Separating and denoising seismic signals with dual-path recurrent neural network architecture

Artemii Novoselov[1,1], Peter Balazs[2,2], and Götz Bokelmann[1,1]

[1]University of Vienna
[2]Austrian Academy of Sciences

November 30, 2022

Abstract

Seismologists have to deal with overlapping and noisy signals. Techniques such as source separation can be used to solve this problem. Over the past few decades, signal processing techniques used for source separation have advanced significantly for multi-station settings. But not so many options are available when it comes to single-station data. Using Machine Learning, we demonstrate the possibility of separating sources for single-station, one-component seismic recordings. The technique that we use for seismic signal separation is based on a dual-path recurrent neural network which is applied directly to the time domain data. Such source separation may find applications in most tasks of seismology, including earthquake analysis, aftershocks, nuclear verification, seismo-acoustics, and ambient-noise tomography. We train the network on seismic data from STanford EArthquake Dataset (STEAD) and demonstrate that our approach is a) capable of denoising seismic data and b) capable of separating two earthquake signals from one another. In this work, we show that Machine Learning is useful for earthquake-induced source separation. We provide a reproducible research repository with the algorithms here: https://github.com/IMGW-univie/source-separation.

SEDENOSS: SEparating and DENOising Seismic Signals with dual-path recurrent neural network architecture

Artemii Novoselov[1], Peter Balazs[2], Götz Bokelmann[1]

[1]University of Vienna, Department of Meteorology and Geophysics, Vienna *1090*, Austria
[2]Acoustics Research Institute, Vienna *1040*, Austria

Key Points:

- Seismic signals can be denoised via separating seismic signal from seismic noise
- Overlapping seismic signals recorded with a single sensor can be separated using techniques from machine learning.
- We provide a software package SEDENOSS for seismic signal separation and denoising

Corresponding author: Artemii Novoselov, `artemii.novoselov@univie.ac.at`

Abstract

Seismologists have to deal with overlapping and noisy signals. Techniques such as source separation can be used to solve this problem. Over the past few decades, signal processing techniques used for source separation have advanced significantly for multi-station settings. But not so many options are available when it comes to single-station data. Using Machine Learning, we demonstrate the possibility of separating sources for single-station, one-component seismic recordings. The technique that we use for seismic signal separation is based on a dual-path recurrent neural network which is applied directly to the time domain data. Such source separation may find applications in most tasks of seismology, including earthquake analysis, aftershocks, nuclear verification, seismo-acoustics, and ambient-noise tomography. We train the network on seismic data from STanford EArthquake Dataset (STEAD) and demonstrate that our approach is a) capable of denoising seismic data and b) capable of separating two earthquake signals from one another. In this work, we show that Machine Learning is useful for earthquake-induced source separation. We provide a reproducible research repository with the algorithms here: `https://github.com/IMGW-univie/source-separation`.

Plain Language Summary

Earthquake scientists have to deal with overlapping and noisy signals. They use signal processing techniques to solve this problem. Over the past few decades, these signal processing techniques have advanced greatly for multi-station settings. But not so many options are available when it comes to single-station data. Using Machine Learning, we demonstrate the possibility of separating sources for single-station, one-component seismic recordings. The technique that we use for seismic signal separation is based on a dual-path recurrent neural network which is applied directly to the time-domain data.

1 Introduction

Seismic recordings, such as those from earthquakes, often contain a significant amount of noise, which obscures the signals and complicates analysis and interpretation. The noisy seismic record is a mixture of both the seismic signal and the noise. When multiple signals compose a mixture, it is often advantageous to separate the mixture back into its individual signals. This is called *source separation.*

Several methods of source separation have been proposed. E.g. *independent-component analysis* (ICA) (Comon, 1994). Cabras et al. (2008) showed that ICA is a suitable technique to separate a volcanic source component from ocean microseisms background noise in a seismic dataset recorded at the Mt. Merapi volcano, Indonesia. Moni et al. (2012) used *degenerate unmixing estimation technique* for separation of long-period events from tremor, long-period events from volcano-tectonic events, and different sources of tremor from each other in the fields recordings obtained during the campaign on Mount Etna in 2008. It is also common to apply *beamforming methods* (Gibbons et al., 2008). E.g. Brooks et al. (2009) used beamforming to separate distinct dispersive waves in the ambient noise recordings. Boué et al. (2013) used *Double Beamforming Processing* to separate low-amplitude body waves from high-amplitude dispersive surface waves. Other methods of source separation, such as *independent-vector analysis* (Hiroe, 2006; Kim et al., 2006) and *MUSIC* (MUltiple SIgnal Classification) (Schmidt, 1986) (which later was extended to 3-component seismic data by Bear et al. (1999)), were also proposed in the field of signal processing.

In the multi-receiver setting, those methods work well. For instance, when more than one seismic station is available, source separation is widely employed. For single receivers (e.g. individual seismic stations with one component), however, there were not many choices available until recently. Separation was only possible if the frequency content of individual signals composing the mixture was different or if they didn't overlap in time.

A single-receiver source separation problem was explored in the Machine Learning domain (a branch of artificial intelligence and computer science that focuses on the use of data and algorithms, see e.g. Goodfellow et al. (2016) for more details). There are successful applications of Machine Learning based source separation to music (Stöter et al., 2019), hearing aids (Nossier et al., 2019), and speech enhancements (Luo et al., 2020).

Some of the Machine Learning source separation techniques (further referred to as *Neural Networks* or *models* interchangeably) operate in *frequency domain* (D. Wang & Brown, 2006; Vincent et al., 2006; Comon & Jutten, 2010a; Isik et al., 2016; Z.-Q. Wang et al., 2018), while others operate in *time-domain* (Luo & Mesgarani, 2018, 2019; L. Zhang et al., 2020; Luo et al., 2020). A Neural Network that can process time-domain (raw) data and output data in the same format is called an *end-to-end network*. At the time of writing, these methods are considered state-of-the-art.

In seismology, machine learning has not yet reached its full potential (Kong et al., 2019; Jiao & Alavi, 2020; Mousavi, Zhu, et al., 2019; X. Zhang et al., 2020; Mousavi & Beroza, 2019; McBrearty et al., 2019; DeVries et al., 2018). From an engineering point of view seismic (waveform) signals are essentially equal to speech signals, and thus methods developed in the speech separation domain can be used in seismology.

By applying source separation techniques to seismic signals, one can achieve advances in several seismological fields, including:

- **Earthquake analysis.** Seismic signals often have a low Signal-to-Noise ratio and are thus difficult to analyze (Mborah & Ge, 2018). One might use denoising (separation of a signal from the noise) to enhance the signal-to-noise ratio to analyze P- and S- phases of earthquakes (time of arrival of Primary and Secondary seismic waves). This capability of Machine Learning denoising was shown in van den Ende et al. (2021) for distributed acoustic sensing (DAS).
- **Aftershock analysis.** Large earthquakes are often accompanied by many aftershocks (Ross et al., 2018), and their number usually decays exponentially (Baranov et al., 2019). Early aftershocks are especially difficult to detect due to significant overlap (Peng & Zhao, 2009). To investigate aftershock properties, source separation (aftershock from the main quake, or one aftershock from the other) might be useful.
- **Acoustic-to-Seismic ground coupling.** Acoustic energy of various origins (e.g. explosions, meteorites, etc), is often coupled into the ground (Novoselov et al., 2020; Edwards, 2010; Schneider et al., 2018). This problem arises especially in nuclear verification (Hoffmann et al., 1999), where seismic data is used to estimate the location and the yielding mass of the potential nuclear explosion. Using a source separation technique, one can potentially separate both seismic and acoustic waves for analysis.
- **Ambient noise tomography**. Ambient noise tomography provides images of the subsurface using ambient noise sources (Shapiro & Campillo, 2004; Shapiro et al., 2005; Schippkus et al., 2018). Since deterministic signals often perturb noise mea-

surements, the latter must be removed. Source separation may preserve the noise portion of the data and therefore improve such imaging.

- **Exploration seismology**. Source separation can also benefit industrial applications, where one is often interested in producing an image of the subsurface from reflected seismic waves, to localize fossil fuels and other resources (Behura & Snieder, 2013). An explosive source is often used to obtain such images, and it may be important to remove the direct signal from the explosion (or an air-gun pulse for a marine setting) from records when dealing with such data. Those capabilities might be empowered on a whole new level by source separation.

In this work we adopt the approach by (Luo et al., 2020) using Dual-Path Recurrent Neural Networks (DPRNN) for source separation and demonstrate how this Machine Learning method can be applied to a) denoise seismic waveforms recorded with a single component individual seismic stations and b) separate two seismic signals, when they overlap in both time and frequency content. We then discuss potential issues and limitations of the proposed approach and draw some conclusions.

2 Data and Methods

2.1 Data

In this study, we utilize seismic data derived from STanford EArthquake Dataset (STEAD) (Mousavi, Sheng, et al., 2019) - a comprehensive dataset of pre-processed earthquakes with standardized metadata. We remove instrument response using stations metadata, normalize the three components on the global (for each individual record) maximum, extract vertical channels to obtain a single-channel record, and resample it to 30 samples per second (to reduce computational costs).

2.2 The network architecture

For the task of separation of seismic sources, we have chosen to adopt an approach by (Luo et al., 2020) (which initially was proposed for speech separation) using Dual-Path Recurrent Neural Networks (DPRNN). The *architecture* of DPRNN (in machine learning, the architecture refers to all of the layers and the major steps taken during the transformation of raw data for enabling the decision making of a system, in our case to output waveforms of separated sources) consists of four major parts (see Fig. 1a):

- **Encoder** - which is responsible for converting a sequential input (raw waveform) into an N-dimensional (where N - number of channels) representation (see Fig. 1c);
- **Separator** - which is responsible for the splitting of mixed signals into individual tracks (see Fig. 1b);
- **Mask Estimation module** - which is responsible for the creation of (S, N)-dimensional mask (where S - number of sources, set to 2 sources in the current paper), which is then applied to the original output of the Encoder (see Fig. 1d) and;
- **Decoder** - which is responsible for converting masked N-dimensional representation back into sequential output (waveform) (see Fig. 1e).

In Appendix A, we explain most of the building blocks required for such a Neural Network in detail.

2.3 Training procedure

To *train* a model, one needs to learn (determine) good values for all the parameters of the Neural Network that define how the input is transformed in the layers of such a network. A machine-learning algorithm builds a model based on many examples and attempts to find a variant of this model that minimizes *loss* with the help of examples. Loss is the penalty for a bad prediction. That is, the loss is a number indicating how bad the model's prediction was on a single example. The goal of training a model is to find a set of parameters that have low loss, on average, across all examples.

The training process involves drawing two samples (see Fig. 2a) from the dataset and summing them together to obtain a mixture (see Fig. 2b). This mixture is then processed through the Neural Network (see Fig. 2c-d), which in turn outputs separated signals (see Fig. 2e). These signals are then compared with the input signals and their correspondence (loss) is calculated. This process is repeated until the model can separate signals with acceptable quality. Each training iteration is defined as an *epoch* - a term used in machine learning, which indicates the number of passes of the entire training dataset the machine learning algorithm has completed. For each sample pair in an epoch, we randomly draw samples from the dataset (in a way that each sample is used only once as a *source 1* and only once as a *source 2*, and hence sample pairs are not repeated).

To improve ability of our model to learn from the given data and apply it to other situations (*generalization*), the following *augmentations* (techniques that increase data by adding slightly modified versions of existing data or new synthetic data made from existing data) were applied to each signal composing the mixture: random *polarity change*, randomly selected *high-pass frequency filter* (in the bounds of 0.5 - 1.5 Hz), randomly selected *low-pass frequency filter* (in the bounds of 10-14 Hz), random *amplitude gain* and *peak normalization* (adjusts the recording based on the highest signal level present in the recording). Augmentations are applied randomly each time a sample is drawn from the dataset.

The training objective (*loss function)* was to minimize the Scale-Invariant Source to Distortion Ratio $\ell_{\text{SI-SDR}}$ (Le Roux et al., 2019) between original individual sources and waveforms predicted by the model. This metric is widely used as a source separation performance indicator in the speech recognition domain (Fan et al., 2018, 2020; Gu et al., 2020).

$$\begin{gathered}\ell_{\text{SI-SDR}} = 10log_{10}\left(\frac{||e_{\text{target}}||^2}{||e_{\text{res}}||^2+\epsilon}\right)\\ e_{\text{target}} = \frac{\hat{s}^T s}{||s||^2}s\\ e_{\text{res}} = \frac{\hat{s}^T s}{||s||^2}s - \hat{s}\end{gathered} \tag{1}$$

where $||e_{\text{target}}||$ is scaled reference signal energy (double vertical bars enclosing an object is the norm of the object), $||e_{\text{res}}||$ is scaled residual energy, s - target signal, $\hat{s}$ - signal produced by the Neural Network, ϵ - a small stabilization value (10^{-8}) added to avoid a division by zero.

One of the limitations of DPRNN is that it doesn't guarantee a proper scaling of the processed signal. SI-SDR is invariant to the scale of the processed signal, which is desirable in this particular application.

Training the network to output several individual sources poses a problem: to calculate the loss function $\ell_{\text{SI-SDR}}$ one needs to know which estimated output corresponds to which *target source* (reference signal). To tackle this problem we use *Utterance level Permutation Invariant Training* (μPIT) (Kolbæk et al., 2017). The idea behind μPIT is rather simple (see Fig. 3): the loss function is computed between each pair of target source and estimated source, the lowest score between corresponding pairs is selected as the final loss.

Training a neural network can be accomplished using an *optimizer*. Optimizers change the attributes of a neural network, such as its weights, to minimize the loss function. In this study we use Ranger (Wright, 2020) - a synergistic optimizer combining RAdam (Rectified Adam) (Kingma & Ba, 2014) and Lookahead (M. Zhang et al., 2019) to speed up the learning process. We select the following *hyperparameters* (number of settings that affect the configuration of the model): encoder dimension=128, feature dimension=128, hidden dimension=64, layer=1, segment size=200, number of speakers = 2, kernel size = 2. The initial learning rate of 1e-3 was decaying by a factor of 0.9 every epoch. We selected those parameters using an empirical hyperparameter optimization approach.

3 Results

We train the DPRNN on seismic data from STanford EArthquake Dataset (STEAD) to demonstrate that our approach is a) capable of denoising seismic data and b) capable of separating two earthquakes signals from one another.

3.1 Denoising of the earthquake data

Most seismic records of earthquakes have low signal-to-noise ratios, i.e. the signal is contaminated with various types of noise. This complicates the analysis of such records. To reduce noise in seismic records, denoising may be applied. Essentially, denoising is a source separation, in the sense that noise is separated from a signal. We train a Neural Network (further referred to as *a model*) to perform a separation of signals (401795 one-minute-long earthquake records from the STEAD dataset with Signal-to-Noise ratio higher than 20 dB) from noise (108578 one-minute-long seismic noise records from the STEAD dataset). We then evaluate the performance of a trained model to denoise seismic data on a set of data previously unseen by the model (*model testing*). For this, we use additional 1000 earthquake records and 1000 noise records from the STEAD dataset.

Results of denoising are presented in Fig. 4 (input with a low Signal-to-Noise ratio), Fig. 5 (input with a medium Signal-to-Noise ratio) and Fig. 6 (input with a rather high Signal-to-Noise ratio). Signal-to-Noise ratio is defined as the standard deviation of signal divided by the standard deviation of noise trace ($SNR = \frac{\sigma_{before\ P}}{\sigma_{afterP}}$, where $\sigma_{before\ P}$

is the standard deviation before P arrival and σ_{afterP} is the standard deviation after P-arrival). Denoising helps to obtain much cleaner seismic records with more pronounced seismic phases. By using our model, we improve the SNR of the noisy signals significantly beyond what can be achieved with a simple highpass frequency filter (see Fig. 7). These results are better than in Zhu et al. (2019). In A06, we provide a comparison with their DeepDenoiser approach.

3.2 Source separation of earthquake data

After that, we try to accomplish something more difficult. Can two earthquake signals recorded by the same sensor at the same time be separated? If noise can be separated from the signal, then perhaps any other type of signal can be separated too. This might be particularly desirable in the aftershock analysis since the detection of overlapping aftershocks with the main quake or with each other is often limited.

We train a model (following the same procedure) to perform a separation of earthquake signals (595165 one-minute-long seismic records + 108578 one-minute-long seismic noise records from the STEAD dataset) from each other (e.g. earthquake 1 and earthquake 2). This is accomplished by composing training pairs randomly from either [signal + signal pairs] or [noise + signal pairs], or [noise + noise pairs]. We test the performance of our model on additional 1000 records of seismic signal mixtures (note, that noise is used only in the training step for augmentation purposes. We test the capability of the model to separate actual earthquake signals).

Fig. 8 - Fig. 10 demonstrate the results of applying our DPRNN implementation to the separation of two earthquake signals. While it is obvious that predicted signals contain under-suppressed signals from each other (as shown on residual plots), they do correspond quite well to their target counterparts. Although separated sources might not be optimal for complex frequency analysis, they certainly can be used to improve phase picking of individual signals (either manually by a trained expert or automatically by using an algorithm like Mousavi et al. (2020)). This way we demonstrate how our model can be used in the earthquake analysis. We might also find our source separation neural network useful in an unusual scientific case - an atmospheric entry of the Mars2020 lander during a marsquake (Fernando et al., 2021). Additional research is being conducted to prove this point.

4 Discussion

4.1 Why does DPRNN work?

Seismologists are used to separating signals (and noise) if they differ in frequency content or time of arrival. If an array is available, signals may also be separated by their different apparent velocity and/or azimuth. The approach presented in this paper does not require that such separating features exist. It may thus seem counterintuitive that we are nevertheless able to extract multiple signals from single-station data. This capability results from knowledge learned by analyzing many realizations of seismic signals and noise and extracting characteristics of seismic signals. Layers of the Neural Network are transforming data and extracting features (note, that those features are not as easy to interpret as frequency spectrum, but the concept is similar). In the case of DPRNN, separation happens in N-dimensional vector-space (where N - is the number of features and channels, learned by the Network). Each row in the Encoder (see Fig. 2B) is a feature vector. The Neural Network learns to pay attention to the statistical distribution of the above-mentioned features during training. For example, if we train the network to separate two signals, it should learn the distribution of features in each signal is and how a mixture of such signals looks. It then attempts to find the most likely option, where features of a signal 1 have a distribution of features corresponding to a real signal, features of a signal 2 has also a distribution of the features of a real signal, and features of their mixture have a distribution of the features of a mixture of two real signals.

4.2 Time representation vs Time-frequency representation

One might ask why we choose an end-to-end approach instead of one based on STFT (Short-Time Fourier Transform) features? First of all, we adopted a state-of-the-art technique (at the time of writing) that is based on end-to-end processing. Second, even though STFT has some benefits like reduction of the computational complexity of the signal, Machine Learning approaches based on STFT have several limitations. By selecting several parameters of the STFT manually and thus forcing precomputed representation of the raw signal, one limits the ability of the network to learn patterns in the raw data itself. Also, the STFT outputs complex values. Neural networks are currently not ready to be working efficiently with complex numbers; although this is an area of current research (Dramsch et al., 2019). So far, one must take the absolute values of such an STFT transform, which in turn leads to the loss of phase information, and has to be compensated by phase retrieval approaches (Průša et al., 2017a). We acknowledge that many of those limitations could be overcome: one way would be to apply phase retrieval (Průša et al., 2017b), which has been done successfully in previous works (Marafioti, Perraudin, et al., 2019b, 2019a; Marafioti, Holighaus, et al., 2019).

A time-frequency representation is likely a useful representation for source separation (see the signal processing approach (Comon & Jutten, 2010b)). It can also be expected that training an algorithm with time-frequency representation could be faster (Schlüter, 2017). At the same time direct comparison between STFT and learned representation of the waveform shown in Heitkaemper et al. (2020), for this particular type of neural network suggests that at least a naïve introduction of STFT would not benefit the source separation.

4.3 SI-SDR loss

When it comes to the choice of the loss function, it is quite common to employ popular mean-square error (MSE or L2) loss when training neural networks. However, SI-SDR loss is more favorable, since minimizing the MSE may not guarantee the highest signal quality. It was demonstrated in Kolbæk et al. (2020), that source separation networks trained with loss function based on SI-SDR achieve superior performance.

It was also shown in Heitkaemper et al. (2020) that the SI-SDR loss function is directly related to the logarithmic MSE (minimum square error) loss function that is used in source separation based on time-frequency domain data and in fact can be re-written as:

$$\text{SI-SDR} = \text{LOG-L2} = 10 * \frac{1}{\mathrm{K}} \sum_k log_{10} \sum_t |y_{t,k} - \hat{y_{t,k}}|^2 \quad (2)$$

where K and $_k$ are the numbers of sources, $_t$ is the sample index.

SI-SDR is invariant to the scale of the processed signal, which is desirable in applications, where the signal processing algorithm does not guarantee a proper scaling of the processed signal, such as DPRNN. But at the same time, this is the greatest limitation of our approach. Information about the absolute amplitude is lost, when the signal is processed through the Neural Network, although relative (to each individual signal) amplitudes are preserved.

4.4 Modification of original DPRNN.

To achieve a reasonable separation quality we needed to make some changes to the original DPRNN architecture (see Luo and Mesgarani (2020) for details). First, we replaced all *activation functions* (such functions define how the weighted sum of the input is transformed into an output from a layer of the network) with Mish activation, as it reducing problems of small gradients inside the network (refer to A01 and Hochreiter et al. (2001) for more details). In addition, we replaced the last activation in the Mask Estimation module with a *Softmax* activation. Softmax operation (defined as $\mathrm{Softmax}(x_i) = \frac{\exp(x_i)}{\sum \exp(x_j)}$) is used to rescale all elements of the input so that the elements of the n-dimensional output tensor lie in the range [0,1] and sum to 1. As a result of Softmax being applied, values correspond to a "masking strength" (where values close to 0 indicate omitting the input in the encoded representation input completely, and 1 indicates to keep this part of the encoded input as it is). This way, sources are masked from the mixture.

The number of sources to separate was set to 2 in the current paper, however, the neural network is not limited to only 2 sources. As was shown in (Luo & Mesgarani, 2018), the number of sources could be 3 and higher. With more sources to separate, the quality of the prediction declines.

4.5 Ways to improve

It may be possible to enhance the network's capability to perform source separation. One can accomplish this by either increasing the complexity of the encoder and seeing whether this improves results, or by replacing the training objective with one requiring better task construction. One may also utilize the attention mechanism (Vaswani et al., 2017). Recently attention mechanisms gained a lot of recognition in Machine Learning research (Y. Wang et al., 2020) and in source separation particularly (Fan et al., 2020). We tried to utilize Simple Self Attention (Cheng et al., 2016) at different layers in the network but we didn't achieve any advances with this approach. Another set of potential solutions is to use unconstrained number of sources in the mixture, perhaps combined with the source counting (Luo & Mesgarani, 2020), additional meta-information-learning (Ephrat et al., 2018; Zeghidour & Grangier, 2020), source classification before separation (Ji et al., 2020; Kinoshita et al., 2020; Mun et al., 2020; Tjandra et al., 2020) and leverage of a Transformer architecture (Vaswani et al., 2017; Karita et al., 2019; Mousavi et al., 2020).

5 Conclusions

We have adopted an approach of signal separation called Dual Path Reccurent Neural Network (DPRNN) from Luo et al. (2020). We trained this Neural Network with seismic data from the STEAD dataset. We have focused on applying source separation first to denoise seismic data, and then to separate two earthquake signals. We demonstrate that our network is capable of denoising and separating these signals.

It is expected that Dual-Path Residual Neural Network can be widely applied in most tasks of seismology. E.g., it can be applied in aftershock analysis and seismoacoustics, where different waves need to be distinguished. Besides that, signal-noise separation is an important problem in the domain of earthquake analysis (e.g. for better defining earthquake phases (Mborah & Ge, 2018)), and ambient noise tomography. Potentially Machine Learning can demonstrate the effectiveness in e.g. an especially noisy environment; collection and characterization of anthropogenic noise data with low-cost seismometers; distinguishing between different types of vehicle noise, such as bus and train; and tracking changes in human activity over time with seismic sensors.

This work proves the concept and steers the direction for further research of earthquake-induced source separation. We provide a reproducible research repository with the algorithms, software (which we called SEDENOSS), and datasets. The successful application of seismic denoising and separation suggests that the source separation approach works not only with speech data but also with earthquake data and perhaps can even be used beyond that to any waveform data in general.

Acknowledgments

The model was trained on GPU provided by Google Colaboratory (Bisong, 2019) and using the Vienna Scientific Cluster (VSC). The authors thank Petr Kolínskỳ for discussions on earthquake wave propagation and properties, Nicki Holighaus for discussions on different data representations. Andrew Delorey helped to understand how a neural network might be able to perform separation and reviewed the text.

We gratefully acknowledge funding by the Austrian Science Fund FWF through project numbers P30707, Y551-N13, and I 3067–N30. Artemii Novoselov was funded via the Emerging Field Project "ThunderSeis" of the Faculty of Geosciences, Geography, and Astronomy of the University of Vienna.

Data processing and analysis was done using Python 3.6.9 (van Rossum, 1997), NumPy 1.18.5 (Harris et al., 2020), SciPy 1.4.1 (Virtanen & et al., 2020), ObsPy 1.2.0 (Beyreuther et al., 2010). PyTorch 1.5.1 (Paszke et al., 2019) and Sklearn 0.22.2 (Pedregosa et al., 2011) were used as frameworks for model building and training, based on the DPRNN implementation by Shi Ziqiang et al. (2020) (Ziqiang, n.d.). Figures were produced with Plotly 4.4.1 (Plotly, 2015), Matplotlib 3.2.2 (Hunter, 2007) and `https://draw.io`.

All codes (software SEDENOSS) to reproduce the results of this work, pre-processing of the dataset as well as pre-trained models, are available at `https://github.com/IMGW-univie/source-separation` and `https://doi.org/10.5281/zenodo.5464483` (Novoselov, 2021).

References

Baranov, S., Gvishiani, A., Narteau, C., & Shebalin, P. (2019). Epidemic type aftershock sequence exponential productivity. *Russian Journal of Earth Sciences*, *19*(6).

Bear, L. K., Pavlis, G. L., & Bokelmann, G. H. (1999). Multi-wavelet analysis of three-component seismic arrays: Application to measure effective anisotropy at pinon flats, california. *Bulletin of the Seismological Society of America*, *89*(3), 693–705.

Behura, J., & Snieder, R. (2013). Virtual real source: Source signature estimation using seismic interferometry. *Geophysics*, *78*(5), Q57–Q68.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157–166.

Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y., & Wassermann, J. (2010). Obspy: A python toolbox for seismology. *Seismological Research*

Letters, *81*(3), 530–533.

Bisong, E. (2019). Google colaboratory. In *Building machine learning and deep learning models on google cloud platform* (pp. 59–64). Springer.

Boué, P., Roux, P., Campillo, M., & de Cacqueray, B. (2013). Double beamforming processing in a seismic prospecting context. *Geophysics*, *78*(3), V101–V108.

Brooks, L. A., Townend, J., Gerstoft, P., Bannister, S., & Carter, L. (2009). Fundamental and higher-mode rayleigh wave characteristics of ambient seismic noise in new zealand. *Geophysical Research Letters*, *36*(23).

Cabras, G., Carniel, R., & Wassermann, J. (2008). Blind source separation: An application to the mt. merapi volcano, indonesia. *Fluctuation and Noise Letters*, *8*(03n04), L249–L260.

Cheng, J., Dong, L., & Lapata, M. (2016, nov). Long short-term memory-networks for machine reading. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 551–561). Austin, Texas: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D16-1053 doi: 10.18653/v1/D16-1053

Comon, P. (1994). Independent component analysis, a new concept? *Signal processing*, *36*(3), 287–314.

Comon, P., & Jutten, C. (2010a). *Handbook of blind source separation: Independent component analysis and applications.* ACADEMIC PR INC. Retrieved from http://www.ebook.de/de/product/9020313/pierre_comon_handbook_of_blind_source_separation_independent_component_analysis_and_applications.html

Comon, P., & Jutten, C. (2010b). *Handbook of blind source separation: Independent component analysis and applications.* Academic press.

Csáji, B. C., et al. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, *24*(48), 7.

DeVries, P. M., Viégas, F., Wattenberg, M., & Meade, B. J. (2018). Deep learning of aftershock patterns following large earthquakes. *Nature*, *560*(7720), 632–634.

Dramsch, J. S., Lüthje, M., & Christensen, A. N. (2019). *Complex-valued neural networks for machine learning on non-stationary physical data.* Preprint at https://arxiv.org/pdf/1905.12321.pdf.

Edwards, W. N. (2010). Meteor generated infrasound: Theory and observation. In

Infrasound monitoring for atmospheric studies (pp. 361–414). Springer.

Ephrat, A., Mosseri, I., Lang, O., Dekel, T., Wilson, K., Hassidim, A., ... Rubinstein, M. (2018, July). Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. *ACM Trans. Graph.*, *37*(4). Retrieved from `https://doi.org/10.1145/3197517.3201357`

Fan, C., Liu, B., Tao, J., Wen, Z., Yi, J., & Bai, Y. (2018). Utterance-level permutation invariant training with discriminative learning for single channel speech separation. In *2018 11th international symposium on chinese spoken language processing (iscslp)* (pp. 26–30).

Fan, C., Tao, J., Liu, B., Yi, J., Wen, Z., & Liu, X. (2020, January). End-to-end post-filter for speech separation with deep attention fusion features. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, *28*, 1303–1314. Retrieved from `https://doi.org/10.1109/TASLP.2020.2982029`

Fernando, B., Wójcicka, N., Froment, M., Maguire, R., Stähler, S. C., Rolland, L., ... others (2021). Listening for the landing: Seismic detections of perseverance's arrival at mars with insight. *Earth and Space Science*, *8*(4), e2020EA001585.

Gibbons, S. J., Ringdal, F., & Kværna, T. (2008). Detection and characterization of seismic phases using continuous spectral estimation on incoherent and partially coherent arrays. *Geophysical Journal International*, *172*(1), 405–421.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT press Cambridge.

Gu, R., Zhang, S., Chen, L., Xu, Y., Yu, M., Su, D., ... Yu, D. (2020). Enhancing end-to-end multi-channel speech separation via spatial feature learning. In *Icassp 2020 - 2020 ieee international conference on acoustics, speech and signal processing (icassp)* (p. 7319-7323).

Harris, C., Millman, K., van der Walt, S., & et al. (2020). Array programming with numpy. *Nature*, *585*, 357–362. Retrieved from `https://doi.org/10.1038/s41586-020-2649-2`

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Heitkaemper, J., Jakobeit, D., Boeddeker, C., Drude, L., & Haeb-Umbach, R.

(2020). Demystifying tasnet: A dissecting approach. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 6359–6363).

Hiroe, A. (2006). Solution of permutation problem in frequency domain ica, using multivariate probability density functions. In *International conference on independent component analysis and signal separation* (pp. 601–608). doi: https://doi.org/10.1007/11679363_75

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al. (2001). *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.* A field guide to dynamical recurrent neural networks. IEEE Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hoffmann, W., Kebeasy, R., & Firbas, P. (1999). Introduction to the verification regime of the comprehensive nuclear-test-ban treaty. *Physics of the Earth and Planetary Interiors*, *113*(1-4), 5–9.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in science & engineering*, *9*(3), 90.

Isik, Y., Roux, J. L., Chen, Z., Watanabe, S., & Hershey, J. R. (2016). *Single-channel multi-speaker separation using deep clustering.* Preprint at `https://arxiv.org/pdf/1607.02173.pdf`.

Ji, X., Yu, M., Zhang, C., Su, D., Yu, T., Liu, X., & Yu, D. (2020). Speaker-aware target speaker enhancement by jointly learning with speaker embedding extraction. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 7294–7298).

Jiao, P., & Alavi, A. H. (2020). Artificial intelligence in seismology: Advent, performance and future trends. *Geoscience Frontiers*, *11*(3), 739–744.

Karita, S., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., ... others (2019). A comparative study on transformer vs rnn in speech applications. In *2019 ieee automatic speech recognition and understanding workshop (asru)* (pp. 449–456).

Kim, T., Eltoft, T., & Lee, T.-W. (2006). Independent vector analysis: An extension of ica to multivariate components. In *International conference on independent component analysis and signal separation* (pp. 165–172).

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization.* Preprint at https://arxiv.org/pdf/1412.6980.pdf.

Kinoshita, K., Delcroix, M., Araki, S., & Nakatani, T. (2020). *Tackling real noisy reverberant meetings with all-neural source separation, counting, and diarization system.* Preprint at `https://arxiv.org/abs/2003.03987`.

Kolbæk, M., Tan, Z.-H., Jensen, S. H., & Jensen, J. (2020). On loss functions for supervised monaural time-domain speech enhancement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 825–838.

Kolbæk, M., Yu, D., Tan, Z.-H., & Jensen, J. (2017). Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(10), 1901–1913.

Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J., & Gerstoft, P. (2019). Machine learning in seismology: Turning data into insights. *Seismological Research Letters*, *90*(1), 3–14.

Le Roux, J., Wisdom, S., Erdogan, H., & Hershey, J. R. (2019). Sdr–half-baked or well done? In *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 626–630).

Luo, Y., Chen, Z., & Yoshioka, T. (2020). Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 46–50).

Luo, Y., & Mesgarani, N. (2018). Tasnet: time-domain audio separation network for real-time, single-channel speech separation. In *2018 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 696–700).

Luo, Y., & Mesgarani, N. (2019). Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, *27*(8), 1256–1266.

Luo, Y., & Mesgarani, N. (2020). *Separating varying numbers of sources with auxiliary autoencoding loss.* Preprint at `https://arxiv.org/abs/2003.12326`.

Marafioti, A., Holighaus, N., Majdak, P., Perraudin, N., et al. (2019). Audio inpainting of music by means of neural networks. In *Audio engineering society convention 146.*

Marafioti, A., Perraudin, N., Holighaus, N., & Majdak, P. (2019a, 09–15 Jun). Adversarial generation of time-frequency features with application in audio synthesis. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (Vol. 97, pp. 4352–4362). Long Beach, California, USA: PMLR. Retrieved from `http://proceedings.mlr.press/v97/marafioti19a.html`

Marafioti, A., Perraudin, N., Holighaus, N., & Majdak, P. (2019b). A context encoder for audio inpainting. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *27*(12), 2362–2372.

Mborah, C., & Ge, M. (2018). Enhancing manual p-phase arrival detection and automatic onset time picking in a noisy microseismic data in underground mines. *International Journal of Mining Science and Technology*, *28*(4), 691–699.

McBrearty, I. W., Gomberg, J., Delorey, A. A., & Johnson, P. A. (2019). Earthquake arrival association with backprojection and graph theory. *Bulletin of the Seismological Society of America*, *109*(6), 2510–2531.

Misra, D. (2019). *Mish: A self regularized non-monotonic neural activation function.* Preprint at `https://arxiv.org/abs/1908.08681`.

Moni, A., Bean, C. J., Lokmer, I., & Rickard, S. (2012). Source separation on seismic data: Application in a geophysical setting. *IEEE Signal Processing Magazine*, *29*(3), 16–28.

Mousavi, S. M., & Beroza, G. C. (2019). A machine-learning approach for earthquake magnitude estimation. *Geophysical Research Letters*.

Mousavi, S. M., Ellsworth, W. L., Zhu, W., Chuang, L. Y., & Beroza, G. C. (2020). Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications*, *11*(1), 1–12.

Mousavi, S. M., Sheng, Y., Zhu, W., & Beroza, G. C. (2019). Stanford earthquake dataset (stead): A global data set of seismic signals for ai. *IEEE Access*.

Mousavi, S. M., Zhu, W., Sheng, Y., & Beroza, G. C. (2019). Cred: A deep residual network of convolutional and recurrent units for earthquake signal detection. *Scientific reports*, *9*(1), 1–14.

Mun, S., Choe, S., Huh, J., & Chung, J. S. (2020). The sound of my voice: speaker representation loss for target voice separation. In *Icassp 2020-2020 ieee in-*

ternational conference on acoustics, speech and signal processing (icassp) (pp. 7289–7293).

Nossier, S. A., Rizk, M., Moussa, N. D., & el Shehaby, S. (2019). Enhanced smart hearing aid using deep neural networks. *Alexandria Engineering Journal*, *58*(2), 539–550.

Novoselov, A. (2021, November). *Imgw-univie/source-separation: v0.1.1-beta.* Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.5464483` doi: 10.5281/zenodo.5464483

Novoselov, A., Fuchs, F., & Bokelmann, G. (2020). Acoustic-to-seismic ground coupling: coupling efficiency and inferring near-surface properties. *Geophysical Journal International*, *223*(1), 144–160.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... others (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems* (pp. 8024–8035).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, Z., & Zhao, P. (2009). Migration of early aftershocks following the 2004 parkfield earthquake. *Nature Geoscience*, *2*(12), 877–881.

Plotly, T. I. (2015). *Collaborative data science.* `https://plot.ly`. Montreal, QC: Plotly Technologies Inc.

Průša, Z., Balazs, P., & Søndergaard, P. L. (2017a). A non-iterative method for (re)construction of phase from stft magnitude. *IEEE Transactions on Audio, Speech and Language Processing*, *25*(5), 1154 - 1164.

Průša, Z., Balazs, P., & Søndergaard, P. L. (2017b). A noniterative method for reconstruction of phase from stft magnitude. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(5), 1154–1164.

Ross, Z. E., Meier, M.-A., Hauksson, E., & Heaton, T. H. (2018). Generalized seismic phase detection with deep learning. *Bulletin of the Seismological Society of America*, *108*(5A), 2894–2901.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Schippkus, S., Zigone, D., Bokelmann, G., Group, A. W., et al. (2018). Ambient-noise tomography of the wider vienna basin region. *Geophysical Journal International*, *215*(1), 102–117.

Schlüter, J. (2017). Deep learning for event detection, sequence labelling and similarity estimation in music signals. *Ph. D. thesis*.

Schmidt, R. (1986). Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, *34*(3), 276–280.

Schneider, F. M., Fuchs, F., Kolínskỳ, P., Caffagni, E., Serafin, S., Dorninger, M., … others (2018). Seismo-acoustic signals of the baumgarten (austria) gas explosion detected by the alparray seismic network. *Earth and Planetary Science Letters*, *502*, 104–114.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, *45*(11), 2673–2681.

Shapiro, N. M., & Campillo, M. (2004). Emergence of broadband rayleigh waves from correlations of the ambient seismic noise. *Geophysical Research Letters*, *31*(7).

Shapiro, N. M., Campillo, M., Stehly, L., & Ritzwoller, M. H. (2005). High-resolution surface-wave tomography from ambient seismic noise. *Science*, *307*(5715), 1615–1618.

Stöter, F.-R., Uhlich, S., Liutkus, A., & Mitsufuji, Y. (2019). Open-unmix-a reference implementation for music source separation. *Journal of Open Source Software*, *4*(41). Retrieved from `https://doi.org/10.21105/joss.01667`

Tjandra, A., Liu, C., Zhang, F., Zhang, X., Wang, Y., Synnaeve, G., … Zweig, G. (2020). Deja-vu: Double feature presentation and iterated loss in deep transformer networks. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 6899–6903).

van den Ende, M., Lior, I., Ampuero, J.-P., Sladen, A., Ferrari, A., & Richard, C. (2021). A self-supervised deep learning approach for blind denoising and waveform coherence enhancement in distributed acoustic sensing data.

van Rossum, G. (1997). Scripting the web with python. *World Wide Web Journal*, *2*(2), 97–120.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vincent, E., Gribonval, R., & Plumbley, M. D. (2006). Oracle estimators for the benchmarking of source separation algorithms. *Signal Processing*.

Virtanen, P., & et al. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, *17*, 261–272. Retrieved from `https://doi.org/10.1038/s41592-019-0686-2`

Wang, D., & Brown, G. J. (2006). *Computational auditory scene analysis: Principles, algorithms, and applications.* Wiley-IEEE Press.

Wang, Y., Mohamed, A., Le, D., Liu, C., Xiao, A., Mahadeokar, J., … others (2020). Transformer-based acoustic modeling for hybrid speech recognition. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 6874–6878).

Wang, Z.-Q., Roux, J. L., Wang, D., & Hershey, J. R. (2018). *End-to-end speech separation with unfolded iterative phase reconstruction.* Preprint at `https://arxiv.org/pdf/1804.10204.pdf`.

Wright, L. (2020). *New deep learning optimizer, ranger: Synergistic combination of radam + lookahead for the best of both.* `https://medium.com/@lessw/new-deep-learning-optimizer-ranger-synergistic-combination-of-radam-lookahead-for-the-best-of-2dc83f79a48d`.

Wu, Y., & He, K. (2018). Group normalization. In *Proceedings of the european conference on computer vision (eccv)* (pp. 3–19).

Zeghidour, N., & Grangier, D. (2020). *Wavesplit: End-to-end speech separation by speaker clustering.* Preprint at `https://arxiv.org/abs/2002.08933`.

Zhang, L., Shi, Z., Han, J., Shi, A., & Ma, D. (2020). Furcanext: End-to-end monaural speech separation with dynamic gated dilated temporal convolutional networks. In *International conference on multimedia modeling* (pp. 653–665).

Zhang, M., Lucas, J., Ba, J., & Hinton, G. E. (2019). Lookahead optimizer: k steps forward, 1 step back. In *Advances in neural information processing systems* (pp. 9593–9604).

Zhang, X., Zhang, J., Yuan, C., Liu, S., Chen, Z., & Li, W. (2020). Locating induced earthquakes with a network of seismic stations in oklahoma via a deep learning method. *Scientific reports*, *10*(1), 1–12.

Zhou, D.-X. (2020). Universality of deep convolutional neural networks. *Applied and computational harmonic analysis*, *48*(2), 787–794.

Zhu, W., Mousavi, S. M., & Beroza, G. C. (2019). Seismic signal denoising and decomposition using deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(11), 9476–9488.

Ziqiang, S. (n.d.). *Dual-path rnns based speech separation. GitHub* https://github.com/ShiZiqiang/dual-path-RNNs-DPRNNs-based-speech-separation.

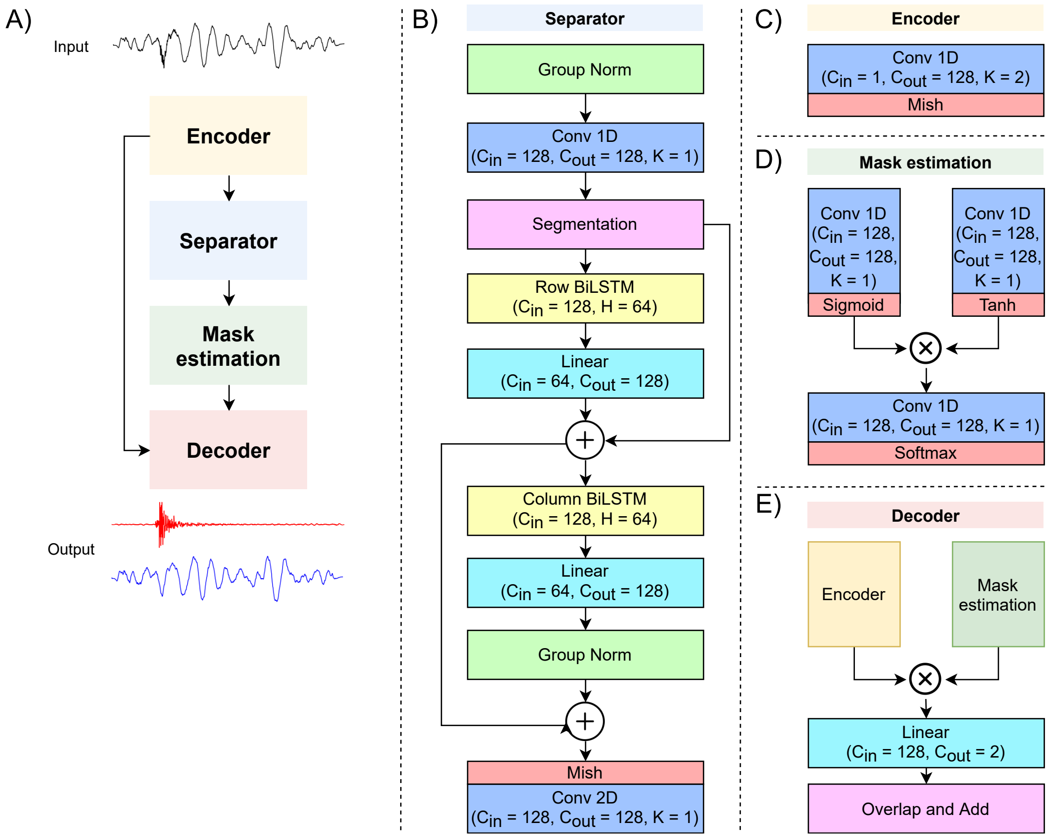

Figure 1. A) Architecture of Dual-Path Recurrent Neural Network (Luo et al., 2020) with modifications. B) Separator module. C) Encoder module. D) Mask estimation module. E) Decoder module.

Conv1D and Conv2D - 1D and 2D convolution operations, correspondingly; Mish, Tanh and Sigmoid - activation functions; Linear - Fully-Connected layer; GroupNorm - Group Normalization, Row and Column BiLSTM - row-wise and column-wise bidirectional Long-Short-Term-Memory Cells; Separation, Merging, Overlap and Add - array manipulations. Arrows indicate an order of operations applied to the input. + is element-wise summation; x is element-wise multiplication. C_{in} - input channels, C_{out} - output channels, K - kernel size. In Appendix A, we explain most of the building blocks required for such a Neural Network in details.

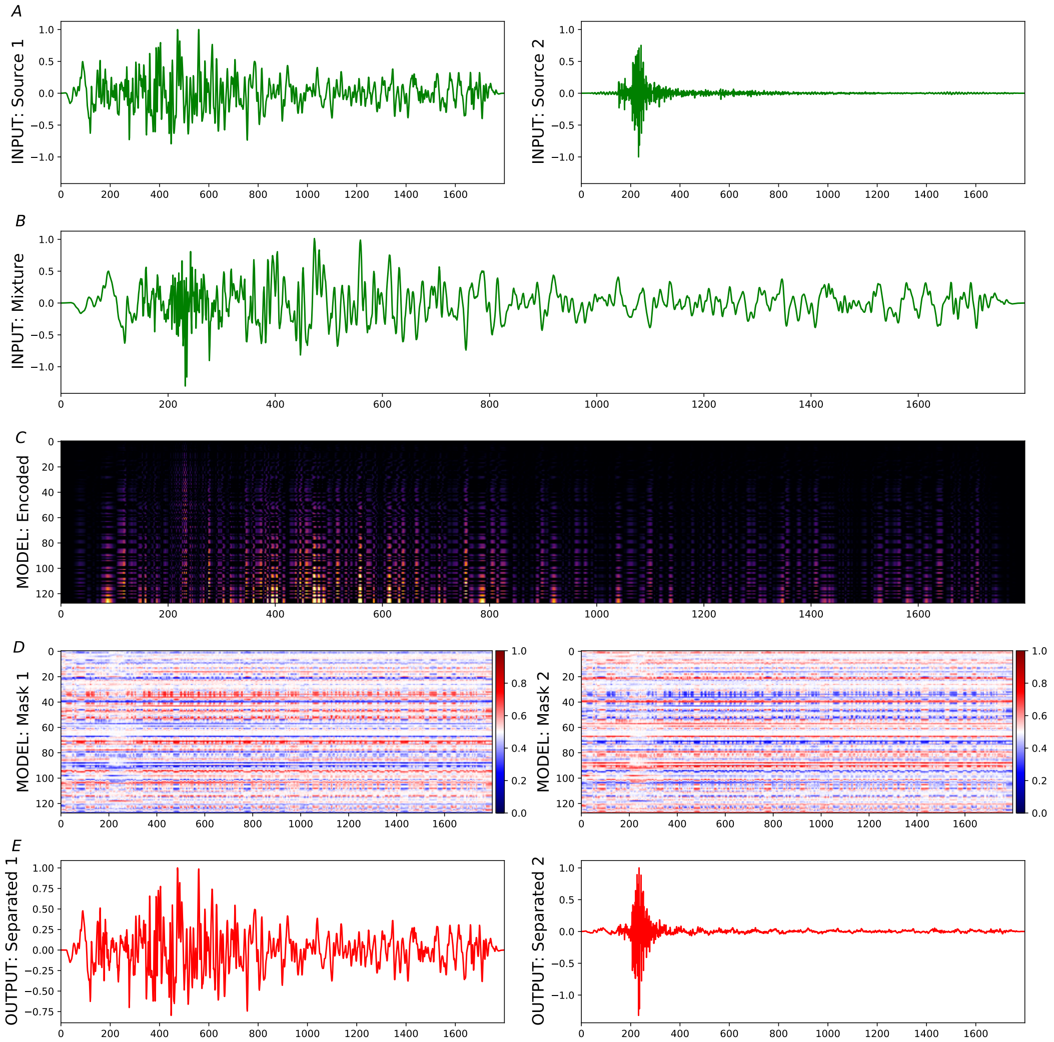

Figure 2. A) An input source 1 (e.g. earthquake 1 - S1) and an input source 2 (e.g. earthquake 2 - S2). B) An input mixture that consists of two sources (S1 + S2). C) An output of the Encoder module of the Neural Network. On the vertical axis, 128 channels (the result of Conv1d operation) are shown. Note that encoder color values are clipped for visibility. D) We further show Estimated Masks, obtained as the result of the processing through the Neural Network (Separation and Mask Estimation modules). We can observe that mask for source 2 is effectively opposite of the mask for source 1, which means multiplying the encoded representation by any of these masks would not lead to the introduction of extra information into the separated sources. E) Source 1 and Source 2 are separated by the Neural Network from the mixture (Encoded representation is multiplied with corresponding masks and then results of this operation are processed with the Decoder module).

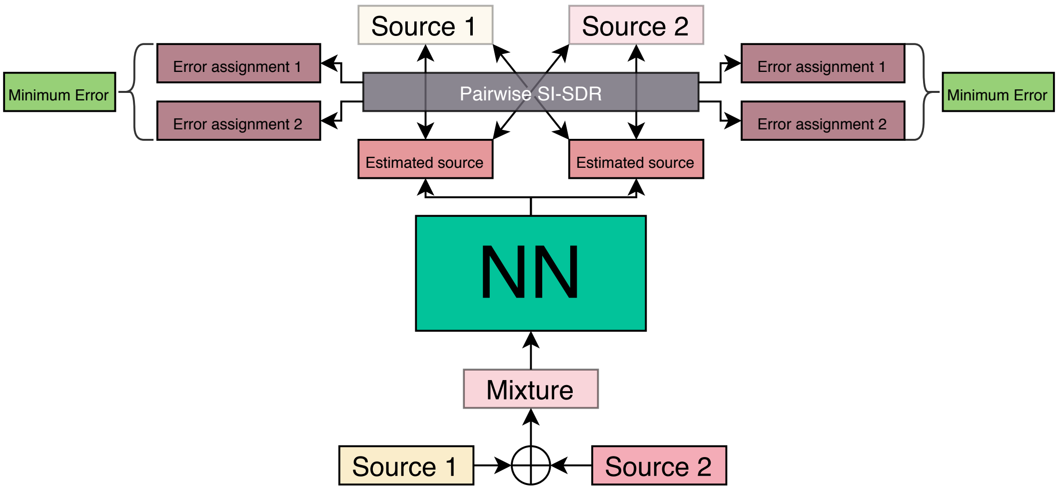

Figure 3. Permutation-invariant training. Target sources are summed together to obtain a mixture. This mixture is fed to the separation network and two estimated sources are obtained. Loss function SI-SDR is then computed for each pair correspondingly. Pairwise metrics are compared, and those with the smallest error are the output of such a training scheme.

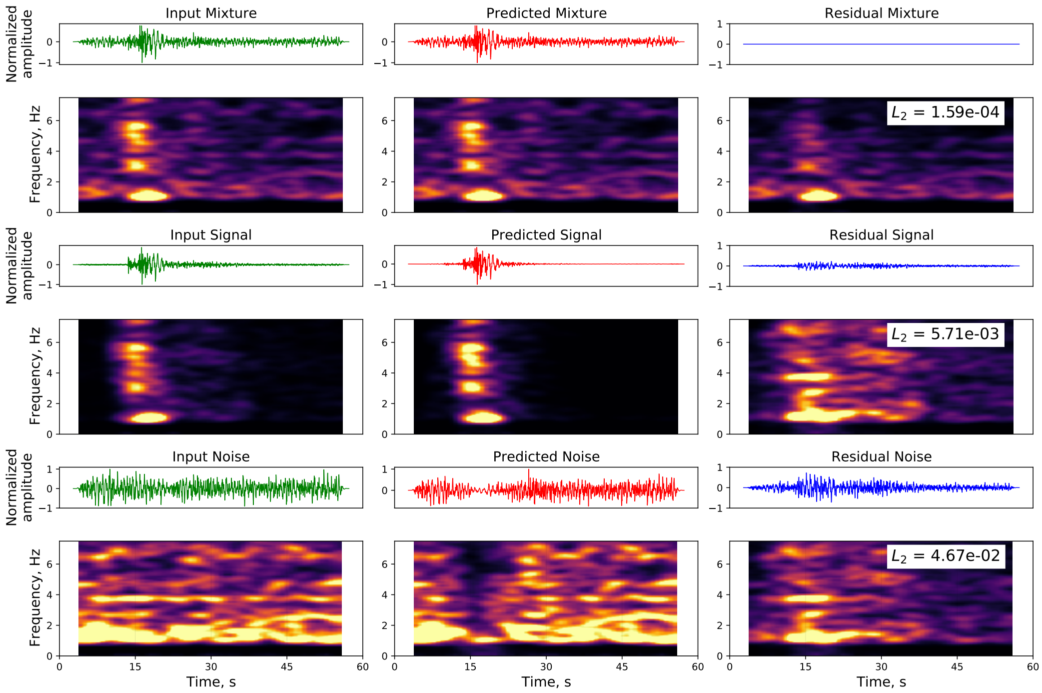

Figure 4. Results (waveforms and spectrograms) of the denoising model, performing denoising in noisy conditions (Signal-to-Noise ratio of a mixture, defined as the standard deviation of signal divided by the standard deviation of noise trace, equals to 1.69). Original signals are colored in green, predicted signals are colored in red, and residual is colored in blue. Top panel - input mixture, middle panel - separated signal, bottom panel - separated noise. L2 misfits (Mean Squared Error MSE = L2 $= \frac{1}{n}\sum(x_i - y_i)^2$, where x_i - input signal, y_i - predicted signal, n - number of signal pairs) are provided for each residual. SNR of denoised signal = 9.09.

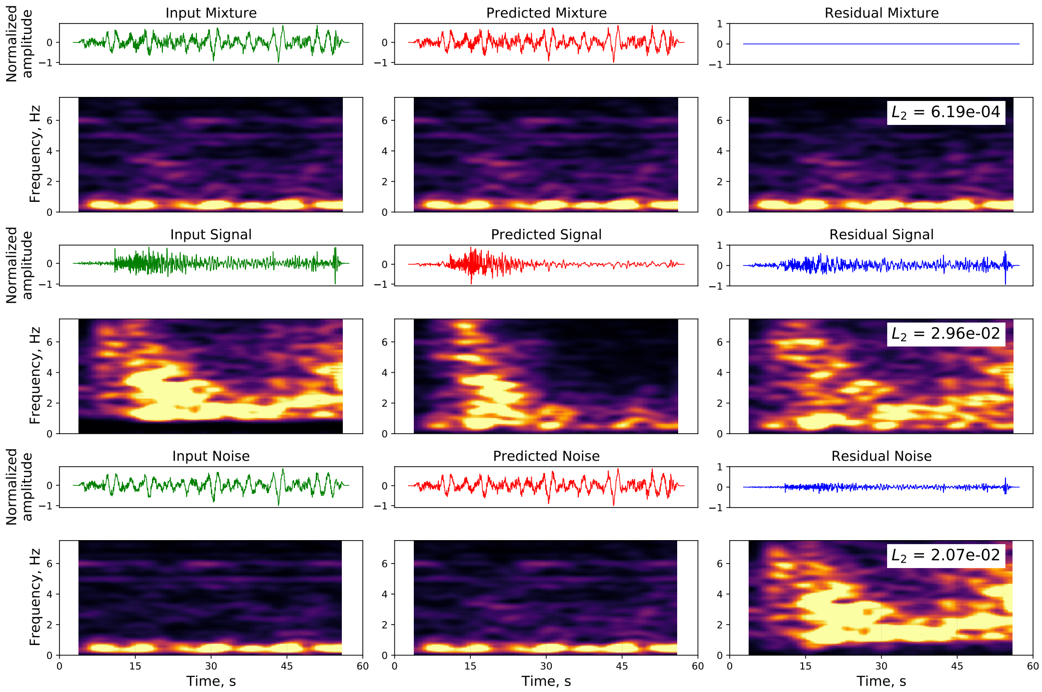

Figure 5. Results of the denoising model for moderately noisy conditions (otherwise as in Fig. 4). Signal-to-Noise ratio is 1.9. SNR of denoised signal = 7.01.

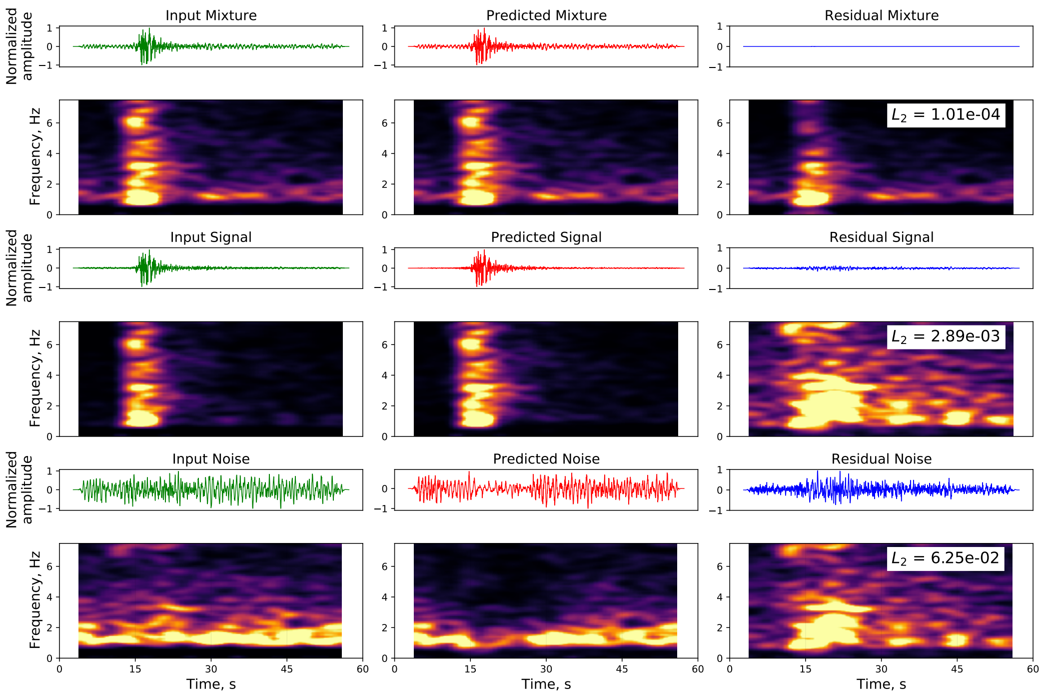

Figure 6. Results of the denoising model for weakly noisy conditions (otherwise as in Fig. 4). Signal-to-Noise ratio is 3.91. SNR of denoised signal = 7.58.

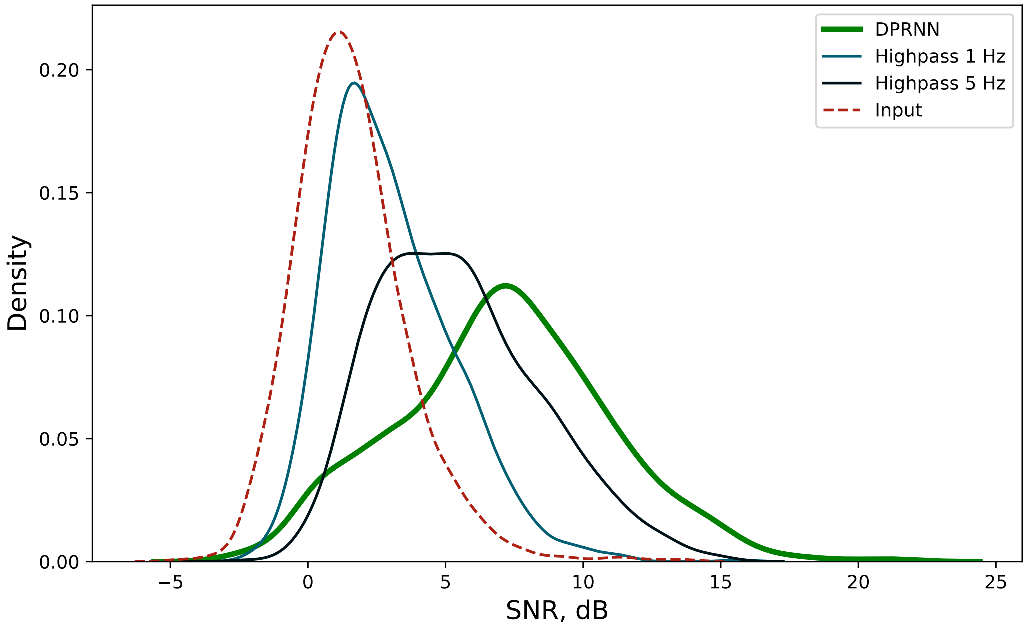

Figure 7. Kernel density estimate plot (histogram) of signal-to-noise ratios for raw data from the test set (in dashed red line) and denoised data (in green). We also compare our denoising capabilities with simple highpass filters for 1 Hz (blue) and 5 Hz (black). We observe that the Dual-Path Recurrent Neural Network (DPRNN) performs better (the higher the values - the better the result) than simple frequency filtering.

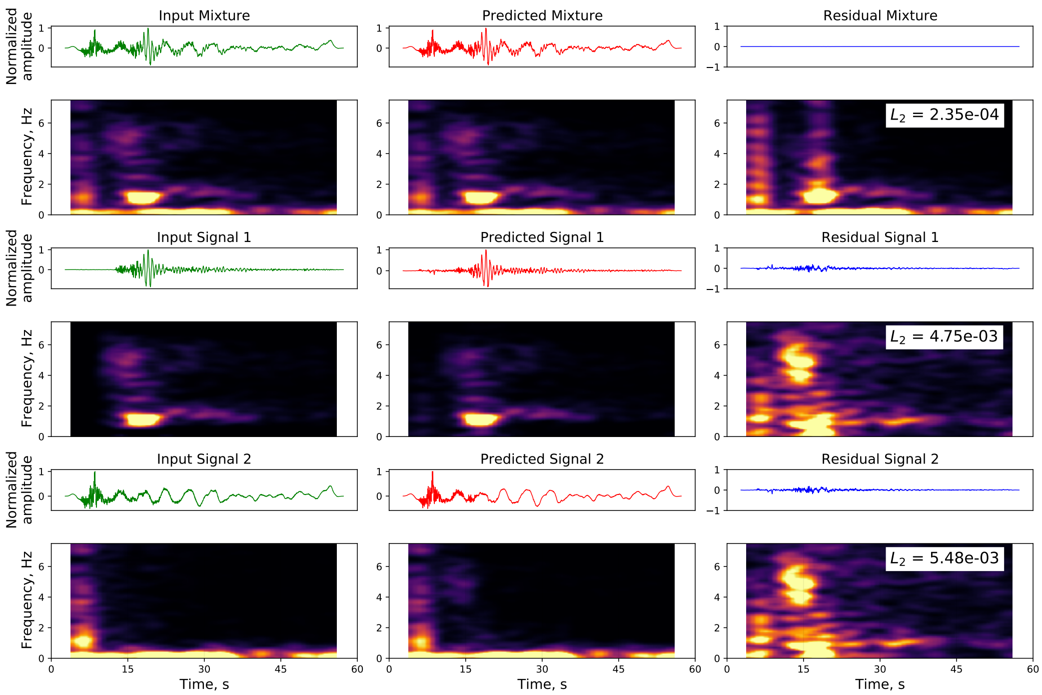

Figure 8. Results (waveforms and spectrograms) of source separation, applying proposed network. Original signals are colored in green, predicted signals are colored in red and residual signals are colored in blue. Misfits are provided as L2 value (L2 $= \frac{1}{n}\sum(x_i - y_i)^2$) for each residual. This example demonstrates an example where sources in the mixture are distinguishable.

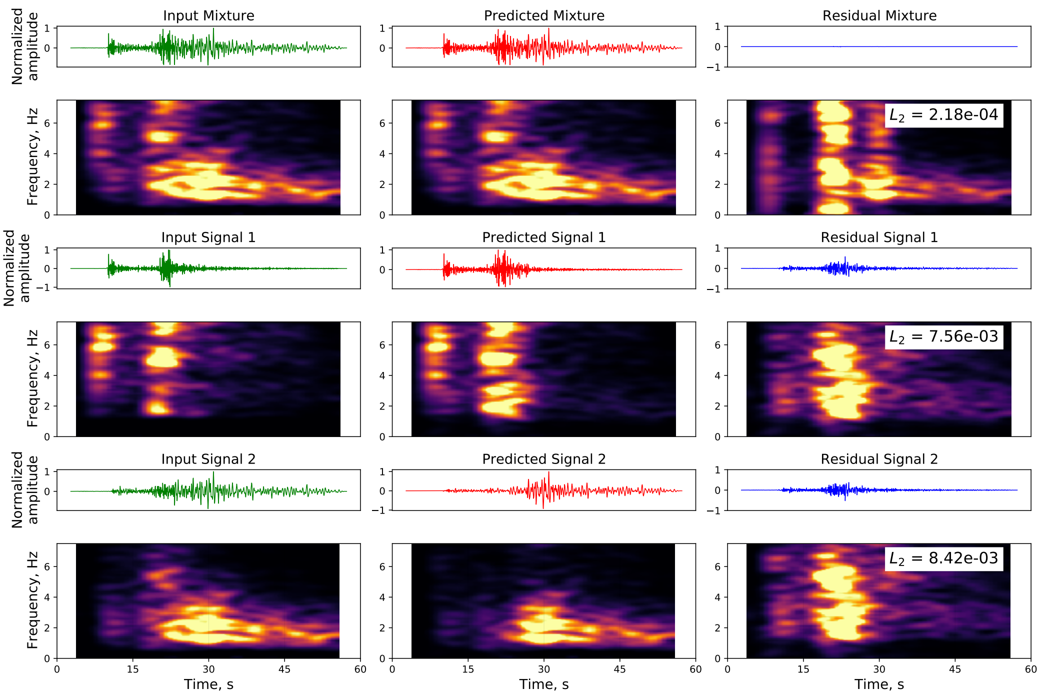

Figure 9. Results of source separation, for an example where sources in the mixture are overlapping in time, but have different frequency content (shown as in Fig. 8).

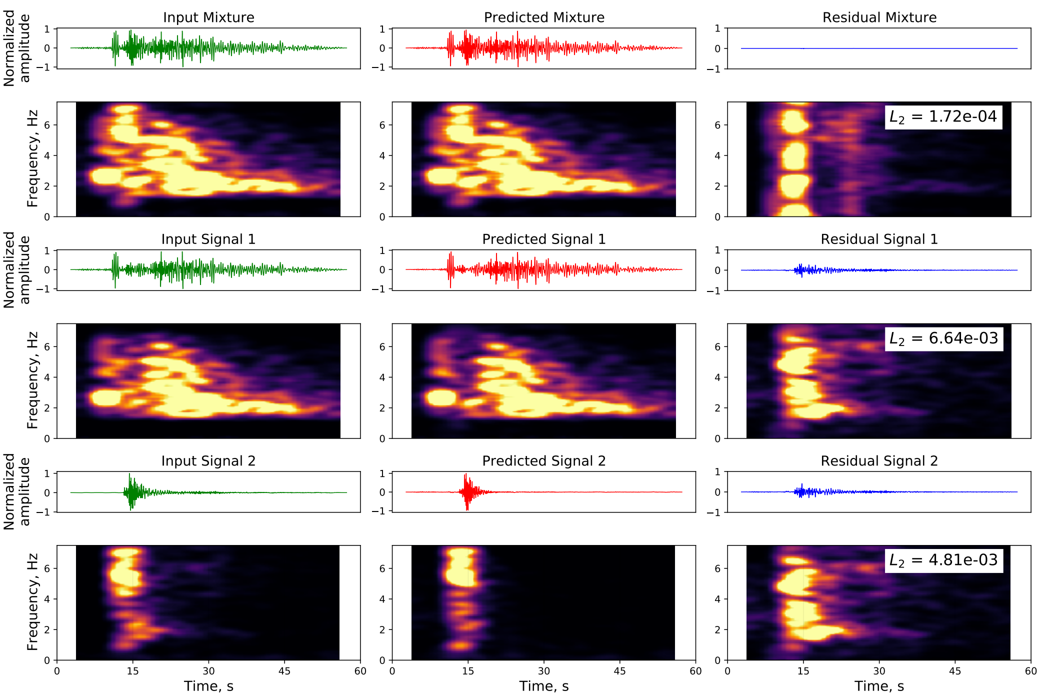

Figure 10. Results of source separation for an example where sources in the mixture are overlapping in both time and frequency content (shown as in Fig. 8).

Appendix A Components of the Neural Network

A01 Activation functions

Activation functions are widely used in neural networks as they equip neural networks with the ability to learn and map the non-linearity in the data and hence give neural networks their representational capacity. Because of this, in part, deep networks can approximate nearly everything (Csáji et al., 2001; Zhou, 2020). Following activation functions (applied element-wise) are used in the source separation network (see Fig. A1).

We use the Mish activation function, as it was shown to achieve better accuracy due to more stable gradients (Misra, 2019). Mish can be defined as $y = x{*}\tanh(\ln(1+e^x))$.

Besides Mish we use such activations as Tanh and Sigmoid. The Hyperbolic tangent function (Tanh) is defined as $y = \frac{\exp(x)-\exp(-x)}{\exp(x)+\exp(-x)}$ and the Sigmoid function is defined as $y = \frac{1}{1+\exp(-x)}$. Those activations are known to cause vanishing gradient problems (which can be mitigated by the means of e.g. skip-connections see (He et al., 2016)) and therefore are used with caution only in the Mask Estimation block (see Fig. 1) as a part of a gated-convolution operation (Oord et al., 2016).

A02 Normalization

In our network we use Group Normalization (Wu & He, 2018). It is applied over a batch of inputs as follows:

$$y = \frac{x - E[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta \tag{A1}$$

where x is the input data, $E[x]$ is the mean of the input data, $\mathrm{Var}[x]$ is the standard deviation, ϵ - is a small number (typically 10^{-8}) to ensure the absence of zeros in the denominator, γ and β are learnable (via back-propagation during the network training) per-channel parameter vectors.

A03 Recurrent Neural Networks

Convolutional Neural Networks (CNN) operate on the input data applying convolutions and various types of non-linear operations but are limited by their receptive fields (how much information is processed at each convolution, e.g. (Oord et al., 2016)). Instead, we use Recurrent Neural Networks (RNN) (Rumelhart et al., 1986) as building blocks inside the bigger network. RNNs allow sequential passage of information into the network (see Fig. A2), thus accumulating information at each time step and capturing temporal dependencies of the data presented to them. (Bengio et al., 1994) showed that networks trained with back-propagation algorithms achieve sub-optimal solutions taking into account only short-term dependencies without even looking at the long-term ones.

A04 Long Short Term Memory Cell

Since this long-term context is needed to achieve good performance of source separation we turn to a sub-class of RNN, which are specifically designed to overcome the long-term context loss problem - the Long Short Term Memory Cells (LSTM) (Hochreiter & Schmidhuber, 1997). Instead of a single simple layer (such as Tanh activation), they use a more complex structure consisting of 4 gates (see Fig. A3).

One of the obstacles that LSTM is facing is that by the time the sequence is passed through the cell, some information from the beginning looks less relevant to the network. To overcome this problem two LSTM Cells can be stacked together forming a Bi-Directional LSTM Cell (Schuster & Paliwal, 1997). The first LSTM would receive an input sequence x and the second LSTM would receive a reversed sequence $\hat{x}$ (see Fig. A4). Such configuration allows equal attention to the beginning and the end of the signal, resulting in a better quality of the model output.

In the context of DPRNN, since the actual separation operation is happening not with the input sequential signal, but rather an N-dimensional output of the encoder, it is important to learn "temporal" patterns not only in the "time" direction but also in the depth direction. For this purpose, we apply row- and column-wise BiLSTM Cells (see Fig. A5).

A05 Additional array manipulations

Fig. A6 demonstrates additional array manipulations necessary to operate the network. We utilize a Segmentation operation to unwrap sequential input of size (N, L) to a three-dimensional input of size (K, N, S). where N - is the number of channels, L - length of the sequence, S - length of the segment, and K - number of segments. We then apply an Overlay and Add operation which is essentially a reverse operation of Segmentation.

A06 Comparison with other methods

We compare our approach with another denoising method based on deep neural networks (see Zhu et al. (2019)). For this, we select 1000 previously unseen earthquake signals and 1000 previously unseen noise signals from the STEAD dataset. It is important to note that fair comparison is impossible in this particular case, since DeepDenoiser is trained on 30 s long samples with a frequency bandwidth of 100 Hz, and our model is trained with 60 s long samples with 30 Hz bandwidth. One needs to have identical data to perform a valid comparison. We try to mitigate this, by resampling 30 Hz data to 100 Hz for DeepDenoiser inference, but this is perhaps not sufficient. The other potential problem is that DeepDenoiser uses un-normalized counts, whether we use normalized displacement as an input. Results of the comparison are presented in Fig. A7, where we compare a particular sample denoised by both our method and DeepDenoiser and Fig. A8, where we compare the distribution of SI-SDR, SDR, and SNR for input mixture, denoised by DPRNN and denoised by DeepDenoiser.

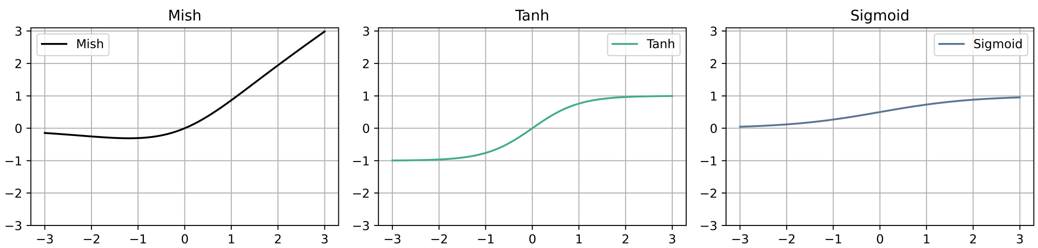

Figure A1. Activation functions used for model building. From left to right: Mish, Tanh and Sigmoid.

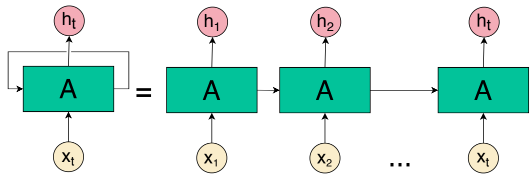

Figure A2. Recurrent Neural Network. Input data x at the time step t is fed to the network A (e.g. Tanh activation of concatenation x_t and previous output of the network), which outputs some value of h for the same time step and also passes this output information to the network A for the next time step.

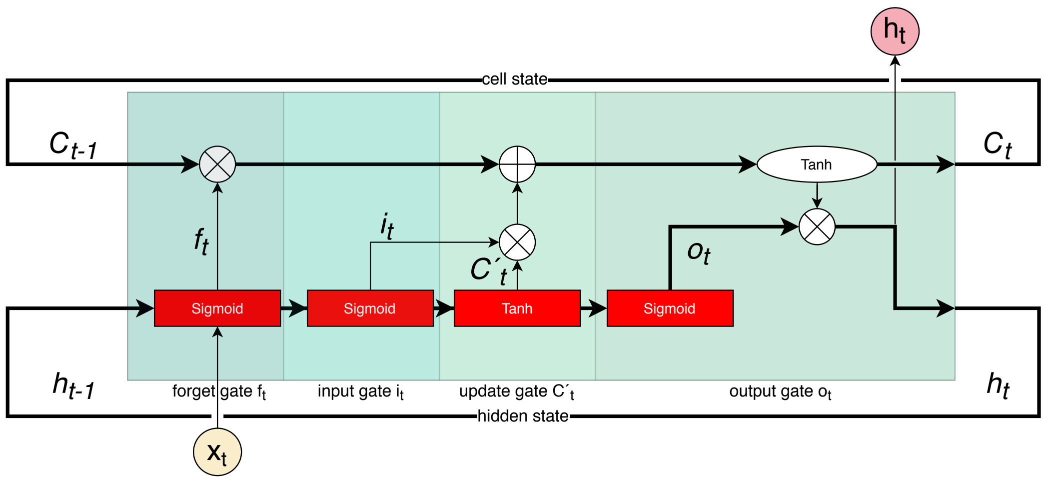

Figure A3. Long Short Term Memory Cell. Input data x at the time step t, previous cell state C_{t-1} and previous hidden state h_{t-1} are fed to the LSTM Cell. Cell outputs values of current cell state C_t and a value of current hidden state h_t. This process happens recurrently for each value of x. Red boxes depict network trainable layers, white shapes - point-wise operations (x - for multiplication, + for summation and tanh for hyperbolic tangents).

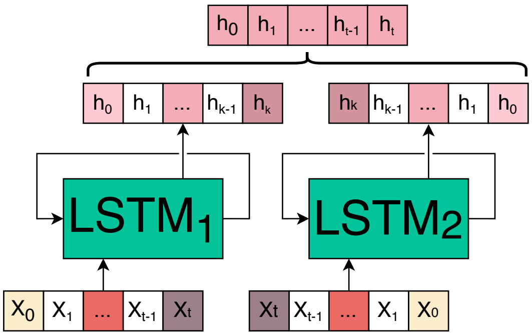

Figure A4. Bi-Directional Long Short Term Memory Cell. Two LSTM layers are stacked side-by-side. First receives an input sequence going from past to future, second LSTM recieves an input going in the reversed direction - from future to the past. Then cell states and hidden states of both cells are combined together (e.g. summation or concatenation).

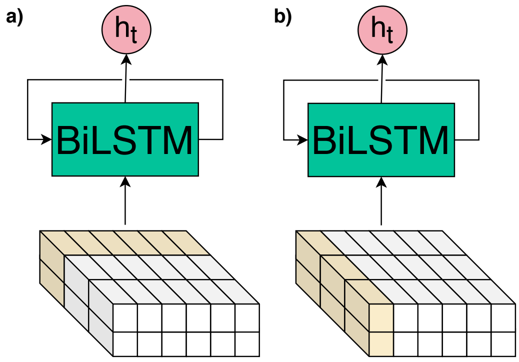

Figure A5. a) Row-wise BiLSTM. Each row of segmented output is processed through the Bi-directional LSTM cell. b) Column-wise BiLSTM. Each column of segmented output is processed through the Bi-directional LSTM cell.

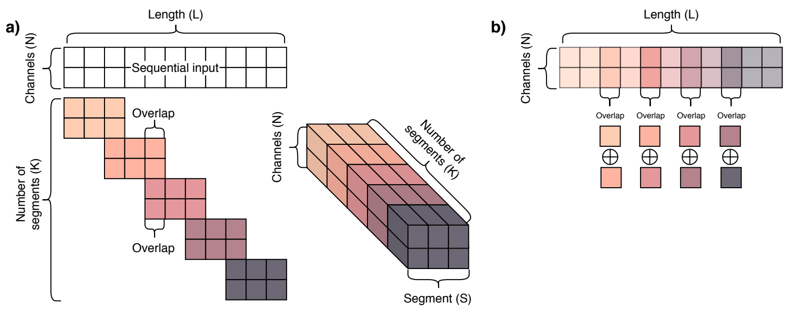

Figure A6. a) Segmentation. Sequential input of shape (N,L) is split into overlapping segments, which are then concatenated into 3D tensor of shape (K,N,S). b) Overlap and add. 3D tensor of shape (K,N,S) is split into segments. These signals are concatenated back into the sequence of shape (N,L). Overlapping parts of signals are added to each other.

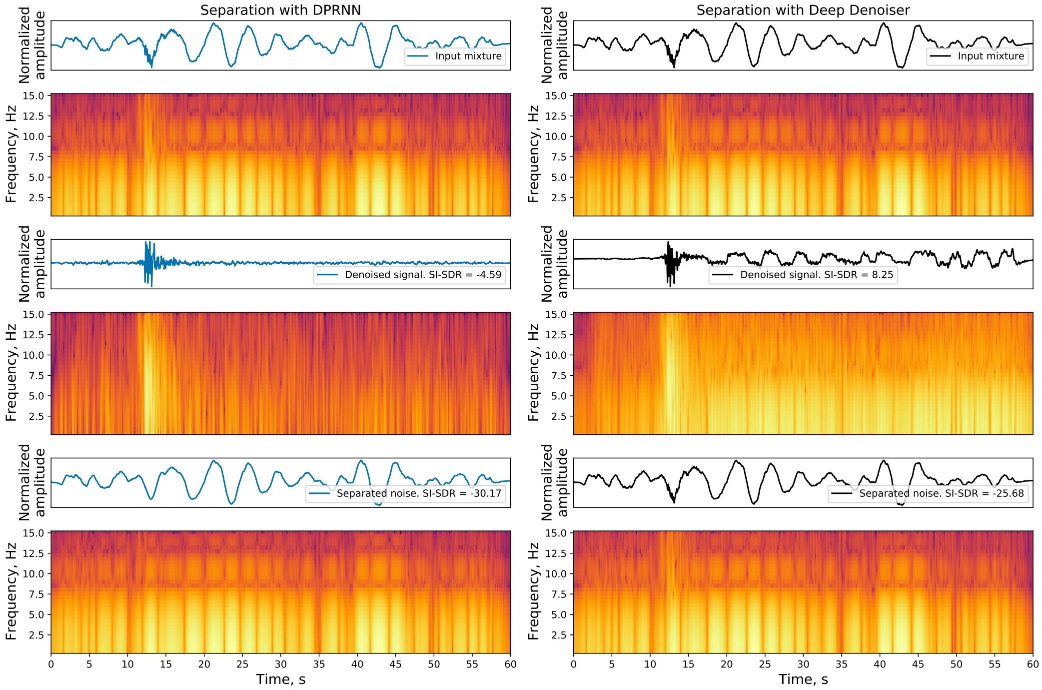

Figure A7. On left panels results of DPRNN denoising are presented. On right panels results of DeepDenoiser (Zhu et al., 2019) are presented. Top panels - input mixture, Middle panels - separated signal, Bottom panels - separated noise.

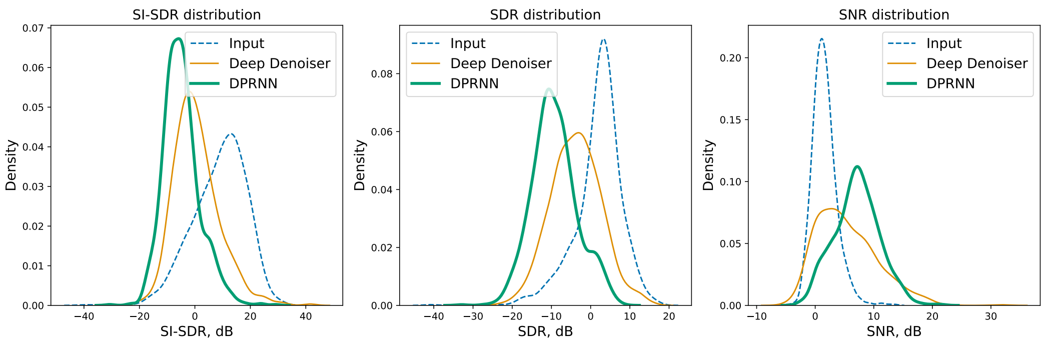

Figure A8. We compare denoising for DPRNN and DeepDenoiser in terms of SI-SDR, SDR and SNR. One can observe that DPRNN is able to achieve higher scores for both SI-SDR, SDR (the lower the value - the better the separated signal matches the original one) and SNR (the higher the values - the better).

Acronyms

BiLSTM Bidirectional LSTM

DPRNN Dual-Path Recurrent Neural Networks

ICA Independent-Component Analysis

LSTM Long-Short Term Memory

MSE Mean-Square Error

MUSIC MUltiple SIgnal Classification

RAdam Rectified Adam

RNN Recurrent Neural Network

SEDENOSS SEparating and DENOising Seismic Signals

SI-SDR Scale-Invariant Source to Distortion Ratio

SNR Signal-to-Noise ratio

STEAD STanford EArthquake Dataset

STFT Short-Time Fourier Transform

Tanh Hyperbolic Tangent

VSC Vienna Scientific Cluster

μPIT Utterance level Permutation Invariant Training