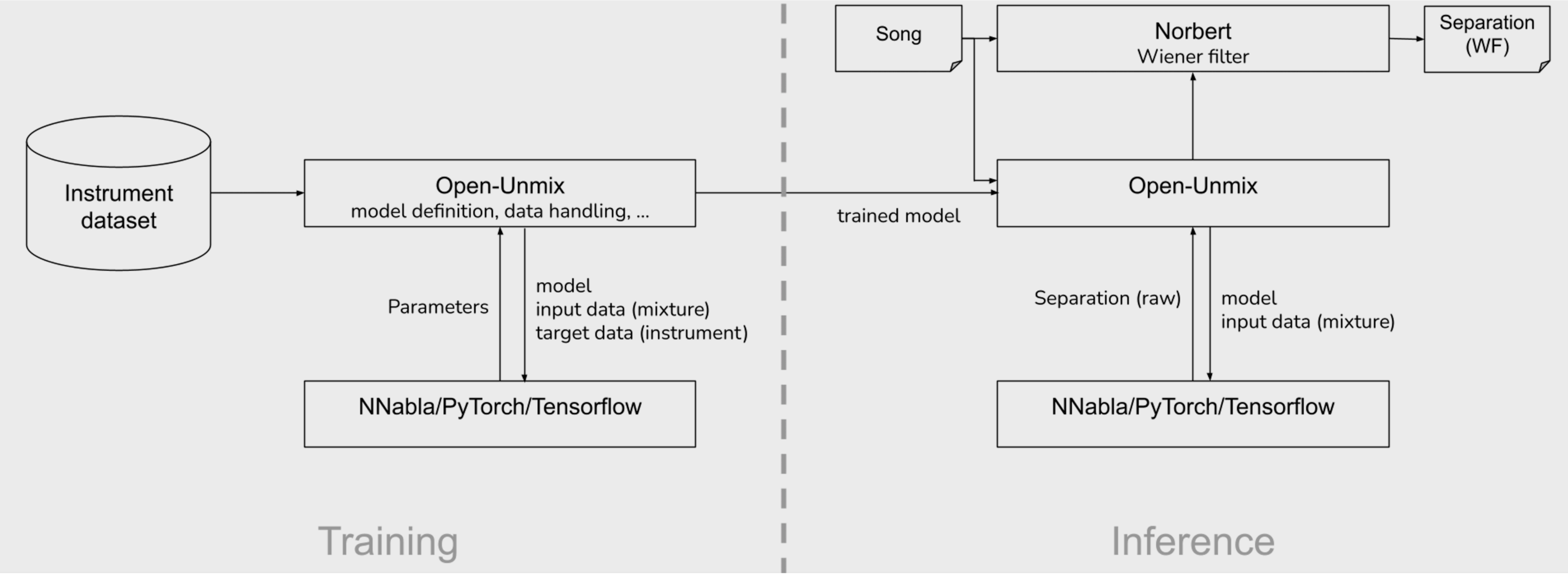
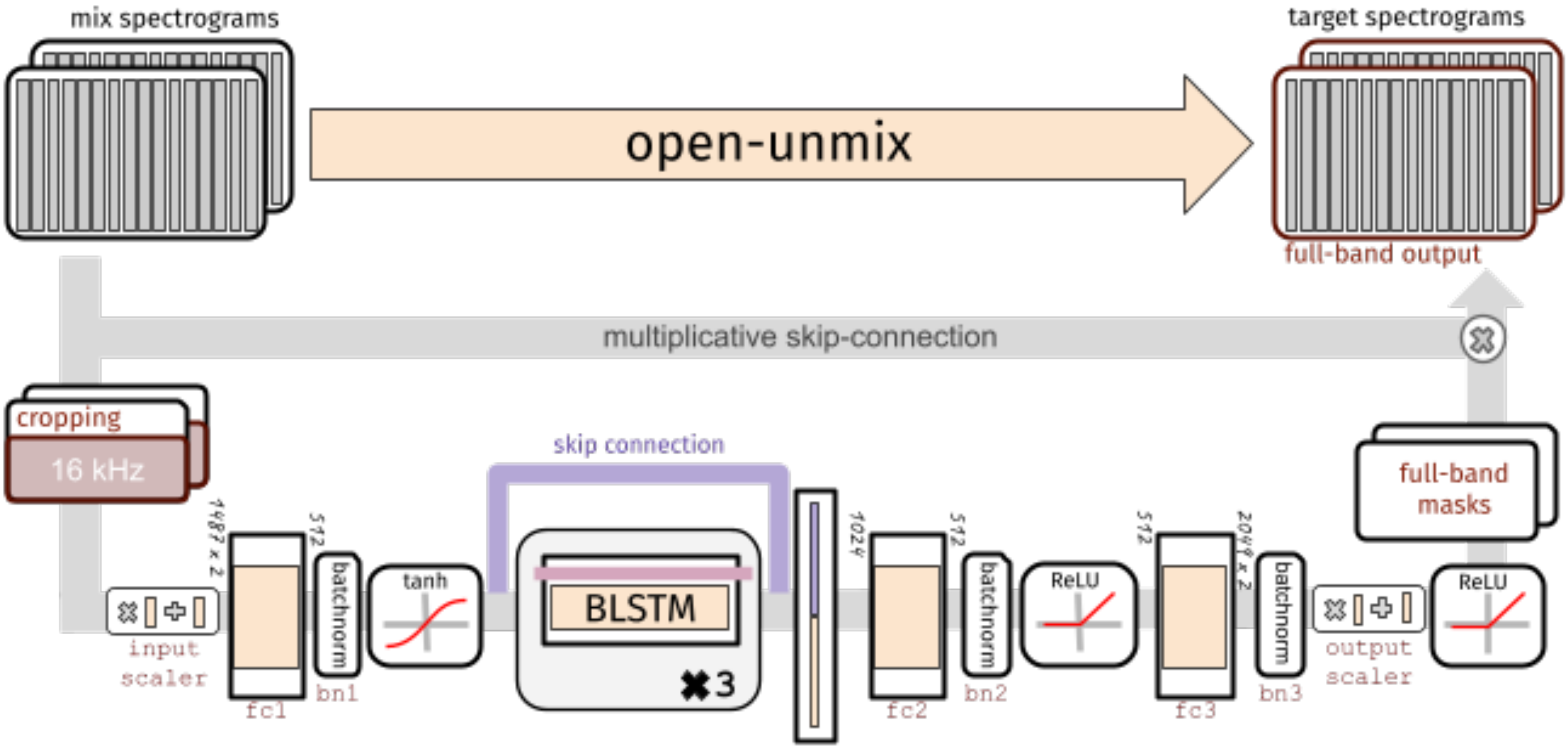
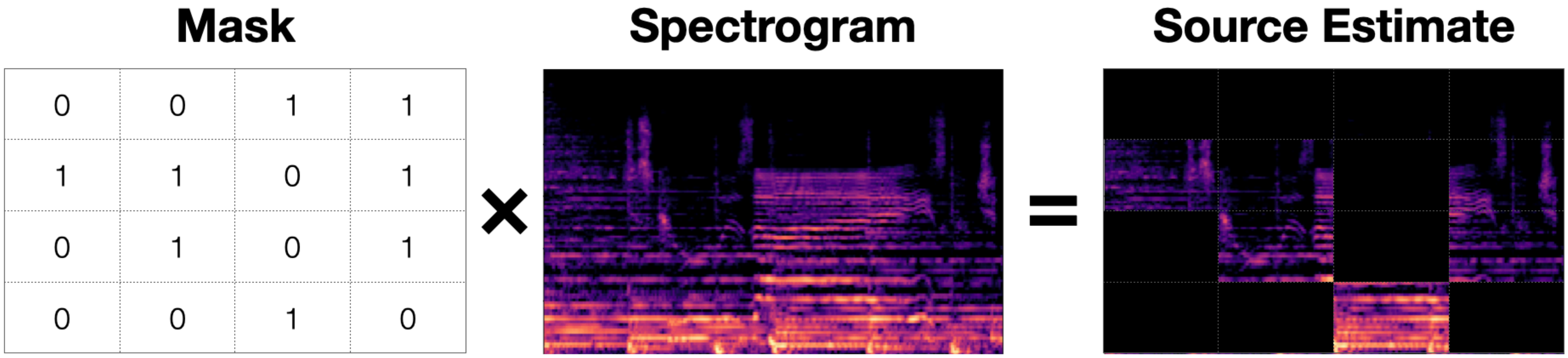
Open-unmix

https://github.com/sigsep/open-unmix-pytorch

https://hal.inria.fr/hal-02293689/document


Instrument dataset
Open-Unmix
model definition, data handling, ...
Parameters
model
input data (mixture)
target data (instrument)
NNabla/PyTorch/Tensorflow
Training
Song
Norbert
Wiener filter
Separation (WF)
trained model
Open-Unmix
Separation (raw)
model
input data (mixture)
NNabla/PyTorch/Tensorflow
Inference


mix spectrograms
open-unmix
target spectrograms
full-band output
multiplicative skip-connection
cropping
16 kHz
skip connection
full-band masks
input
scaler
1487 x 2
512
fc1
batchnorm
bn1
tanh
BLSTM
✖3
1024
fc2
512
batchnorm
bn2
ReLU
512
fc3
2049 x 2
batchnorm
bn3
output
scaler
ReLU



0	0	1	1
1	1	0	1
0	1	0	1
0	0	1	0

Open-Unmix operates in the time-frequency domain to perform its prediction. The input of the model is either:

- **models.Separator:**

A time domain signal tensor of shape (nb_samples, nb_channels, nb_timesteps), where nb_samples are the samples in a batch, nb_channels is 1 or 2 for mono or stereo audio, respectively, and nb_timesteps is the number of audio samples in the recording. In this case, the model computes STFTs with either torch or asteroid_filteranks on the fly.

- **models.OpenUnmix:**

The core open-unmix takes **magnitude spectrograms** directly (e.g. when pre-computed and loaded from disk). In that case, the input is of shape (nb_frames, nb_samples, nb_channels, nb_bins), where nb_frames and nb_bins are the time and frequency-dimensions of a Short-Time-Fourier-Transform.

Dimensionality reduction

- The LSTM is not operating on the original input spectrogram resolution. Instead, in the first step after the normalization, the network learns to compresses the frequency and channel axis of the model to reduce redundancy and make the model converge faster.

- The input spectrogram is *standardized* using the global mean and standard deviation for every frequency bin across all frames.

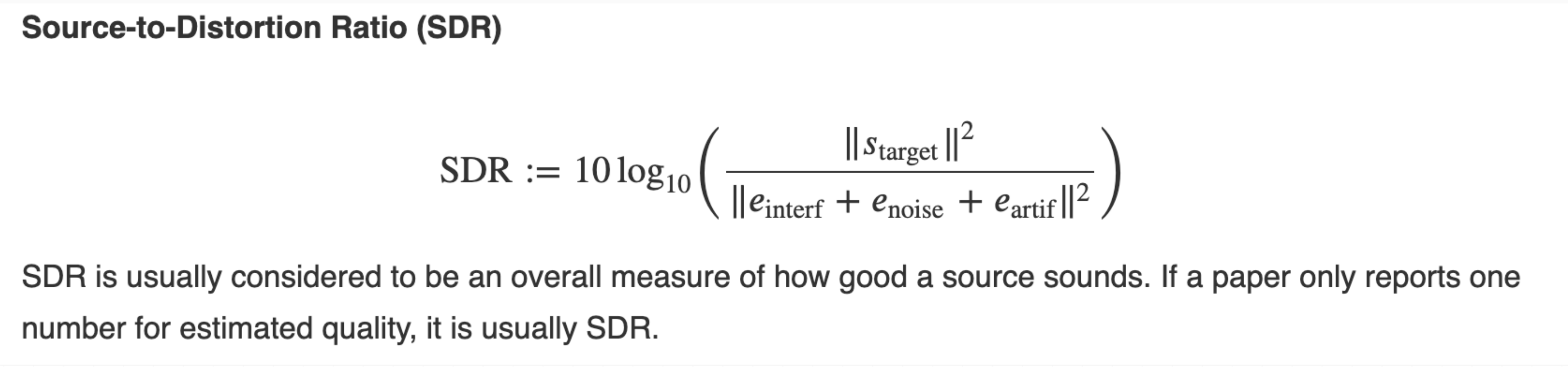
- Furthermore, we apply batch normalization in multiple stages of the model to make the training more robust against gain variation.

- *Open-unmix* comprises multiple models that are trained for each particular target.
- Each *Open-Unmix* source model is based on a three-layer bidirectional deep LSTM.
- Internally, the prediction is obtained by applying a mask on the input.
- The model is optimized in the magnitude domain using mean squared error.

Wiener Filter?

- models.Separator puts together Open-unmix spectrogram model for each desired target, and combines their output through a multichannel generalized Wiener filter, before application of inverse STFTs using torchaudio. The filtering is differentiable (but parameter-free) version of norbert. The separator is currently currently only used during inference.

Metric



Source-to-Distortion Ratio (SDR)

$$\text{SDR} := 10\log_{10}\left(\frac{\|s_{\text{target}}\|^2}{\|e_{\text{interf}} + e_{\text{noise}} + e_{\text{artif}}\|^2}\right)$$

SDR is usually considered to be an overall measure of how good a source sounds. If a paper only reports one number for estimated quality, it is usually SDR.

Results

target	SDR	SDR	SDR
model	UMX	UMXHQ	UMXL
vocals	6.32	6.25	7.21
bass	5.23	5.07	6.02
drums	5.73	6.04	7.15
other	4.02	4.28	4.89

Training API

- Details in the link:
  - https://github.com/sigsep/open-unmix-pytorch/blob/master/docs/training.md

- Examaple:

  -
    ```
    python train.py --root /data --dataset trackfolder_var --target-file vocals.flac --ext .wav
    ```

Changing the Model

- Details in the link:
  - https://github.com/sigsep/open-unmix-pytorch/blob/master/docs/extensions.md