

---

# Improving DNN-based Music Source Separation using Phase Features

---

Joachim Muth<sup>1</sup> Stefan Uhlich<sup>2</sup> Nathanaël Perraudin<sup>3</sup> Thomas Kemp<sup>2</sup> Fabien Cardinaux<sup>2</sup> Yuki Mitsufuji<sup>4</sup>

## Abstract

Music source separation with deep neural networks typically relies only on amplitude features. In this paper we show that additional phase features can improve the separation performance. Using the theoretical relationship between STFT phase and amplitude, we conjecture that derivatives of the phase are a good feature representation opposed to the raw phase. We verify this conjecture experimentally and propose a new DNN architecture which combines amplitude and phase. This joint approach achieves a better signal-to-distortion ratio on the DSD100 dataset for all instruments compared to a network that uses only amplitude features. Especially, the bass instrument benefits from the phase information.

## 1. Introduction

*Music source separation* (MSS) refers to the problem of obtaining instrument estimates  $\hat{\mathbf{s}}_j(n) \in \mathbb{R}^I$  from the mixture

$$\mathbf{x}(n) = \sum_{j \in \mathcal{J}} \mathbf{s}_j(n), \quad (1)$$

where  $n$  denotes the discrete time index,  $I$  gives the number of channels and  $\mathcal{J}$  is the set of instruments. A common setup is the extraction of  $\mathcal{J} := \{\text{bass, drums, vocals, other}\}$  from stereo mixtures, i.e.,  $I = 2$ . This setup was used for the last *SiSEC* contests on MSS (Ono et al., 2015; Liutkus et al., 2017; Stöter et al., 2018) and is also the basis of our work.

This paper studies the appropriateness of the *short-time Fourier transform* (STFT) phase as an input feature for MSS systems based on *deep neural networks* (DNNs). Current

state-of-the-art approaches perform MSS by only considering the mixture STFT amplitude from which they estimate the target instrument STFT amplitude (Huang et al., 2014a;b; Uhlich et al., 2015; Nugraha et al., 2016; Uhlich et al., 2017; Takahashi & Mitsufuji, 2017; Takahashi et al., 2018b). It is well known that the STFT phase contains useful information for speech enhancement, see e.g. (Gerkmann et al., 2015) and, therefore, should not be neglected. Recent attempts have been made in order to improve MSS using phase. (Lee et al., 2017) proposed a fully complex-valued DNN, which predicts the complex STFT of the target instrument from the complex mixture STFT. (Dubey et al., 2017) analyzed whether phase is beneficial as input feature for a DNN compared to a network using only amplitude. Another approach is (Takahashi et al., 2018a), which estimates the phase of the instrument from the mixture amplitude and phase by treating the phase retrieval problem as a classification problem. This paper presents another approach where phase is used as an additional input feature to improve the amplitude estimation. In contrast to (Dubey et al., 2017), we propose a special architecture to exploit the information that is present in the STFT phase as a simple concatenation of amplitude and phase at the input of the network yields trained networks that focus only on amplitude information.

We first show that expressing the phase through its *instantaneous frequency* (time derivative) and its *group delay* (frequency derivative) greatly improves the efficiency of DNNs compared to networks fed with raw phase inputs. This is done by looking at experimental results as well as studying the theoretical relationship between phase and amplitude of a continuous-time STFT. Moreover, we demonstrate that the discrete-time STFT introduces systematic shifts into the phase and that correcting these shifts improves the efficiency of the DNN to exploit these features.

Finally, we design a network architecture which takes full advantage of this additional feature. It is formed by two independent networks, taking respectively amplitude and phase, whose outputs are concatenated afterwards through a dense layer. Intuitively, each network independently extracts features from amplitude and phase and forwards them to a fusion layer, which reconstructs the spectrum based on these features. With the suggested data pre-processing method and architecture design, our system achieves on average a relative improvement of 2.3% and up to 6% for

<sup>1</sup>École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland <sup>2</sup>Sony European Technology Center (EuTEC), Stuttgart, Germany <sup>3</sup>Swiss Data Science Center, EPFL and ETH Zürich, Switzerland <sup>4</sup>Sony Corporation, Audio Technology Development Department, Tokyo, Japan. Correspondence to: Joachim Muth <j Joachim.h.muth@gmail.com>, Stefan Uhlich <stefan.uhlich@sony.com>.

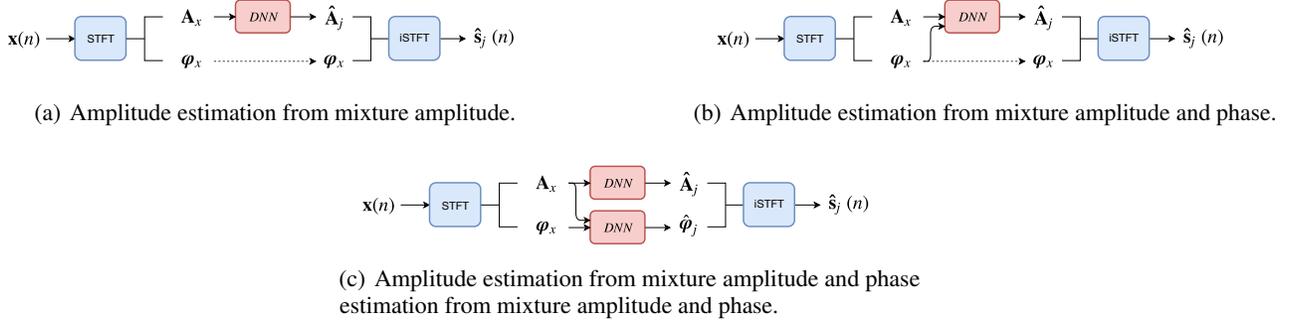


Figure 1. Comparison of different MSS approaches.

Table 1. Comparison of upper limits reachable by two different approaches. Results on DSD100 test set (SDR in dB).

Instrument	DNN <sub>A</sub> using approach (a)	Upper baseline of approach (b)	Upper baseline of approach (c)
		(A <sub>IRM</sub> & $\varphi_{\text{mixture}}$ )	(A <sub>DNN</sub> & $\varphi_{\text{oracle}}$ )
Bass	3.24	7.92	6.59
Drums	4.68	8.53	6.15
Other	3.54	8.19	5.35
Vocals	4.78	11.10	6.86

bass compared to an amplitude-only system.

For clarity, the paper is divided into two parts. Sec. 2 is dedicated to the properties of the phase as a feature for MSS. In this section, we consider the problem of using only the phase for estimating the instrument amplitude. This allows us to better understand this feature and the development of an appropriate pre-processing method. Sec. 3, in contrast, considers both amplitude and phase from the mixture signal in order to produce an improved estimate of the instrument amplitude, which is the ultimate goal of the paper.

## 2. Phase as Input Feature

### 2.1. Motivation

Fig. 1 shows three different approaches for MSS, where  $\mathbf{A}(k, m) \in \mathbb{R}_+^I$  and  $\varphi(k, m) \in [-\pi, \pi]^I$  denote the STFT amplitude and phase at frequency bin index  $k$  and frame index  $m$ .  $\hat{s}_j(n) \in \mathbb{R}^I$  is the estimated target instrument signal.

Typically, approach (a) is used where the instrument amplitude  $\hat{\mathbf{A}}_j$  is estimated from the mixture amplitude  $\mathbf{A}_x$ , while the instrument phase is simply approximated by the mixture phase  $\varphi_x$ . The estimated instrument  $\hat{s}_j(n)$  is produced by applying an inverse STFT with the estimated source amplitude  $\hat{\mathbf{A}}_j$  and mixture phase  $\varphi_x$ .

Approaches (b) and (c) show two different ways to improve upon (a). Approach (b), which was, e.g., used in (Dubey

et al., 2017), is similar in all respects except that the mixture phase is used to improve the instrument amplitude estimation. Approach (c), which was, e.g., followed by (Takahashi et al., 2018a), estimates the instrument phase  $\hat{\varphi}_j$  which can then be used for the inverse STFT.

In order to choose between the two possible improvements, we did a simple experiment shown in Table 1. We compare the upper limits achievable by both strategies: on one side a signal synthesized with the *ideal ratio mask* (IRM) amplitude and the mixture phase; on the other side the oracle phase and the amplitude estimation from the network DNN<sub>A</sub><sup>1</sup>. We can see that approach (b) has more room for improvement as the upper limit achievable has a relative improvement of 122%. In contrast, the upper limit of approach (c) allows an average relative improvement of 57%, which indicates that currently the amplitude estimation is still the main bottleneck for MSS performance. We therefore investigate approach (b) in this paper.

### 2.2. Theoretical Relationship

Interestingly, for the continuous-time STFT

$$X(\omega, t) = A(\omega, t)e^{j\varphi(\omega, t)} \quad (2)$$

of a continuous-time signal  $x(t)$ , there is a theoretical relationship between the amplitude  $A(\omega, t)$  and the phase  $\varphi(\omega, t)$ . The continuous-time STFT is given by

$$X(\omega, t) = e^{j\omega t/2} \int_{-\infty}^{\infty} x(u)h(t-u)e^{-j\omega u} du. \quad (3)$$

Using a Gaussian window  $h(t) = \lambda^{-1/2}\pi^{-1/4}e^{-t^2/(2\lambda^2)}$ ,

<sup>1</sup>DNN<sub>A</sub> is a network which estimates the instrument amplitude from the mixture amplitude. Please refer to Sec. 3 for more details about this network.

(Auger et al., 2012) showed that

$$\frac{\partial}{\partial t} \varphi(\omega, t) = \lambda^{-2} \frac{\partial}{\partial \omega} \log(A(\omega, t)) + \frac{\omega}{2}, \quad (4a)$$

$$\frac{\partial}{\partial \omega} \varphi(\omega, t) = -\lambda^2 \frac{\partial}{\partial t} \log(A(\omega, t)) - \frac{t}{2}. \quad (4b)$$

From (4), we can see that the derivatives of phase and log-magnitude are linked and, therefore, we hope that the amplitude estimation for our target instrument from the mixture phase can be improved by using phase features.

Furthermore, we conjecture from (4) that better results for the amplitude estimation can be obtained if we work with time/frequency derivatives of the phase instead of the raw phase. This intuition will be experimentally confirmed in Sec. 2.5. As we work with discrete-time signals, we will approximate the derivatives by differences, i.e., in the following we will use

$$\Delta_t \varphi := \varphi(k, m) - \varphi(k, m - 1), \quad (5a)$$

$$\Delta_f \varphi := \varphi(k, m) - \varphi(k - 1, m). \quad (5b)$$

Please note that from the phase information, we are able to recover the amplitude up to an unknown scale. This can be seen from considering a signal  $s(n) \in \mathbb{C}$  and a scaled version  $s'(n) = a \cdot s(n)$  with  $a > 0$  as  $\angle s(n) = \angle s'(n)$  whereas  $|s(n)| \neq |s'(n)|$ . Hence, the phase only contains information about variations of the amplitude. This property is consistent with (4) which links phase and log-amplitude through their derivatives.

### 2.3. Shifts in discrete Short-Time Fourier Transform

Fig. 2 shows the distribution of  $\Delta_t \varphi = \varphi(k, m) - \varphi(k, m - 1)$  for consecutive frequency bins. We can observe a systematic offset in the statistical distribution which can be explained by the shift theorem of the *discrete Fourier transform* (DFT) (Smith, 2007). It states that a delay in the time domain results in a linear phase term in the frequency domain, i.e.,

$$x(n - n_0) \xrightarrow{DFT} e^{j \frac{2\pi}{N} k n_0} X(k), \quad (6)$$

where  $n_0$  is the shift and  $N$  the DFT/FFT size.

Therefore, in the case of a stationary signal transformed by an STFT, with hop size  $n_0$ , the phase of two consecutive frequency bins is expected to be shifted by a term

$$\text{phase shift} = -\frac{2\pi}{N} k n_0 \quad (7)$$

For example, an overlap of 75% results in a shift of  $-k \frac{\pi}{2}$  which can also be seen in Fig. 2.

DNNs are known to be sensitive to the feature distribution and, therefore, this shift should be properly compensated for during the pre-processing stage, as described in Sec. 2.4, in order to ensure a proper training of the DNN.

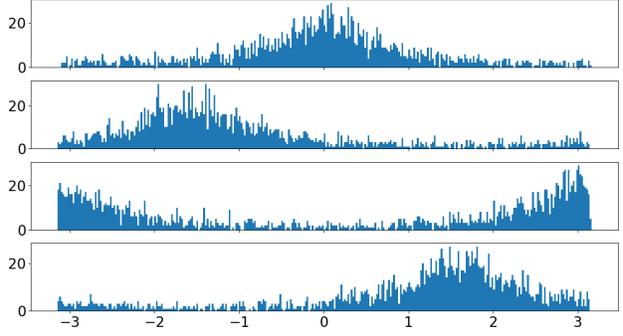


Figure 2. Distribution of *instantaneous frequencies* ( $\Delta_t \varphi$ ) over all time frames for one song from DSD100. From top to bottom, four successive frequency bins are considered and the histograms show a shift of  $-k \frac{\pi}{2}$  introduced by the STFT with FFT size  $N = 4096$  and hop size  $n_0 = 1024$  (overlap of 75%).

### 2.4. Pre-processing

According to the conclusions drawn in Sec. 2.2 and Sec. 2.3, we apply the following pre-processing steps to the raw phase:

- The time and frequency derivatives are first approximated by the difference between two consecutive time frames ( $\Delta_t \varphi$ ) and by the difference between two consecutive frequency bins ( $\Delta_f \varphi$ ), respectively.
- A linear term  $2\pi k \frac{n_0}{N}$  is added to the time differences in order to compensate for the effect described in Sec. 2.3. Consequently, for a stationary signal  $\Delta_t \varphi = 0$ .
- For  $\Delta_f \varphi$ , we could empirically observe a systematic shift of  $\pi$  in its statistical distribution, see Fig. 3 (a). We compensate it by subtracting  $\pi$  in order to obtain  $\mathbb{E}(\Delta_f \varphi) = 0$ .
- Finally, all values are wrapped to  $[-\pi, \pi)$  using

$$\Delta \varphi = ((\Delta \varphi + \pi) \bmod 2\pi) - \pi. \quad (8)$$

The effects of this pre-processing method on feature statistical distribution are illustrated in Fig. 3.

### 2.5. Experimental Validation

In order to see whether our pre-processing is effective, we run two experiments, which we now describe in detail.

The network used to evaluate the suggested pre-processing method is formed by two dense layers of 500 hidden units, intersected by ReLU non-linearities and completed by a dense output layer matching the target dimensions. At the very end, a bias layer initialized with the average amplitude per frequency bin over the training set shifts the output and a ReLU non-linearity ensures non-negative output values. We use a context of five preceding/succeeding frames as

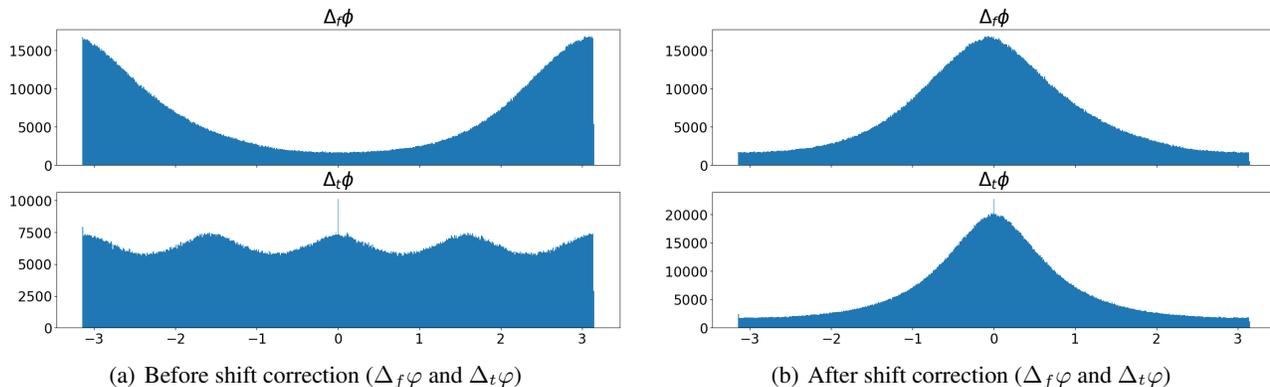


Figure 3. Statistical distribution of the group delays ( $\Delta_f\varphi$ ) and instantaneous frequencies ( $\Delta_t\varphi$ ) before and after shift correction.

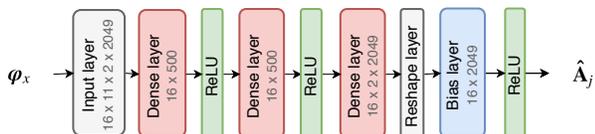


Figure 4. Network architecture with phase feature only.

temporal context. Fig. 4 shows the network structure and the overall MSS framework is summarized in Fig. 5.

We now give the results for estimating the STFT amplitude from the phase using a DNN. By these experiments, we show that it is advantageous to consider the time/frequency derivatives instead of the raw phase. Furthermore, the pre-processing described in Sec. 2.4, is also shown to be relevant.

In the first experiment, we reconstruct the instrument amplitude from the instrument phase. Thus, we do not consider a separation problem but instead focus on the ability of a DNN to recover a signal knowing its phase. By this, we can compare different phase feature representations and observe their effects on the DNN learning power. The training MSE curves are shown in Fig. 6. We can observe that the phase derivatives show the best performance as we previously conjectured.

In the second experiment, we estimate the instrument amplitude from the mixture phase. This goes one step further than the previous experiment by involving separation in the comparative analysis of the pre-processing methods. The trained networks are then integrated in the MSS framework illustrated in Fig. 5. Estimations are scored following SiSEC 2016 policy (Liutkus et al., 2017). Fig. 7 shows the *signal-to-distortion ratio* (SDR) values (Vincent et al., 2007) on the DSD100 dataset where the values are obtained by first averaging the SDR values for each song and then computing the median over all 50 songs of the train set or test set, respectively. Again, we can observe that phase derivatives are

a much better feature representation and that shift correction systematically improves learning power of the system, leading occasionally to overfitting. The best test SDR is achieved by the frequency-derivative representation of the phase which generalizes better than the time-derivative representation.

Note that a network fed with phase features can only estimate the amplitude values up to a scale, meaning that it uses the average amplitude value per frequency bin of the training set as a starting point and estimates the variations from it based on the phase input. The post-processing stage uses a multi-channel Wiener filter (Sivasankaran et al., 2015; Nugraha et al., 2016; Uhlich et al., 2017) to recover the correct scale afterwards.

### 3. Combining Amplitude and Phase Features

In the previous section, we have seen that phase features can be used to estimate the instrument STFT amplitude. Therefore, we now turn to the problem of combining phase and amplitude features.

#### 3.1. Proposed Architecture

The most straight-forward way of combining amplitude and phase is a concatenation of both features at the input of the DNN. However, training such an approach results in networks that only rely on amplitude features as they set all weights in the input layer corresponding to the phase close to zero.<sup>2</sup> We could observe this if we use the raw phase as well as if we use the phase pre-processing described in Sec. 2.4.

Hence, we have to take special care to exploit the informa-

<sup>2</sup>In our opinion, this behaviour is due to the fact that the information in the STFT mixture amplitude is more easily accessible than the information in the STFT phase.

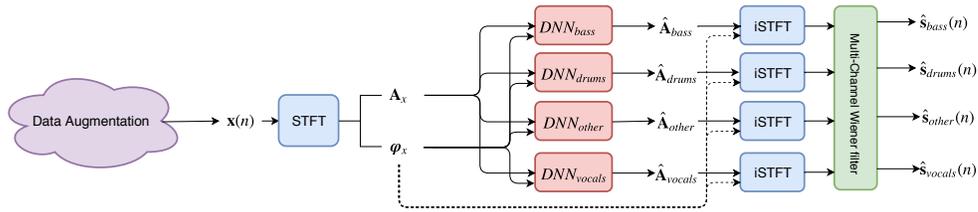


Figure 5. Music source separation general framework. In Sec. 2.5, the DNNs are only fed with the phase features in order to assess suggested pre-processing methods. Thus, with respect to the figure, the links between  $A_x$  and DNNs are removed. In Sec. 3, both amplitude and phase features are input to the DNNs as depicted in the figure.

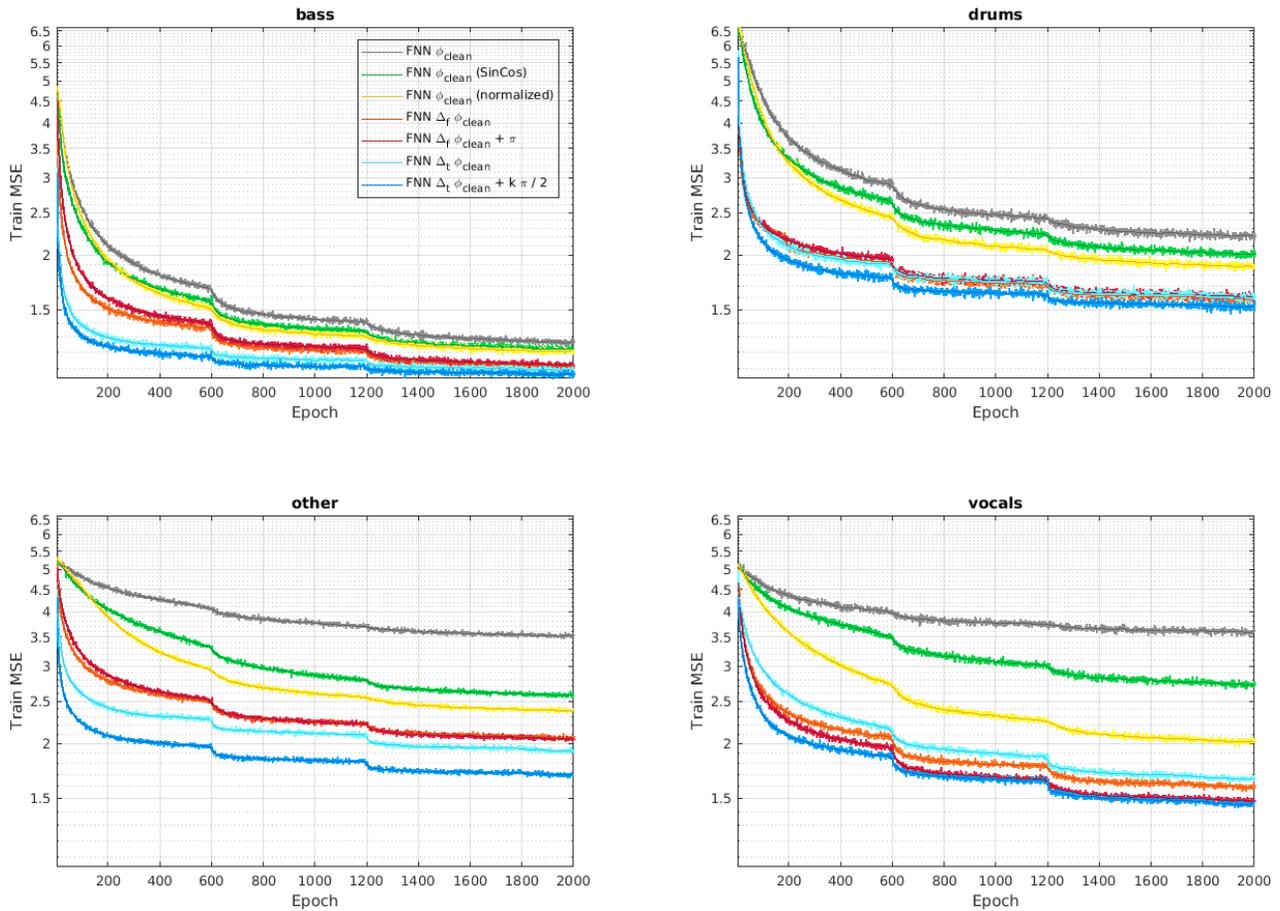


Figure 6. Training MSE for reconstructing amplitude from phase. Suggested pre-processing methods drastically improve the learning power of the network in comparison with raw phase. In particular, instantaneous frequencies and group delays are good representations. Moreover, shift correction systematically improves performances for all instrument.

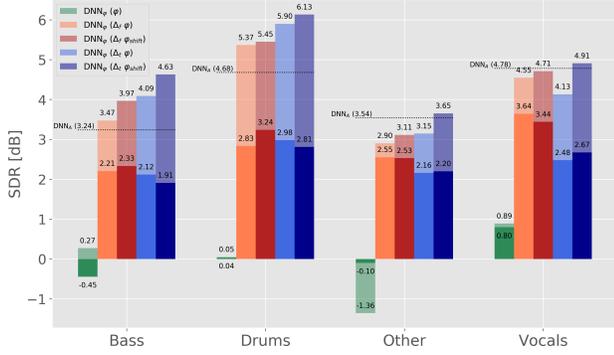


Figure 7. SDR results for phase only networks on DSD100. Comparison between various pre-processing methods. Note the poor results of raw phase without pre-processing on green. Each bar represents the accuracy obtained with a certain pre-processing method for a certain instrument. Height of opaque bar states the score on test set, while height of transparent bar (typically higher) is the score on training set. Horizontal dotted lines over each instrument give the score obtained by a DNN<sub>A</sub> on the test set.

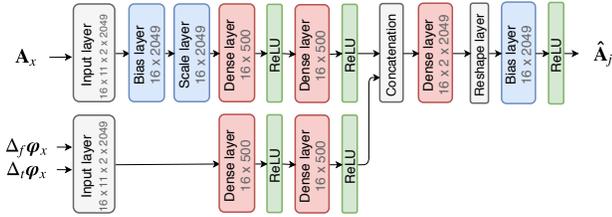


Figure 8. Proposed architecture for combining amplitude and phase.

tion of the phase features and we use the DNN architecture that is shown in Fig. 8. Instead of concatenating the features directly, we first process both through two dense layers before concatenating them.

The upper part of the network in Fig. 8 deals with the amplitude features. The features are first normalized by a bias layer and a scale layer, initialized with the mean and standard deviation per frequency bin over the training set. Two fully connected layers of 500 hidden units perform the feature extraction. The lower part of the network in Fig. 8 deals with the phase features. It takes as input both time and frequency derivatives, properly pre-processed as described in Sec. 2.4 and stacked together into an extra dimension. As for amplitude, two fully connected layers of 500 hidden units perform the feature extraction. The concatenation layer stacks the output of both previous networks and produces the amplitude estimates, which are de-normalized with the help of a final bias layer and a ReLU non-linearity, as described in Sec. 2.5. The training process is similar to the one described in Sec. 2.5. Time context is, as well, kept to five preceding/succeeding frames.

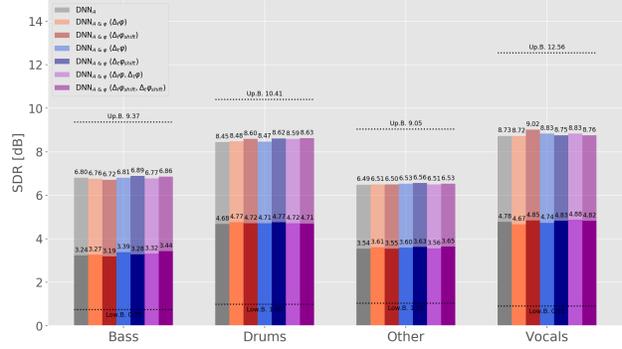


Figure 9. SDR results for networks combining phase and amplitude on DSD100. Comparison between various pre-processing methods. Each bar represents the accuracy obtained with a certain pre-processing method for a certain instrument. Height of opaque bar (typically higher) is the score on training set, while height of transparent bar (typically higher) is the score on test set. First score of each instrument, in gray, is given for illustrative purpose only and show the score obtained by DNN<sub>A</sub>.

Table 2. Results for concatenation architecture on DSD100 test set (SDR in dB).

Instrument	DNN <sub>A</sub>	DNN <sub>A</sub> & φ	Relative improv.
Bass	3.24	3.44	+6.17%
Drums	4.68	4.71	+0.64%
Other	3.54	3.65	+3.11%
Vocals	4.78	4.82	+0.84%

## 3.2. Results

Fig. 9 shows the results obtained with amplitude and phase combination. For comparison, we also trained a network DNN<sub>A</sub>, which uses only amplitude as feature and has the same structure as shown in Fig. 8 with the phase branch removed. Therefore, the amplitude information undergoes the same transformations and we can directly compare the two networks.

We use different combinations of pre-processing methods described in 2.4 in order to experience the individual relevance of each step. As expected, applying all proposed phase pre-processing methods together is beneficial for the MSS performance.

Finally, Table 2 shows the SDR obtained on the DSD100 test set. Comparing the baseline system DNN<sub>A</sub> with DNN<sub>A</sub> & φ (Δ<sub>t</sub>φ<sub>shift</sub>, Δ<sub>t</sub>φ<sub>shift</sub>), we observe that we can improve the SDR for all instruments and that especially the bass instrument improves by 0.2 dB.

## 4. Conclusion

In this paper, we proposed to consider the phase as an additional input feature to enhance the amplitude estimation. We

studied the relationship between the phase and the amplitude of an STFT and deduced a meaningful pre-processing, which was experimentally confirmed as relevant. We also found that special care must be taken in order to combine phase and amplitude features and, consequently, designed an adequate network architecture. The developed system improved SDRs on DSD100 for all instruments compared to an amplitude-only network with a similar network structure which showed the effectiveness of our system. Perceptually, this results in instruments more clearly separated from each other.

## References

- Auger, F., Chassande-Mottin, É., and Flandrin, P. On phase-magnitude relationships in the short-time Fourier transform. *IEEE Signal Processing Letters*, 19(5):267–270, 2012.
- Dubey, Mohit, Kenyon, Garrett, Carlson, Nils, and Thresher, Austin. Does phase matter for monaural source separation? *arXiv preprint arXiv:1711.00913*, 2017.
- Gerkmann, Timo, Krawczyk-Becker, Martin, and Le Roux, Jonathan. Phase processing for single-channel speech enhancement: History and recent advances. *IEEE Signal Processing Magazine*, 32(2):55–66, 2015.
- Huang, Po-Sen, Kim, Minje, Hasegawa-Johnson, Mark, and Smaragdis, Paris. Deep learning for monaural speech separation. *Proc. ICASSP*, pp. 1562–1566, 2014a.
- Huang, Po-Sen, Kim, Minje, Hasegawa-Johnson, Mark, and Smaragdis, Paris. Singing-voice separation from monaural recordings using deep recurrent neural networks. *International Society for Music Information Retrieval Conference (ISMIR)*, 2014b.
- Lee, Yuan-Shan, Wang, Chien-Yao, Wang, Shu-Fan, Wang, Jia-Ching, and Wu, Chung-Hsien. Fully complex deep neural network for phase-incorporating monaural source separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pp. 281–285. IEEE, 2017.
- Liutkus, A., Stöter, F.-R., Rafii, Z., Kitamura, D., Rivet, B., Ito, N., Ono, N., and Fontecave, J. The 2016 signal separation evaluation campaign. In *Proc. LVA/ICA*, pp. 323–332. Springer, 2017.
- Nugraha, A. A., Liutkus, A., and Vincent, E. Multichannel music separation with deep neural networks. In *Proc. EUSIPCO*, pp. 1748–1752, 2016.
- Ono, Nobutaka, Rafii, Zafar, Kitamura, Daichi, Ito, Nobutaka, and Liutkus, Antoine. The 2015 signal separation evaluation campaign. In *International Conference on Latent Variable Analysis and Signal Separation*, pp. 387–395. Springer, 2015.
- Sivasankaran, Sunit, Nugraha, Aditya Arie, Vincent, Emmanuel, Morales-Cordovilla, Juan A, Dalmia, Siddharth, Illina, Irina, and Liutkus, Antoine. Robust ASR using neural network based speech enhancement and feature simulation. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 482–489. IEEE, 2015.
- Smith, Julius O. *Mathematics of the Discrete Fourier Transform (DFT)*. W3K Publishing, 2007. ISBN 978-0-9745607-4-8.
- Stöter, F.-R., Liutkus, A., and Ito, N. The 2018 signal separation evaluation campaign. *arXiv preprint arXiv:1804.06267*, 2018.
- Takahashi, N. and Mitsufuji, Y. Multi-scale multi-band DenseNets for audio source separation. In *Proc. WASPAA*, pp. 21–25. IEEE, 2017.
- Takahashi, N., Agrawal, P., Goswami, N., and Mitsufuji, Y. Phasenet: Discretized phase modeling with deep neural networks for audio source separation. In *Proc. Interspeech*, 2018a.
- Takahashi, N., Goswami, N., and Mitsufuji, Y. MMDenseLSTM: An efficient combination of convolutional and recurrent neural networks for audio source separation. *accepted for IWAENC*, 2018b.
- Uhlich, S., Giron, F., and Mitsufuji, Y. Deep neural network based instrument extraction from music. In *Proc. ICASSP*, pp. 2135–2139. IEEE, 2015.
- Uhlich, S., Porcu, M., Giron, F., Enenkl, M., Kemp, T., Takahashi, N., and Mitsufuji, Y. Improving music source separation based on deep neural networks through data augmentation and network blending. In *Proc. ICASSP*, pp. 261–265. IEEE, 2017.
- Vincent, Emmanuel, Sawada, Hiroshi, Bofill, Pau, Makino, Shoji, and Rosca, Justinian P. First stereo audio source separation evaluation campaign: data, algorithms and results. In *International Conference on Independent Component Analysis and Signal Separation*, pp. 552–559. Springer, 2007.