Audio declipping with not-MIWAE

Adrien Tousnakhoff, Pierre Aguié
{adrien.tousnakhoff,pierre.aguie}@polytechnique.edu

February 4, 2025

1 Introduction

Clipping is a phenomenon in which an input signal is limited in amplitude. For a given input signal $\mathbf{x} \in \mathbb{R}^T$ and a clipping threshold $b > 0$ the clipped signal $\mathbf{x}^c \in \mathbb{R}^T$ is defined as:

$$\forall t \in \{1, \dots, T\}, \quad x_t^c = \begin{cases} x_t & \text{if } |x_t| < b, \\ b \operatorname{sign}(x_t) & \text{otherwise.} \end{cases} \tag{1}$$

In the case of audio, clipping can be caused by physical limitations of the recording device used to acquire the signal, and causes an audible degradation of the recording's quality. The task of audio declipping consists in retrieving the original signal from its clipped recording, in order to restore the audio's quality. Multiple approaches from the signal processing community have been developed [6], most of them using sparse approximation techniques to decompose the reliable part of the signal (i.e. the samples x_t, such that $|x_t| < b$) into sinusoidal atoms to reconstruct the full original signal [1]. However, as shown in [6], these methods are generally not able to accurately retrieve the original recording when the audio is heavily distorted, and can be computationally expensive.

In more recent works, Deep Latent Variable Models (DLVMs) have been used to impute missing values in datasets [5]. Most of these works make assumptions on the missing data patterns. The most common assumptions are that the data is either Missing Completely At Random (meaning that the pattern of missingness does not depend on the values of the dataset), or Missing At Random (meaning that the pattern of missingness depends only on the observed values). Both of these assumptions are not verified in the case of clipped audio data, for which the missing pattern depends on the missing value itself. This specific setting, called Missing Not At Random, is explored in depth in the paper "not-MIWAE: Deep Generative Modelling With Missing Not At Random Data" by Ipsen et al. [3]. This project aims to apply the method presented in [3] to the task of audio declipping, and takes inspiration from the experiment on image declipping presented in the paper.

1.1 Contributions statement

In this project, we first design a not-MIWAE architecture adapted to audio data with Missing Not At Random values. We then propose an experiment that show that not-MIWAE is a valid approach for handling Missing Not At Random data, and an experiment that studies the impact of latent dimension on imputation quality. The code used to implement our methods and run our experiments[1] is made from scratch, taking inspiration from the implementation of MIWAE made by the authors of [4] using `PyTorch`[2]. The repartition of work within our group is as follows:

- Pierre coded the generic not-MIWAE architecture presented in Section 2 and the missing model presented in Section 3.
- Adrien coded the convolutional architecture presented in 3, found the dataset used in our experiments and coded the preprocessing pipeline used to process our audio data.

[1] Our scripts are available here: github.com/pierreaguie/not-MIWAE-for-audio-inpainting
[2] The MIWAE implementation by the authors of [4] is available here: github.com/pamattei/miwae

2 Background

2.1 Assumptions on missing data patterns

Let $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top \in \mathbb{R}^{n \times T}$ a training dataset where $\mathbf{x}_1, \dots, \mathbf{x}_n$ are i.i.d copies of a random variable $\mathbf{x}$ which takes values in $\mathbb{R}^T$. In the case where part of the data is missing, each sample $\mathbf{x}_i$ can be separated into observed features and missing features, so that $\mathbf{x}_i = (\mathbf{x}_i^o, \mathbf{x}_i^m)$. The missing pattern can be encoded into a mask matrix $\mathbf{S} = (\mathbf{s}_1, \dots, \mathbf{s}_n)^\top \in \{0,1\}^{n \times T}$ such that $s_{i,j} = 1$ if and only if $x_{i,j}$ is observed. $\mathbf{s}_1, \dots, \mathbf{s}_n$ are i.i.d copies of a random variable $\mathbf{s}$, which takes values in $\{0,1\}^p$. Given a conditional distribution $p_\phi(\mathbf{s}|\mathbf{x}) = p_\phi(\mathbf{s}|\mathbf{x}^o, \mathbf{x}^m)$, there are three possible assumptions on the missing mechanism:

- if $p_\phi(\mathbf{s}|\mathbf{x}) = p_\phi(\mathbf{s})$ (the missing pattern does not depend on any value in the dataset), the data is said to be Missing Completely At Random (MCAR).
- if $p_\phi(\mathbf{s}|\mathbf{x}) = p_\phi(\mathbf{s}|\mathbf{x}^o)$ (the missing pattern depends only on the observed values), the data is Missing At Random (MAR).
- if $p_\phi(\mathbf{s}|\mathbf{x}) = p_\phi(\mathbf{s}|\mathbf{x}^o, \mathbf{x}^m)$ (the missing pattern depend both on the missing and observed data), the data is Missing Not At Random (MNAR).

In the case of clipped audio, since the audio samples are clipped if their absolute value goes beyond a clipping threshold, the missing pattern depends on the missing values, and the data is MNAR.

2.2 not-MIWAE: Importance-Weighed AutoEncoders for Missing Not at Random Data

The not-MIWAE model introduced in [3] is an Importance-Weighed Autoencoder (IWAE) [2] designed to handle MNAR data. The model aims to learn a parametric distribution $p_{\theta,\phi}(\mathbf{x}, \mathbf{s}) = p_\theta(\mathbf{x}) p_\phi(\mathbf{s}|\mathbf{x})$, by maximizing the log-likelihood of the observed data given the model parameters:

$$L\left((\mathbf{x}_i, \mathbf{s}_i)_{i=1,\dots,n}; \theta, \phi\right) = \sum_{i=1}^{n} \log p_{\theta,\phi}(\mathbf{x}_i^o, \mathbf{s}_i). \tag{2}$$

In practice, since the probability density function $p_{\theta,\phi}(\mathbf{x}, \mathbf{s})$ is defined for inputs of fixed size, $p_{\theta,\phi}(\mathbf{x}^o, \mathbf{s})$ is obtained by filling the missing values with 0s. Using a latent variable $\mathbf{z} \sim p(\mathbf{z})$, assuming that $p_\theta(\mathbf{x}|\mathbf{z}) = \prod_j p_\theta(x_j|\mathbf{z})$, the log-likelihoods can be written as:

$$\log p_{\theta,\phi}(\mathbf{x}^o, \mathbf{s}) = \log \int p_\phi(\mathbf{s}|\mathbf{x}^o, \mathbf{x}^m) p_\theta(\mathbf{x}^o|\mathbf{z}) p_\theta(\mathbf{x}^m|\mathbf{z}) p(\mathbf{z}) \mathrm{d}\mathbf{z} \mathrm{d}\mathbf{x}^m.$$

This integral is intractable, and cannot be used to optimize the model parameters θ and ϕ to maximize the log-likelihood $L\left((\mathbf{x}_i, \mathbf{s}_i)_{i=1,\dots,n}; \theta, \phi\right)$. IWAEs bypass this issue by introducing a learned parametric distribution $q_\gamma(\mathbf{z}|\mathbf{x}^o)$, which allow us to approximate $\log p_{\theta,\phi}(\mathbf{x}_i^o, \mathbf{s}_i)$ by performing importance sampling:

$$\begin{aligned}
\log p_{\theta,\phi}(\mathbf{x}^o, \mathbf{s}) &= \log \int \frac{p_\phi(\mathbf{s}|\mathbf{x}^o, \mathbf{x}^m) p_\theta(\mathbf{x}^o|\mathbf{z}) p(\mathbf{z})}{q_\gamma(\mathbf{z}|\mathbf{x}^o)} q_\gamma(\mathbf{z}|\mathbf{x}^o) p_\theta(\mathbf{x}^m|\mathbf{z}) \mathrm{d}\mathbf{z} \mathrm{d}\mathbf{x}^m \\
&= \log \mathbb{E}_{\mathbf{z} \sim q_\gamma(\mathbf{z}|\mathbf{x}^o), \mathbf{x}^m \sim p_\theta(\mathbf{x}^m|\mathbf{z})} \left[\frac{p_\phi(\mathbf{s}|\mathbf{x}^o, \mathbf{x}^m) p_\theta(\mathbf{x}^o|\mathbf{z}) p(\mathbf{z})}{q_\gamma(\mathbf{z}|\mathbf{x}^o)} \right].
\end{aligned}$$

By performing a Monte Carlo estimation of the expected value with K i.i.d. samples $\{\mathbf{z}_{i,k}, \mathbf{x}_{i,k}^m\}_k \sim q_\gamma(\mathbf{z}|\mathbf{x}_i^o) \cdot p_\theta(\mathbf{x}_i^m|\mathbf{z})$ for all $i = 1, \dots, N$, the objective to maximize becomes:

$$\mathcal{L}_K(\theta, \phi, \gamma) = \sum_{i=1}^{N} \mathbb{E}\left[\log \frac{1}{K} \sum_{k=1}^{K} \frac{p_\phi(\mathbf{s_i}|\mathbf{x}_i^o, \mathbf{x}_{i,k}^m) p_\theta(\mathbf{x}_i^o|\mathbf{z}_{i,k}) p(\mathbf{z}_{i,k})}{q_\gamma(\mathbf{z}_{i,k}|\mathbf{x}_i^o)} \right]$$

This new objective, which is a lower bound of the original log likelihood (2), can be used to optimize our model's parameters, and has the property of converging monotonically to the log-likelihood

(i.e. $\mathcal{L}_1(\theta,\phi,\gamma) \leq \cdots \leq \mathcal{L}_K(\theta,\phi,\gamma) \longrightarrow L\left((\mathbf{x}_i,\mathbf{s}_i)_{i=1,\ldots,n};\theta,\phi\right)$) under some specific conditions, as shown in [2].

In practical implementations, the prior distribution for the latent variable $\mathbf{z}$ is typically chosen to be a standard normal distribution $p(\mathbf{z}) \sim \mathcal{N}(0,I)$. The learned distributions $p_\theta(\mathbf{x}^m|\mathbf{z})$ and $\mathbf{q}_\gamma(\mathbf{z}|\mathbf{x}^o)$ are typically assumed to be Gaussian, and the distribution $p_\phi(\mathbf{s}|\mathbf{x})$ is assumed to be a Bernoulli distribution. In these cases, a neural network is used to predict the parameters of each conditional distribution, taking respectively as inputs $\mathbf{z}, \mathbf{x}^o$, and $(\mathbf{x}^o, \mathbf{x}^m)$.

2.3 Missing values imputation

Once we have a trained not-MIWAE model, we can use it to impute clipped values in the audio signal. As advised in Appendix B of [3], we estimate it by:

$$\hat{\mathbf{x}}^m := \sum_{k=1}^{K} \alpha_k \mathbb{E}\left[\mathbf{x}^m|\mathbf{z}_k\right],$$

where $\alpha_k := \frac{w_k}{\sum_{j=1}^K w_j}$, $w_k := \frac{p_\phi(\mathbf{s_i}|\mathbf{x}_i^o,\mathbf{x}_{i,k}^m)p_\theta(\mathbf{x}_i^o|\mathbf{z}_{i,k})p(\mathbf{z}_{i,k})}{q_\gamma(\mathbf{z}_{i,k}|\mathbf{x}_i^o)}$, and $\{\mathbf{z}_{i,k}, \mathbf{x}_{i,k}^m\}_{k=1,\ldots,K}$ are K i.i.d. samples of $q_\gamma(\mathbf{z}|\mathbf{x}_i^o) \cdot p_\theta(\mathbf{x}_i^m|\mathbf{z})$.

3 Method: not-MIWAE for Audio Declipping

In this section, we present the not-MIWAE model we created for audio declipping.

3.1 Autoencoder architecture

Two main families of neural network architectures exist for audio processing: those that take as input spectrograms of the audio data (typically obtained by performing Fast Fourier Transform on the waveforms), and those that directly take as input the raw waveform. Since the clipping condition depends on the amplitude of the audio signal, we focused on models that take waveforms as input. This choice will make the design of our missing model (see Subsection 3.2) much easier. The architecture of our not-MIWAE autoencoder is based on 1D convolutional layers, and is described in detail in Appendix A.

3.2 Missing model

As explained in [3], prior knowledge on the missing pattern and on the structure of the input data of the not-MIWAE can guide us in the design of our missing model $p_\phi(\mathbf{s}|\mathbf{x}^o,\mathbf{x}^m)$. We take inspiration from the missing model presented in Subsection 4.3 of [3], taking into account that contrary to the images in [3], our audio signals are clipped both from below and from above. This yields the following modified missing model:

$$p_\phi(\mathbf{s}|\mathbf{x}) = \prod_{j=1}^{T} p(s_j|x_j) = \prod_{j=1}^{T} \mathcal{B}(s_j|\pi_\phi(x_j)),$$

where $\phi = (W,b) \in \mathbb{R}^2$ and

$$\pi_\phi(x_j) := \frac{1}{1+\exp(-W(|x_j|-b))}.$$

In practice, this logistic missing model means that if a value x_j is above b or below $-b$, and if $W > 0$, then the sample x_j has a high chance of missing, which effectively emulates a clipping phenomenon at threshold b. This emulation is more faithful to the true clipping phenomenon described in (1) as W increases. Note that this missing model assumes that all the audio signals given as input to our not-MIWAE model are clipped at the same threshold b.

4 Experiments

4.1 Dataset and Data Processing Pipeline

The dataset used for our experiments is based on the MusicNet dataset[3], which is made of 330 recordings of classical music pieces of varying lengths.

We perform some preprocessing on the `.wav` files of MusicNet to build our dataset. We want our not-MIWAE model to take as inputs only fragments of the pieces and the missing mask. To do this, we first downsample the audio files from 44100 Hz to $f_s = 16000$ Hz. This sampling rate gives a good trade-off between the dimensionality of our dataset and audio quality. We then randomly select $n = 20000$ frames of length $T = 1024$ (which represents 64 ms of audio at $f_s = 16000$ Hz) among all the pieces of MusicNet. These frames are split evenly between a training and a validation set. We then normalize each individual frame, so that the maximal amplitude for each frame is 1. This allows us to build the matrix $\mathbf{X} \in \mathbb{R}^{n \times T}$. To sample the missing mask $\mathbf{S}$, we apply, for a fixed $(W, b) \in \mathbb{R}^2$, the following missing model:

$$s_{i,j}|x_{i,j} \sim \mathcal{B}\left(\frac{1}{1+\exp(-W(|x_j|-b))}\right).$$

For all of our experiments, the missing model of our not-MIWAE $p_\phi(\mathbf{s}|\mathbf{x})$ has fixed parameters, which are the same as the true missing model, as was done in the image clipping experiment presented in [3].

4.2 Metrics

To evaluate the performance of our model, we report the imputation Root Mean Squared Error (RMSE). This metric provides a clear and quantitative measure of how accurately our model retrieves audio by calculating the L_2 distance between the original missing values and their imputations. For an original signal $\mathbf{x}$ with M missing values, indexed by I_m, the RMSE between $\mathbf{x}$ and its imputation $\hat{\mathbf{x}}$ is:

$$\text{RMSE} := \frac{1}{M}\sum_{i \in I_m}(x_j - \hat{x}_j)^2.$$

4.3 Results

In this section, we showcase the results to the experiments we conducted on our dataset, using our not-MIWAE model. The first experiment shows the evolution of imputation RMSE during training. The second studies the impact of the clipping threshold b on the performance of our model. For all of these experiments, our not-MIWAE models had a latent dimension of 128, were trained using the Adam optimizer with parameters $(\beta_1, \beta_2) = (0.9, 0.999)$ and a learning rate of 10^{-4}, using a batch size of 256, and had $-\mathcal{L}_1$ as their loss function (i.e. we set $K = 1$).

4.3.1 Evolution of imputation RMSE during training

We first train a not-MIWAE model on our dataset, clipping values with $W = 5$ and $b = 0.5$, which results in about 20% of missing values in the training dataset. The evolution of the training and validation losses and the average validation imputation RMSE are shown in Figure 1. These plots confirm that optimizing the not-MIWAE bound is a conceptually sound approach to impute MNAR values, as the validation imputation RMSE and the validation loss dip simultaneously.

4.3.2 Imputation RMSE and clipping threshold

We propose an experiment to assess not-MIWAE's imputation performance at different proportions of missing data in the training and validation set. We set $W = 5$, and make the clipping threshold b vary from 0.3 to 0.9. This analysis allows us to assess the model's performance across varying

[3]This dataset is available at https://www.kaggle.com/datasets/imsparsh/musicnet-dataset.

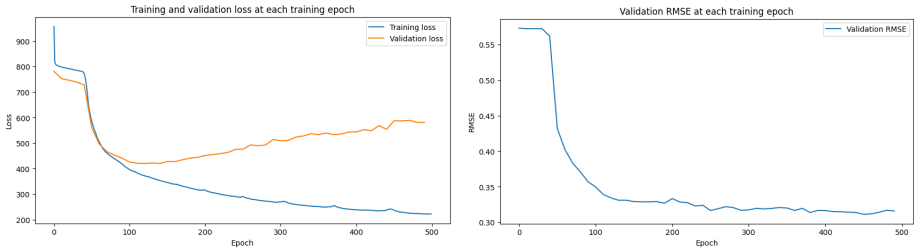

Figure 1: Left: training and validation losses at each training epoch. Right: validation RMSE at each training epoch. The validation imputation RMSE decreases during the training, confirming the validity of the not-MIWAE approach.

levels of signal degradation. By correlating the RMSE with the extent of clipping, we can better understand the model's robustness and effectiveness in reconstructing audio signals under different clipping scenarios. The results are reported in Figure 2.

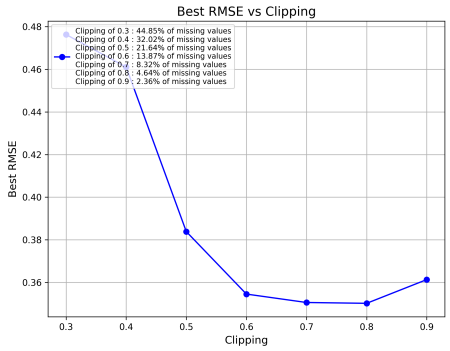

Figure 2: Average validation imputation RMSE with respect to the clipping threshold.

The results correspond to what we are expecting intuitively. The reconstruction is of lower quality with a low clipping threshold as there are too many missing values, and becomes better as the clipping threshold augments with a slight decay at the end.

4.4 Quality of the imputations

We show some of our model's imputations on Figure 3 to assess the faithfulness of the reconstructions to the original audios, clipped using $W = 5, b = 0.6$. We can see the imputations are fairly close to the original signals, except for some outliers. This confirms that our model is generally able to reconstruct clipped audio fairly faithfully, despite some failures for some missing samples.

To determine the audio quality of full-length declipped signals, we cut a single audio in multiple frames, and listened to the aggregated reconstructed frames by our model. The reconstructed signal presents some noisy artifacts that are not present in the original audio, showing that the reconstruction is not totally faithful. The spectrograms of both the original and the reconstructed audio are shown in Figure 4. The reconstructed audio present much more noise, which matches to what we hear. To further evaluate the reconstruction, we look at the spectral convergence, which quantifies how close the spectrum of the reconstructed signal is to the original one:

$$\frac{\|S_{reconstructed} - S_{original}\|_F}{\|S_{original}\|_F}$$

where $S_{reconstructed}$ and $S_{original}$ are the frequency domain representations of the audios and $\|.\|_F$ is the Frobenius norm. The spectral convergence between the two spectrograms is 0.297, which means that 70.3% of the spectral content matches the original. This suggest that the method preserved a significant amount of spectral information but that there is still some discrepancies.

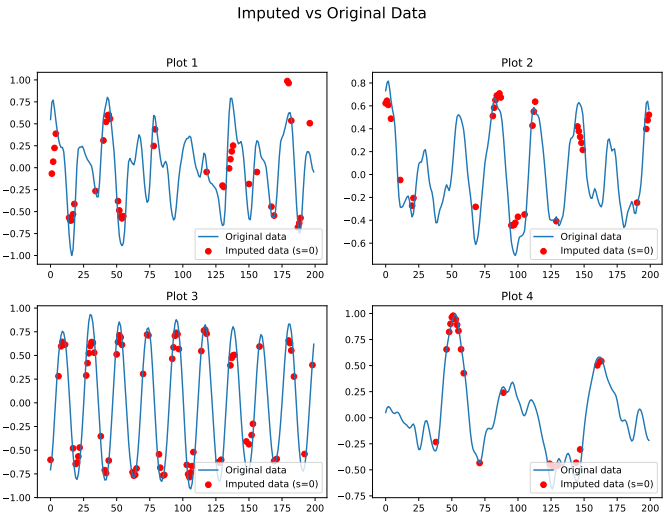

Figure 3: Imputations (in red) of missing values of different waveforms from our validation set.

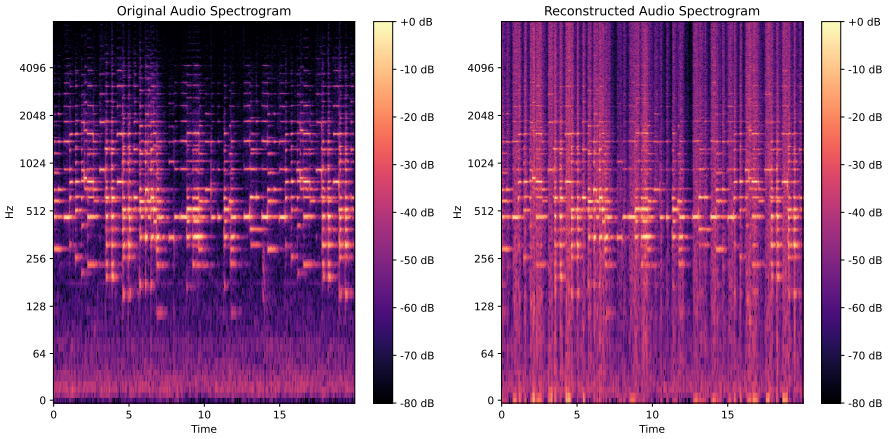

Figure 4: Spectrograms of the original audio (left) and the reconstructed audio (right).

5 Conclusion

Our experiments show that not-MIWAE is a sound approach to handle datasets where the missing pattern is MNAR. Our model was able to reconstruct clipped signals fairly faithfully, even though we trained it on a small dataset, on a limited number of epochs and parameters. However, when compared to other audio declipping frameworks, not-MIWAE suffers from some limitations.

First, classical Sparse Coding methods typically do not require to be trained on a large corpus of clipped audio samples to learn the declipping process, which is the case for our not-MIWAE. Secondly, as it is presented in this report, each of our not-MIWAE models can only handle one missing data distribution each (i.e. each of our model can only declip signals clipped at a specific threshold). This issue could be solved by choosing a more intelligent missing model, which does not sample s_j by only taking into account the value of x_j, but also compares it to all of the other values of the signal.

References

[1] Amir Adler et al. "Audio Inpainting". In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.3 (2012), pp. 922–932. DOI: 10.1109/TASL.2011.2168211.

[2] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. *Importance Weighted Autoencoders*. 2016. arXiv: 1509.00519 [cs.LG]. URL: https://arxiv.org/abs/1509.00519.

[3] Niels Bruun Ipsen, Pierre-Alexandre Mattei, and Jes Frellsen. *not-MIWAE: Deep Generative Modelling with Missing not at Random Data*. 2021. arXiv: 2006.12871 [stat.ML]. URL: https://arxiv.org/abs/2006.12871.

[4] Pierre-Alexandre Mattei and Jes Frellsen. *MIWAE: Deep Generative Modelling and Imputation of Incomplete Data*. 2019. arXiv: 1812.02633 [stat.ML]. URL: https://arxiv.org/abs/1812.02633.

[5] Alfredo Nazabal et al. *Handling Incomplete Heterogeneous Data using VAEs*. 2020. arXiv: 1807.03653 [cs.LG]. URL: https://arxiv.org/abs/1807.03653.

[6] Pavel Zaviska et al. "A Survey and an Extensive Evaluation of Popular Audio Declipping Methods". In: *IEEE Journal of Selected Topics in Signal Processing* 15.1 (Jan. 2021), pp. 5–24. ISSN: 1941-0484. DOI: 10.1109/jstsp.2020.3042071. URL: http://dx.doi.org/10.1109/JSTSP.2020.3042071.

A Architecture of our not-MIWAE model

Layer	Size of output
Input x	$(1, T)$
Conv1D	$(16, T/2)$
Conv1D	$(32, T/4)$
Conv1D	$(64, T/8)$
Conv1D	$(128, T/16)$
Flatten	$(128 \cdot T/16)$
Linear: $\mu_{\mathbf{z}}$	(latent_dim)
Linear: $\log \sigma^2_{\mathbf{z}}$	(latent_dim)

Layer	Size of output
Input z	(latent_dim)
Linear	$(128 \cdot T/16)$
Reshape	$(128, T/16)$
TransposedConv1D	$(64, T/8)$
TransposedConv1D	$(32, T/4)$
TransposedConv1D	$(16, T/2)$
TransposedConv1D	$(1, T)$
Sigmoid: $\mu_{\mathbf{x}}$	$(1, T)$
TransposedConv1D	$(16, T/2)$
TransposedConv1D: $\log \sigma^2_{\mathbf{x}}$	$(1, T)$

Table 1: Architecture of our not-MIWAE encoder (left) and decoder (right) for audio declipping. Between every layer, a ReLU activation is applied. The outputs of the encoder are the mean and the log-variance of $q_\gamma(\mathbf{z}|\mathbf{x}^o) \sim \mathcal{N}(\mu_{\mathbf{z}}, \sigma^2_{\mathbf{z}})$, and the outputs of the decoder are the mean and the log-variance of $p_\theta(\mathbf{x}|\mathbf{z}) \sim \mathcal{N}(\mu_{\mathbf{x}}, \sigma^{\in}_{\mathbf{x}})$