$\mathrm{L2}_{\mathrm{freq}}$	10.54
14.03	10	0.1	$\mathrm{L2}_{\mathrm{freq}}$	10.72
14.09	15	0.1	$\mathrm{L2}_{\mathrm{freq}}$	10.91
13.99	5	0.1	$\mathbb{C}\mathrm{SimLoss}$	10.96
14.03	10	0.1	$\mathbb{C}\mathrm{SimLoss}$	**11.34**
14.09	15	0.1	$\mathbb{C}\mathrm{SimLoss}$	11.22
13.99	5	0.2	$\mathrm{L2}_{\mathrm{freq}}$	10.67
14.03	10	0.2	$\mathrm{L2}_{\mathrm{freq}}$	10.90
14.09	15	0.2	$\mathrm{L2}_{\mathrm{freq}}$	10.90
13.99	5	0.2	$\mathbb{C}\mathrm{SimLoss}$	11.23
14.03	10	0.2	$\mathbb{C}\mathrm{SimLoss}$	**11.29**
14.09	15	0.2	$\mathbb{C}\mathrm{SimLoss}$	11.23
13.99	5	0.3	$\mathrm{L2}_{\mathrm{freq}}$	10.71
14.03	10	0.3	$\mathrm{L2}_{\mathrm{freq}}$	10.06
14.09	15	0.3	$\mathrm{L2}_{\mathrm{freq}}$	10.91
13.99	5	0.3	$\mathbb{C}\mathrm{SimLoss}$	**11.21**
14.03	10	0.3	$\mathbb{C}\mathrm{SimLoss}$	11.12
14.09	15	0.3	$\mathbb{C}\mathrm{SimLoss}$	11.06
13.99	5	0.4	$\mathrm{L2}_{\mathrm{freq}}$	10.72
14.03	10	0.4	$\mathrm{L2}_{\mathrm{freq}}$	10.74
14.09	15	0.4	$\mathrm{L2}_{\mathrm{freq}}$	10.83
13.99	5	0.4	$\mathbb{C}\mathrm{SimLoss}$	11.09
14.03	10	0.4	$\mathbb{C}\mathrm{SimLoss}$	11.08
14.09	15	0.4	$\mathbb{C}\mathrm{SimLoss}$	**11.12**

focus on real-valued models, but, instead, will concentrate on transformations and losses that are appropriate for complex-valued models.

Three major observations can be inferred from the numbers displayed in Table 1: 1- Wider and deeper models improve the quality of separation in terms of SDRs; 2- The increase in the number of input transformations has a positive impact on the task of separating audio sources, as additional input transformations achieve higher SDR scores; 3- For a given number of input transformations, the best results are obtained with losses computed in the spectral domain. For all the experiments reported in Table 1, either the $\mathbb{C}\mathrm{SimLoss}$ or the $\mathrm{L2}_{\mathrm{freq}}$ achieve the highest SDR.

The scores reported in Table 1 show that the local ensembling procedure is beneficial to the task of speech separation. This rewarding impact is confirmed in all experiments of Table 2 (See also Figure 3). As mentioned in section 4, each mask could be seen as a feature of the speaker embedding and the generated masks together constitute the whole embedding. Performing dropout on the masks might then allow to perform regularization for the retrieval and separation mechanism. Dropping out a mask is equivalent to a dropout of input mixture transformations or clean speech candidates. Since spectral loss functions yielded higher SDRs than their time-domain counterparts, we adopted them to evaluate the effect of applying different dropout rates to the input

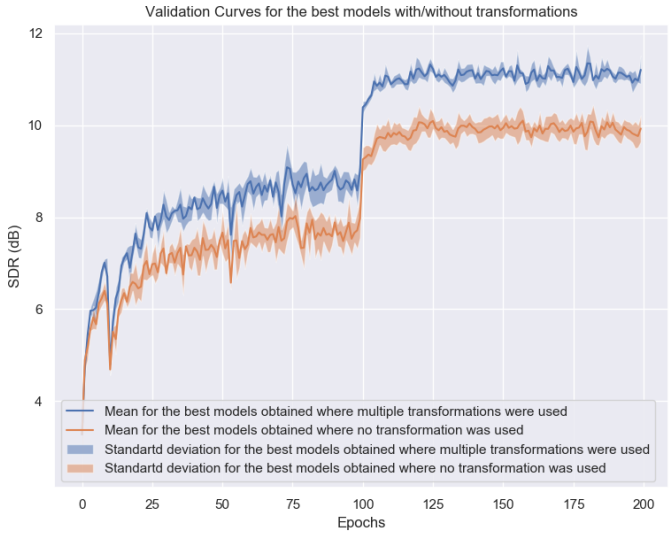

Figure 3: **Validation curves of models with and without performing multiple input transformations**. The plotted curves relate to models reported in Table 2. Models with multiple input transformations outperform those without transformations. The former achieved higher SDR scores, on average.

transformations. Wider and deeper models with Start Fmaps = 44 and k=2 residual blocks are tested in the conducted experiments. Results are reported in Table 2.

In the absence of dropout and multiple transformations, we observe from the results displayed in Table 2, that wider models are not necessarily more beneficial to the separation task. The SDRs reported in the case of no mixture transformations are 9.88 and 9.87 for the wider model. These SDR scores correspond to the $\mathrm{L2}_{\mathrm{freq}}$ and $\mathbb{C}\mathrm{SimLoss}$ losses respectively. However, for the narrower models, SDRs of 10.30 and 10.21 were respectively reported for the same losses in Table 1. This means that wider models have the potential to overfit. On the other hand, if input transformations are taken into account, a jump in the SDR is observed. When 10 input transformations are introduced, SDR scores of 11.05 and 10.90 are recorded with the $\mathbb{C}\mathrm{SimLoss}$ and the $\mathrm{L2}_{\mathrm{freq}}$ losses, respectively. Lower SDR performances are recorded when ensembling is implemented with mixtures of 5 and 15 transformations, respectively. This means that the local ensembling procedure is acting as a regularizer. However, a tradeoff in terms of the number of input transformations (and so in terms of clean speech candidates) has to be made as increasing the number of input transformations might worsen the performance of the model and lead to overfitting.

Dropping out the speech candidates using a small probability rate has a further regularization effect on the wider model. This could be inferred from the results reported in Table 2 (See also Figure 5). We employed different dropout rates varying from 0 to 0.4. A rate of 0.1 yielded the best result as it caused a jump of SDR score from 11.05 to 11.34. It is important to emphasize again the importance of having a compromise in terms of the number of transformations. For instance, for most of the dropout rates we experimented, a number of 10 mixture transformations yielded the highest SDRs. In all the experiments reported in Table 2, the $\mathbb{C}\mathrm{SimLoss}$ clearly outperformed the $\mathrm{L2}_{\mathrm{freq}}$ (See Figure 4). In fact, regardless of the dropout rate and the number of input transformations employed, for wider models using the $\mathrm{L2}_{\mathrm{freq}}$ training loss function, the SDR score did not cross the threshold of 10.91 dB. The highest SDR score obtained, when using the $\mathrm{L2}_{\mathrm{freq}}$ loss function, is 10.93. This value corresponds to a narrower model with 15 input transformations (see Table 1).

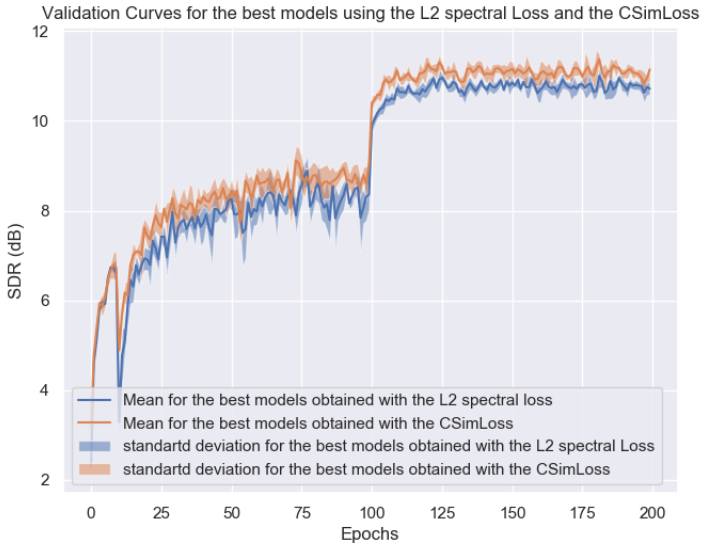

Figure 4: Validation curves of the models that yielded the highest SDRs using either the L2 spectral loss or our $\mathbb{C}$SimLoss. The Drawn curves are related to models reported in Table 2.

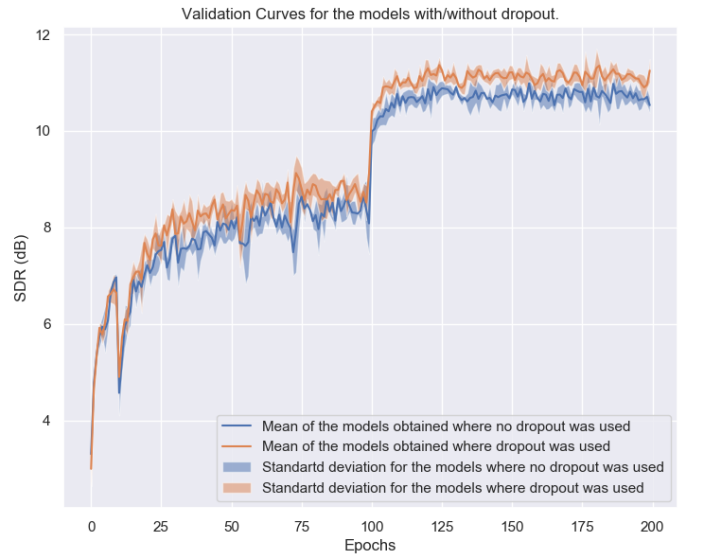

Figure 5: Validation curves of the models that yielded the highest SDRs for both cases where dropout on the input mixture transformations was used and where it was not. The Drawn curves are related to models reported in Table 2.