# Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models

**Fanny Roche**[1,3] **Thomas Hueber**[3] **Samuel Limier**[1] **Laurent Girin**[2,3]
[1]Arturia, Meylan, France [2]Inria Grenoble Rhône-Alpes, France
[3]Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, Grenoble, France
fanny.roche@gipsa-lab.fr

## ABSTRACT

This study investigates the use of non-linear unsupervised dimensionality reduction techniques to compress a music dataset into a low-dimensional representation which can be used in turn for the synthesis of new sounds. We systematically compare (shallow) autoencoders (AEs), deep autoencoders (DAEs), recurrent autoencoders (with Long Short-Term Memory cells – LSTM-AEs) and variational autoencoders (VAEs) with principal component analysis (PCA) for representing the high-resolution short-term magnitude spectrum of a large and dense dataset of music notes into a lower-dimensional vector (and then convert it back to a magnitude spectrum used for sound resynthesis). Our experiments were conducted on the publicly available multi-instrument and multi-pitch database NSynth. Interestingly and contrary to the recent literature on image processing, we can show that PCA systematically outperforms shallow AE. Only deep and recurrent architectures (DAEs and LSTM-AEs) lead to a lower reconstruction error. The optimization criterion in VAEs being the sum of the reconstruction error and a regularization term, it naturally leads to a lower reconstruction accuracy than DAEs but we show that VAEs are still able to outperform PCA while providing a low-dimensional latent space with nice "usability" properties. We also provide corresponding objective measures of perceptual audio quality (PEMO-Q scores), which generally correlate well with the reconstruction error.

## 1. INTRODUCTION

Deep neural networks, and in particular those trained in an unsupervised (or self-supervised) way such as autoencoders [1] or GANs [2], have shown nice properties to extract latent representations from large and complex datasets. Such latent representations can be sampled to generate new data. These types of models are currently widely used for image and video generation [3–5]. In the context of a project aiming at designing a music sound synthesizer driven by high-level control parameters and propelled by data-driven machine learning, we investigate the use of such techniques for music sound generation as an alternative to classical music sound synthesis techniques like additive synthesis, subtractive synthesis, frequency modulation, wavetable synthesis or physical modeling [6].

So far, only a few studies in audio processing have been proposed in this line, with a general principle that is similar to image synthesis/transformation: projection of the signal space into a low-dimensional latent space (encoding or embedding), modification of the latent coefficients, and inverse transformation of the modified latent coefficients into the original signal space (decoding).

In [7, 8], the authors implemented this principle with autoencoders to process normalized magnitude spectra. An autoencoder (AE) is a specific type of artificial neural network (ANN) architecture which is trained to reconstruct the input at the output layer, after passing through the latent space. Evaluation was made by computing the mean squared error (MSE) between the original and the reconstructed magnitude spectra.

In [9], NSynth, an audio synthesis method based on a time-domain autoencoder inspired from the WaveNet speech synthesizer [10] was proposed. The authors investigated the use of this model to find a high-level latent space well-suited for interpolation between instruments. Their autoencoder is conditioned on pitch and is fed with raw audio from their large-scale multi-instrument and multi-pitch database (the NSynth dataset). This approach led to promising results but has a high computational cost.

Another technique to synthesize data using deep learning is the so-called variational autoencoder (VAE) originally proposed in [11], which is now popular for image generation. A VAE can be seen as a probabilistic/generative version of an AE. Importantly, in a VAE, a prior can be placed on the distribution of the latent variables, so that they are well suited for the control of the generation of new data. This has been recently exploited for the modeling and transformation of speech signals [12, 13] and also for music sounds synthesis [14], incorporating some fitting of the latent space with a perceptual timbre space. VAEs have also been recently used for speech enhancement [15–17].

In line with the above-presented studies, the goal of the present paper is i) to provide an extensive comparison of several autoencoder architectures including shallow, deep, recurrent and variational autoencoders, with a systematic comparison to a linear dimensionality reduction technique, in the present case Principal Component Analysis (PCA) (to the best of our knowledge, such comparison of non-linear approaches with a linear one has never been done in previous studies). This is done using both an objec-

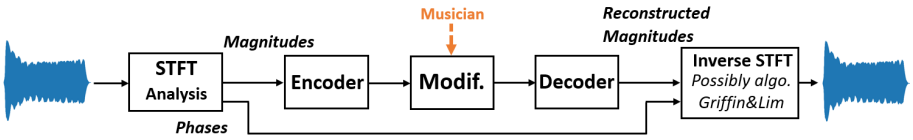



Figure 1: Global diagram of the sound analysis-transformation-synthesis process.

tive physical measure (root mean squared error – RMSE) and an objective perceptual measure (PEMO-Q [18]); ii) to compare the properties of the latent space in terms of correlation between the extracted dimensions; and iii) to illustrate how interpolation in the latent space can be performed to create interesting hybrid sounds.

## 2. METHODOLOGY

The global methodology applied for (V)AE-based analysis-transformation-synthesis of audio signals in this study is in line with previous works [7, 8, 12, 13]. It is illustrated in Fig. 1 and is described in the next subsections.

### 2.1 Analysis-Synthesis

First, a Short-Term Fourier Transform (STFT) analysis is performed on the input audio signal. The magnitude spectra are sent to the model (encoder input) on a frame-by-frame basis, and the phase spectra are stored for the synthesis stage. After possible modifications of the extracted latent variables (at the bottleneck layer output, see next subsection), the output magnitude spectra is provided by the decoder. The output audio signal is synthesized by combining the decoded magnitude spectra with the phase spectra, and by applying inverse STFT with overlap-add. If the latent coefficients are not modified in between encoding and decoding, the decoded magnitude spectra are close to the original ones and the original phase spectra can be directly used for good quality waveform reconstruction. If the latent coefficients are modified so that the decoded magnitude spectra become different from the original one, then the Griffin & Lim algorithm [19] is used to estimate/refine the phase spectra (the original phase spectra are used for initialization) and finally reconstruct the time-domain signal. A few more technical details regarding data pre-processing are given in Section 3.2.

### 2.2 Dimensionality Reduction Techniques

**Principal Component Analysis**: As a baseline, we investigated the use of PCA to reduce the dimensionality of the input vector $\mathbf{x}$. PCA is the optimal linear orthogonal transformation that provides a new coordinate system (i.e. the latent space) in which basis vectors follow modes of greatest variance in the original data [20].

**Autoencoder**: An AE is a specific kind of ANN traditionally used for dimensionality reduction thanks to its diabolo shape [21], see Fig. 2. It is composed of an encoder and a decoder. The encoder maps a high-dimensional low-level input vector $\mathbf{x}$ into a low-dimensional higher-level latent vector $\mathbf{z}$, which is assumed to nicely encode properties or attributes of $\mathbf{x}$. Similarly, the decoder reconstructs an estimate $\hat{\mathbf{x}}$ of the input vector $\mathbf{x}$ from the latent vector $\mathbf{z}$. The model is written as:

$$\mathbf{z} = f_{\text{enc}}(\mathbf{W}_{\text{enc}}\mathbf{x}+\mathbf{b}_{\text{enc}}) \quad \text{and} \quad \hat{\mathbf{x}} = f_{\text{dec}}(\mathbf{W}_{\text{dec}}\mathbf{z}+\mathbf{b}_{\text{dec}}),$$

where $f_{\text{enc}}$ and $f_{\text{dec}}$ are (entry-wise) non-linear activation functions, $\mathbf{W}_{\text{enc}}$ and $\mathbf{W}_{\text{dec}}$ are weight matrices and $\mathbf{b}_{\text{enc}}$ and $\mathbf{b}_{\text{dec}}$ are bias vectors. For regression tasks (such as the one considered in this study), a linear activation function is generally used for the output layer.

At training time, the weight matrices and the bias vectors are learned by minimizing some cost function over a training dataset. Here we consider the mean squared error (MSE) between the input $\mathbf{x}$ and the output $\hat{\mathbf{x}}$.

The model can be extended by adding hidden layers in both the encoder and decoder to create a so-called deep autoencoder (DAE), as illustrated in Fig. 2. This kind of architecture can be trained globally (end-to-end) or layer-by-layer by considering the DAE as a stack of shallow AEs [1, 22].

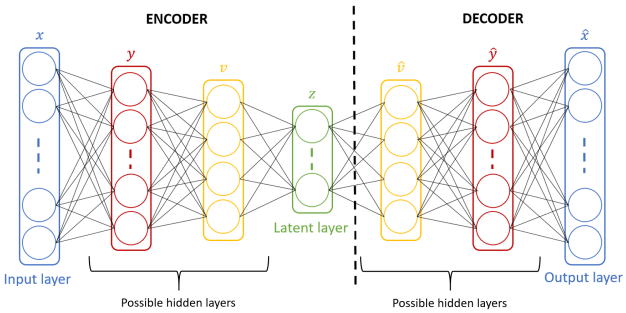



Figure 2: General architecture of a (deep) autoencoder.

**LSTM Autoencoder**: In a general manner, a recurrent neural network (RNN) is an ANN where the output of a given hidden layer does not depend only on the output of the previous layer (as in a feedforward architecture) but also on the internal state of the network. Such internal state can be defined as the output of each hidden neuron when processing the previous input observations. They are thus well-suited to process time series of data and capture their time dependencies. Such networks are here expected to extract latent representations that encode some aspects of the sound dynamics. Among different existing RNN architectures, in this study we used the Long Short-Term Memory (LSTM) network [23], which is known to tackle correctly the so-called vanishing gradient problem in RNNs [24]. The structure of the model depicted in Fig. 2 still holds while replacing the classical neuronal cells by LSTM cells, leading to a LSTM-AE. The cost function to optimize remains the same, i.e. the MSE between the input $\mathbf{x}$ and the

output $\hat{\mathbf{x}}$. However, the model is much more complex and has more parameters to train [23].

**Variational Autoencoder**: A VAE can be seen as a probabilistic AE which delivers a parametric model of the data distribution, such as:

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z}),$$

where $\theta$ denotes the set of distribution parameters. In the present context, the likelihood function $p_\theta(\mathbf{x}|\mathbf{z})$ plays the role of a probabilistic decoder which models how the generation of observed data $\mathbf{x}$ is conditioned on the latent data $\mathbf{z}$. The prior distribution $p_\theta(\mathbf{z})$ is used to structure (or regularize) the latent space. Typically a standard Gaussian distribution $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ is used, where $\mathbf{I}$ is the identity matrix [11]. This encourages the latent coefficients to be mutually orthogonal and lie on a similar range. Such properties may be of potential interest for using the extracted latent coefficients as control parameters of a music sound generator. The likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ is defined as a Gaussian density:

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\theta(\mathbf{z}), \boldsymbol{\sigma}^2_\theta(\mathbf{z})),$$

where $\boldsymbol{\mu}_\theta(\mathbf{z})$ and $\boldsymbol{\sigma}^2_\theta(\mathbf{z})$ are the outputs of the decoder network (hence $\theta = \{\mathbf{W}_{\text{dec}}, \mathbf{b}_{\text{dec}}\}$). Note that $\boldsymbol{\sigma}^2_\theta(\mathbf{z})$ indifferently denotes the covariance matrix of the distribution, which is assumed diagonal, or the vector of its diagonal entries.

The exact posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$ corresponding to the above model is intractable. It is approximated with a tractable parametric model $q_\phi(\mathbf{z}|\mathbf{x})$ that will play the role of the corresponding probabilistic encoder. This model generally has a form similar to the decoder:

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \tilde{\boldsymbol{\mu}}_\phi(\mathbf{x}), \tilde{\boldsymbol{\sigma}}^2_\phi(\mathbf{x})),$$

where $\tilde{\boldsymbol{\mu}}_\phi(\mathbf{x})$ and $\tilde{\boldsymbol{\sigma}}^2_\phi(\mathbf{x})$ are the outputs of the encoder ANN (the parameter set $\phi$ is composed of $\mathbf{W}_{enc}$ and $\mathbf{b}_{enc}$; $\tilde{\boldsymbol{\sigma}}^2_\phi(\mathbf{x})$ is a diagonal covariance matrix or is the vector of its diagonal entries).

Training of the VAE model, i.e. estimation of $\theta$ and $\phi$, is done by maximizing the marginal log-likelihood $\log p_\theta(\mathbf{x})$ over a large training dataset of vectors $\mathbf{x}$. It can be shown that the marginal log-likelihood can be written as [11]:

$$\log p_\theta(\mathbf{x}) = \mathrm{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})|p_\theta(\mathbf{z}|\mathbf{x})) + \mathcal{L}(\phi, \theta, \mathbf{x}),$$

where $\mathrm{D}_{\mathrm{KL}} \geq 0$ denotes the Kullback-Leibler divergence (KLD) and $\mathcal{L}(\phi, \theta, \mathbf{x})$ is the variational lower bound (VLB) given by:

$$\mathcal{L}(\phi, \theta, \mathbf{x}) = \underbrace{-\mathrm{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})|p_\theta(\mathbf{z}))}_{\text{regularization}} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{reconstruction accuracy}}. \quad (1)$$

In practice, the model is trained by maximizing $\mathcal{L}(\phi, \theta, \mathbf{x})$ over the training dataset with respect to parameters $\phi$ and $\theta$. We can see that the VLB is the sum of two terms. The first term acts as a regularizer encouraging the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ to be close to the prior $p_\theta(\mathbf{z})$. The second term represents the average reconstruction accuracy. Since the expectation w.r.t. $q_\phi(\mathbf{z}|\mathbf{x})$ is difficult to compute analytically, it is approximated using a Monte Carlo estimate and samples drawn from $q_\phi(\mathbf{z}|\mathbf{x})$. For other technical details that are not relevant here, the reader is referred to [11].

As discussed in [12] and [25], a weighting factor, denoted $\beta$, can be introduced in (1) to balance the regularization and reconstruction terms:

$$\mathcal{L}(\phi, \theta, \beta, \mathbf{x}) = -\beta\, \mathrm{D}_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})|p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})], \quad (2)$$

This enables the user to better control the trade-off between output signal quality and compactness/orthogonality of the latent coefficients $\mathbf{z}$. Indeed, if the reconstruction term is too strong relatively to the regularization term, then the distribution of the latent space will be poorly constrained by the prior $p_\theta(\mathbf{z})$, turning the VAE into an AE. Conversely, if it is too weak, then the model may focus too much on constraining the latent coefficients to follow the prior distribution while providing poor signal reconstruction [25]. In the present work we used this type of $\beta$-VAE and we present the results obtained with different values of $\beta$. These latter were selected manually after pilot experiments to ensure that the values of the regularization and the reconstruction accuracy terms in (2) are in the same range.

## 3. EXPERIMENTS

### 3.1 Dataset

In this study, we used the NSynth dataset introduced in [9]. This is a large database (more than 30 GB) of 4s long monophonic music sounds sampled at 16 kHz. They represent 1,006 different instruments generating notes with different pitches (from MIDI 21 to 108) and different velocities (5 different levels from 25 to 127). To generate these samples different methods were used: Some acoustic and electronic instruments were recorded and some others were synthesized. The dataset is labeled with: i) instrument family (e.g., keyboard, guitar, synth_lead, reed), ii) source (acoustic, electronic or synthetic), iii) instrument index within the instrument family, iv) pitch value, and v) velocity value. Some other labels qualitatively describe the samples, e.g. brightness or distortion, but they were not used in our work.

To train our models, we used a subset of 10,000 different sounds randomly chosen from this NSynth database, representing all families of instruments, different pitches and different velocities. We split this dataset into a training set (80%) and testing set (20%). During the training phase, 20% of the training set was kept for validation. In order to have a statistically robust evaluation, a $k$-fold cross-validation procedure with $k = 5$ was used to train and test all different models (we divided the dataset into 5 folds, used 4 of them for training and the remaining one for test, and repeated this procedure 5 times so that each sound of the initial dataset was used once for testing).

### 3.2 Data Pre-Processing

For magnitude and phase short-term spectra extraction, we applied a 1,024-point STFT to the input signal using a sliding Hamming window with 50% overlap. Frames corre-

sponding to silence segments were removed. The corresponding 513-point positive-frequency magnitude spectra were then converted to log-scale and normalized in energy: We fixed the maximum of each log-spectrum input vector to 0 dB (the energy coefficient was stored to be used for signal reconstruction). Then, the log-spectra were thresholded, i.e. every log-magnitude below a fixed threshold was set to the threshold value. Finally they were normalized between $-1$ and 1, which is a usual procedure for ANN inputs. Three threshold values were tested: $-80$ dB, $-90$ dB and $-100$ dB. Corresponding denormalization, log-to-linear conversion and energy equalization were applied after the decoder, before signal reconstruction with transmitted phases and inverse STFT with overlap-add.

### 3.3 Autoencoder Implementations

We tried different types of autoencoders: AE, DAE, LSTM-AE and VAE. For all the models we investigated several values for the encoding dimension, i.e. the size of the bottleneck layer / latent variable vector, from $enc = 4$ to 100 (with a fine-grained sampling for $enc \leq 16$). Different architectures were tested for the DAEs: [513, 128, $enc$, 128, 513], [513, 256, $enc$, 256, 513] and [513, 256, 128, $enc$, 128, 256, 513]. Concerning the LSTM-AE, our implementation used two vanilla forward LSTM layers (one for the encoder and one for the decoder) with non-linear activation functions giving the following architecture: [513, $enc$, 513]. Both LSTM layers were designed for many-to-many sequence learning, meaning that a sequence of inputs, i.e. of spectral magnitude vectors, is encoded into a sequence of latent vectors of same temporal size and then decoded back to a sequence of reconstructed spectral magnitude vectors. The architecture we used for the VAE was [513, 128, $enc$, 128, 513] and we tested different values of the weight factor $\beta$. For all the neural models, we tested different pairs of activation functions for the hidden layers and output layer, respectively: (tanh, linear), (sigmoid, linear) and (tanh, sigmoid).

AE, DAE, LSTM-AE and VAE models were implemented using the *Keras* toolkit [26] (we used the *scikit-learn* [27] toolkit for the PCA). Training was performed using the Adam optimizer [28] with a learning rate of $10^{-3}$ over 600 epochs with early stopping criterion (with a patience of 30 epochs) and a batch size of 512. The DAEs were trained in two different ways, with and without layer-wise training.

### 3.4 Experimental Results for Analysis-Resynthesis

Fig. 3 shows the reconstruction error (RMSE in dB) obtained with PCA, AE, DAE and LSTM-AE models on the test set (averaged over the 5 folds of the cross-validation procedure), as a function of the dimension of the latent space. The results obtained with the VAE (using the same protocol, and for different $\beta$ values) are shown in Fig. 4. For the sake of clarity, we present here only the results obtained for i) a threshold of $-100$ dB applied on the log-spectra, and ii) a restricted set of the tested AE, DAE and VAE architectures (listed in the legends of the figures). Similar trends were observed for other thresholds and other tested architectures. For each considered dimension of the latent space, a 95% confidence interval of each reconstruction error was obtained by conducting paired t-test, considering each sound (i.e. each audio file) of the test set as an independent sample.

RMSE provides a global measure of magnitude spectra reconstruction but can be insufficiently correlated to perception depending on which spectral components are correctly or poorly reconstructed. To address this classical issue in audio processing, we also calculated objective measures of perceptual audio quality, namely PEMO-Q scores [18]. The results are reported in Fig. 5 and Fig. 6.

As expected, the RMSE decreases with the dimension of the latent space for all methods. Interestingly, PCA systematically outperforms (or at worst equals) shallow AE. This somehow contradicts recent studies on image compression for which a better reconstruction is obtained with AE compared to PCA [1]. To confirm this unexpected result, we replicated our PCA vs. AE experiment on the MNIST image dataset [29], using the same AE implementation and a standard image preprocessing (i.e. vectorization of each $28 \times 28$ pixels gray-scale image into a 784-dimensional feature vector). In accordance with the literature, the best performance was systematically obtained with AE (for any considered dimension of the latent space). This difference of AE's behavior when considering audio and image data was unexpected and, to our knowledge, it has never been reported in the literature.

Then, contrary to (shallow) AE, DAEs systematically outperform PCA (and thus AE), with up to almost 20% improvement (for $enc = 12$ and $enc = 16$). Our experiments did not reveal notable benefit of layer-by-layer DAE training over end-to-end training. Importantly, for small dimensions of the latent space (e.g. smaller than 16), RMSE obtained with DAE decreases much faster than with PCA and AE. This is even more the case for LSTM-AE which shows an improvement of the reconstruction error of more than 23% over PCA (for $enc = 12$ and $enc = 16$). These results confirm the benefits of using a more complex architecture than shallow AE, here deep or recurrent, to efficiently extract high-level abstractions and compress the audio space. This is of great interest for sound synthesis for which the latent space has to be kept as low-dimensional as possible (while maintaining a good reconstruction accuracy) in order to be "controlled" by a musician.

Fig. 4 shows that the overall performance of VAEs is in between the performance of DAEs (even equals DAEs for lower encoding dimensions, say smaller than 12) and the performances of PCA and AE. Let us recall that minimizing the reconstruction accuracy is not the only goal of VAE which also aims at constraining the distribution of the latent space. As shown in Fig. 4, the parameter $\beta$, which balances regularization and reconstruction accuracy in (2), plays a major role. As expected, high $\beta$ values foster regularization at the expense of reconstruction accuracy. However, with $\beta \leqslant 2.10^{-6}$ the VAE clearly outperforms PCA, e.g. up to 20% for $enc = 12$.

It can be noticed that when the encoding dimension is high ($enc = 100$), PCA seems to outperform all the other models. Hence, in that case, the simpler (linear model)

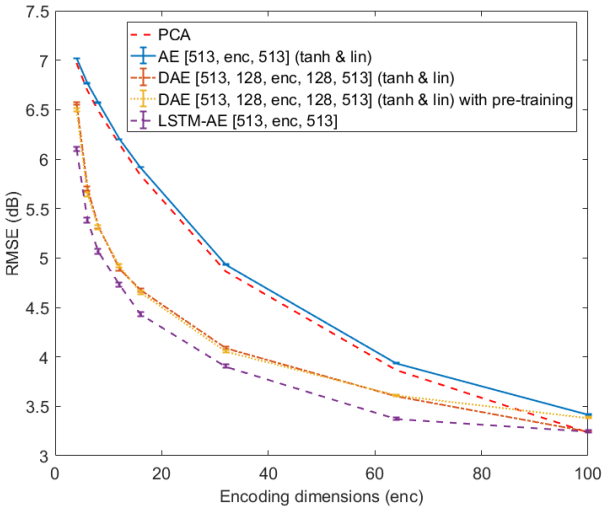



Figure 3: Reconstruction error (RMSE in dB) obtained with PCA, AE, DAE (with and without layer-wise training) and LSTM-AE, as a function of latent space dimension.

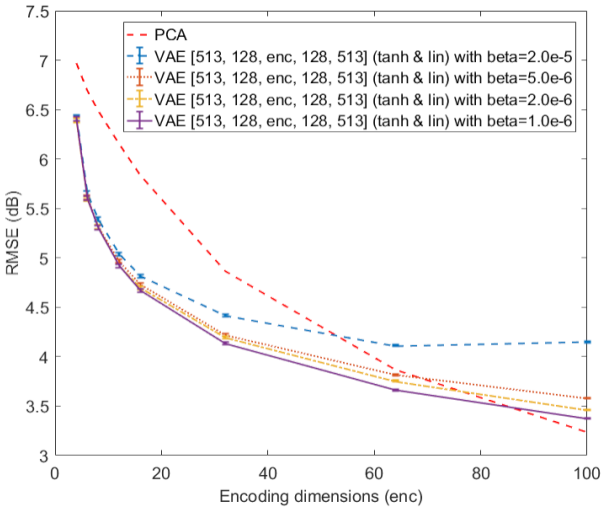



Figure 4: Reconstruction error (RMSE in dB) obtained with VAEs as a function of latent space dimension (RMSE obtained with PCA is also recalled).

seems to be the best (we can conjecture that achieving the same level of performance with autoencoders would require more training data, since the number of free parameters of these model increases drastically). However, using such high-dimensional latent space as control parameters of a music sound generator is impractical.

Similar conclusions can be drawn from Fig. 5 and Fig. 6 in terms of audio quality. Indeed, in a general manner, the PEMO-Q scores are well correlated with RMSE measures in our experiments. PEMO-Q measures for PCA and AE are very close, but PCA still slightly outperforms the shallow AE. The DAEs and the VAEs both outperform the PCA (up to about $11\%$ for $enc = 12$ and $enc = 16$) with the audio quality provided by the DAEs being a little better than for the VAEs. Surprisingly, and contrary to RMSE scores, the LSTM-AE led to a (slightly) lower PEMO-Q scores, for all considered latent dimensions. Further investigations will be done to assess the relevance of such differences at the perceptual level.

### 3.5 Decorrelation of the Latent Dimensions

Now we report further analyses aiming at investigating how the extracted latent dimensions may be used as *control* parameters by the musician. In the present sound synthesis framework, such control parameters are expected to respect (at least) the following two constraints i) to be as decorrelated as possible in order to limit the redundancy in the spectrum encoding, ii) to have a clear and easy-to-understand perceptual meaning. In the present study, we focus on the first constraint by comparing PCA, DAEs, LSTM-AE and VAEs in terms of correlation of the latent dimensions. More specifically, the absolute values of the correlation coefficient matrices of the latent vector $\mathbf{z}$ were computed on each sound from the test dataset and Fig. 7 reports the mean values averaged over all the sounds of the test dataset. For the sake of clarity, we present here these results only for a latent space of dimension 16 for one model of DAE ([513, 128, 16, 128, 513] (tanh & lin) with end-to-end training) and for VAEs with the same architecture ([513, 128, 16, 128, 513] (tanh & lin)) and different values of $\beta$ (from $1.10^{-6}$ to $2.10^{-5}$).

As could be expected from the complexity of its structure, we can see that the LSTM-AE extracts a latent space where the dimensions are significantly correlated with each other. Such additional correlations may come from the sound dynamics which provide redundancy in the prediction. We can also see that PCA and VAEs present similar behaviors with much less correlation of the latent dimensions, which is an implicit property of these models. Interestingly, and in accordance with (2), we can notice that the higher the $\beta$, the more regularized the VAE and hence the more decorrelated the latent dimensions. Importantly, Fig. 7 clearly shows that for a well-chosen $\beta$ value, the VAE can both extract latent dimensions that are much less correlated than for corresponding DAEs, which makes it a better candidate for extracting good control parameters, while allowing fair to good reconstruction accuracy (see Fig. 4). The $\beta$ value has thus to be chosen wisely in order to find the optimal trade-off between decorrelation of the latent dimensions and reconstruction accuracy.

### 3.6 Examples of Sound Interpolation

As a first step towards the practical use of the extracted latent space for navigating through the sound space and creating new sounds, we illustrate how it can be used to interpolate between sounds, in the spirit of what was done for instrument hybridization in [9]. We selected a series of pairs of sounds from the NSynth dataset with the two sounds in a pair having different characteristics. For each pair, we proceeded to separate encoding, entry-wise linear interpolation of the two resulting latent vectors, decoding, and finally individual signal reconstruction with inverse STFT and the Griffin and Lim algorithm to reconstruct the phase spectrogram [19]. We experimented dif-

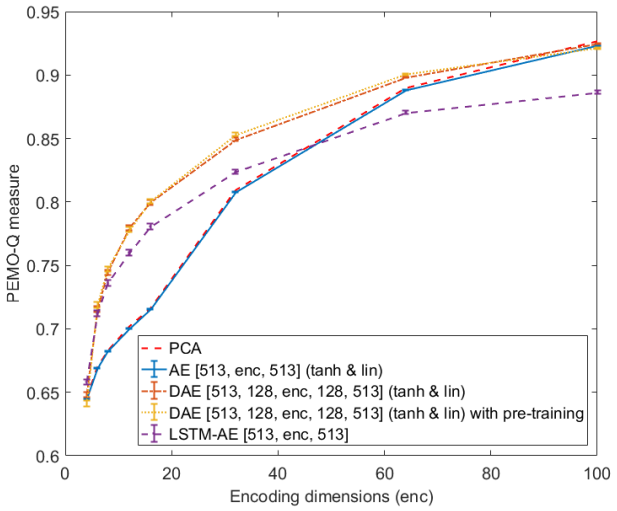



Figure 5: PEMO-Q measures obtained with PCA, AE, DAEs (with and without layer-wise training) and LSTM-AE, as a function of latent space dimension.

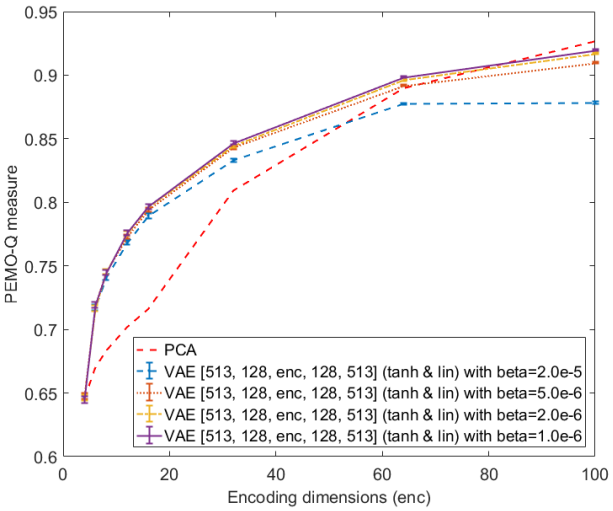



Figure 6: PEMO-Q measures obtained with VAEs as a function of latent space dimension (measures obtained with PCA are also recalled).

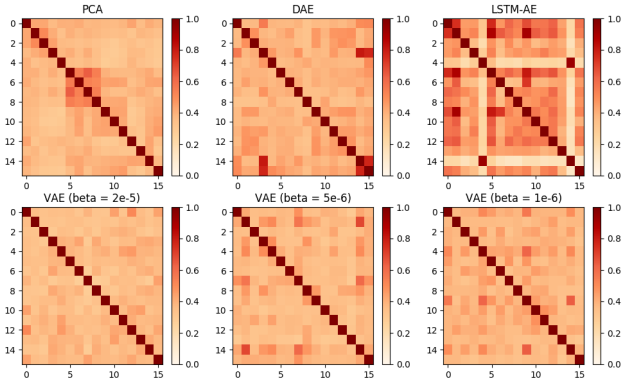



Figure 7: Correlation matrices of the latent dimensions (average absolute correlation coefficients) for PCA, DAE, LSTM-AE and VAEs.

ferent degrees of interpolation between the two sounds: $\hat{\mathbf{z}} = \alpha\ \mathbf{z}_1 + (1-\alpha)\ \mathbf{z}_2$, with $\mathbf{z}_i$ the latent vector of sound $i$, $\hat{\mathbf{z}}$ the new interpolated latent vector, and $\alpha \in [0, 0.25, 0.5, 0.75, 1]$ (this interpolation is processed independently on each pair of vectors of the time sequence). The same process was applied using the different AE models we introduced earlier.

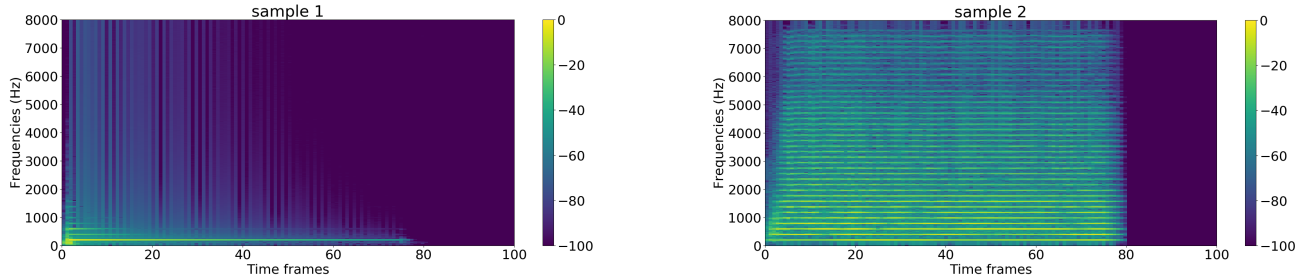
Fig. 8 displays one example of results obtained with PCA, with the LSTM-AE and with the VAE (with $\beta = 1.10^{-6}$), with an encoding dimension of 32. Qualitatively, we note that interpolations in the latent space lead to a smooth transition between source and target sound. By increasing sequentially the degree of interpolation, we can clearly go from one sound to another in a consistent manner, and create interesting hybrid sounds. The results obtained using PCA interpolation are (again qualitatively) below the quality of the other models. The example spectrogram obtained with interpolated PCA coefficients is blurrier around the harmonics and some audible artifacts appear. On the opposite, the LSTM-AE seems to outperform the other models by better preserving the note attacks (see comparison with VAE in Fig. 8). More interpolation examples along with corresponding audio samples can be found at `https://goo.gl/Tvvb9e`.

## 4. CONCLUSIONS AND PERSPECTIVES

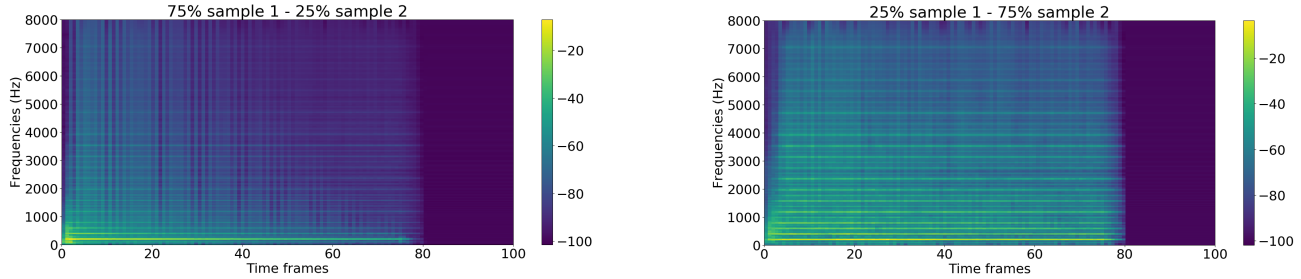
In this study, we investigated dimensionality reduction based on autoencoders to extract latent dimensions from a large music sound dataset. Our goal is to provide a musician with a new way to generate sound textures by exploring a low-dimensional space. From the experiments conducted on a subset of the publicly available database NSynth, we can draw the following conclusions: i) Contrary to the literature on image processing, shallow autoencoders (AEs) do not here outperform principal component analysis (in terms of reconstruction accuracy); ii) The best performance in terms of signal reconstruction is always obtained with deep or recurrent autoencoders (DAEs or LSTM-AE); iii) Variational autoencoders (VAEs) lead to a fair-to-good reconstruction accuracy while constraining the statistical properties of the latent space, ensuring some amount of decorrelation across latent coefficients and limiting their range. These latter properties make the VAEs good candidates for our targeted sound synthesis application.

In line with the last conclusion, future works will mainly focus on VAEs. First, we will investigate recurrent architecture for VAE such as the one proposed in [30]. Such approach may lead to latent dimensions encoding separately the sound texture and its dynamics, which may be of potential interest for the musician.

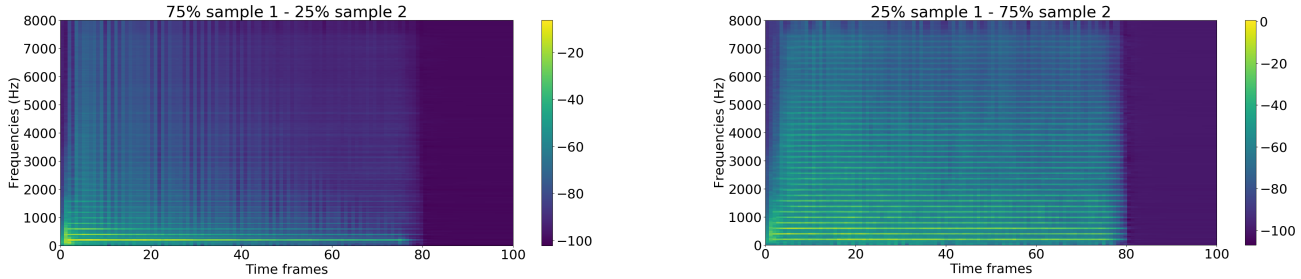
Then, we will address the crucial question of the perceptual meaning/relevance of the latent dimensions. Indeed using a non-informative prior distribution of $\mathbf{z}$ such as a standard normal distribution does not ensure that each dimension of $\mathbf{z}$ represents an interesting perceptual dimension of the sound space, although this is a desirable objec-



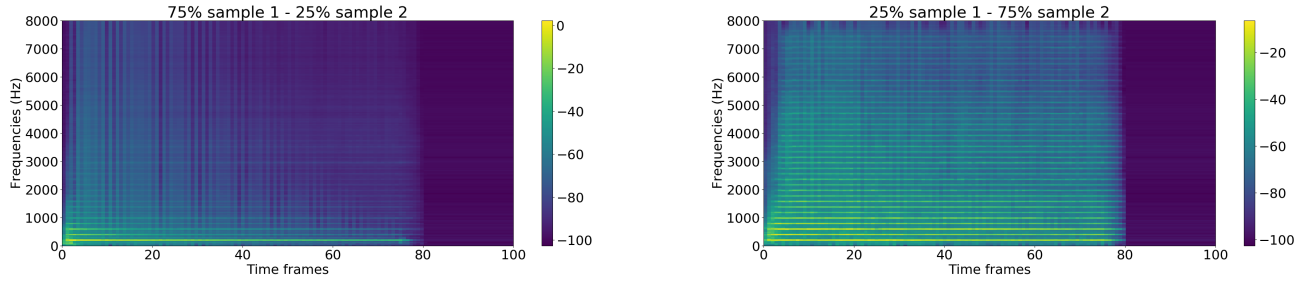

(a) Original samples - Left : bass_electronic_010-055-100, Right : brass_acoustic_050-055-100




(b) PCA




(c) LSTM-AE




(d) VAE

Figure 8: Examples of decoded magnitude spectrograms after sound interpolation of 2 samples (top) in the latent space using respectively PCA (2nd row), LSTM-AE (3rd row) and VAE (bottom). A more detailed version of the figure can be found at `https://goo.gl/Tvvb9e`.

tive. In [14], the authors recently proposed a first solution to this issue in the context of a restricted set of acoustic instruments. They introduced in the variational lower bound (2) of the VAE loss an additional regularization term encouraging the latent space to respect the structure of the instrument timbre. In the same spirit, our future works will investigate different strategies to model the complex relationships between sound textures and their perception, and introduce these models at the VAE latent space level.

## 5. ACKNOWLEDGMENT

The authors would like to thank Simon Leglaive for our fruitful discussions. This work was supported by ANRT in the framework of the PhD program CIFRE 2016/0942.

## 6. REFERENCES

[1] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Process. Systems*, Montreal, Canada, 2014.

[3] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," in *Proc. of the Int. Conf. on Machine Learning*, New York, NY, 2016.

[4] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. Courville, and Y. Bengio, "SampleRNN: An unconditional end-to-end neural audio generation model." in *Int. Conf. on Learning Representations*, 2017.

[5] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proc. of the IEEE conf. on computer vision and pattern recognition*, 2018.

[6] E. Miranda, *Computer Sound Design: Synthesis Techniques and Programming*, ser. Music Technology series. Focal Press, 2002.

[7] A. Sarroff and M. Casey, "Musical audio synthesis using autoencoding neural nets," in *Joint Int. Computer Music Conf. and Sound and Music Computing Conf.*, Athens, Greece, 2014.

[8] J. Colonel, C. Curro, and S. Keene, "Improving neural net auto encoders for music synthesis," in *Audio Engineering Society Convention*, New-York, NY, 2017.

[9] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, "Neural audio synthesis of musical notes with wavenet autoencoders," *arXiv preprint arXiv:1704.01279*, 2017.

[10] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[11] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[12] M. Blaauw and J. Bonada, "Modeling and transforming speech using variational autoencoders." in *Conf. of the Int. Speech Comm. Association (Interspeech)*, San Francisco, CA, 2016.

[13] W.-N. Hsu, Y. Zhang, and J. Glass, "Learning latent representations for speech generation and transformation," *arXiv preprint arXiv:1704.04222*, 2017.

[14] P. Esling, A. Chemla-Romeu-Santos, and A. Bitton, "Generative timbre spaces with variational audio synthesis," in *Proc. of the Int. Conf. on Digital Audio Effects 2018*, Aveiro, Portugal, 2018.

[15] S. Leglaive, L. Girin, and R. Horaud, "A variance modeling framework based on variational autoencoders for speech enhancement," in *IEEE Int. Workshop on Machine Learning for Signal Process.*, Aalborg, Denmark, 2018.

[16] ——, "Semi-supervised multichannel speech enhancement with variational autoencoders and non-negative matrix factorization," in *IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, Brighton, UK, 2019.

[17] L. Li, H. Kameoka, and S. Makino, "Fast MVAE: Joint separation and classification of mixed sources based on multichannel variational autoencoder with auxiliary classifier," in *IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, Brighton, UK, 2019.

[18] R. Huber and B. Kollmeier, "PEMO-Q: A new method for objective audio quality assessment using a model of auditory perception," *IEEE Transactions on Audio, Speech, and Language Process.*, vol. 14, no. 6, pp. 1902–1911, 2006.

[19] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Process.*, vol. 32, no. 2, pp. 236–243, 1984.

[20] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[22] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Process. Systems*, Vancouver, Canada, 2007.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[25] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "$\beta$-VAE: learning basic visual concepts with a constrained variational framework." in *Int. Conf. on Learning Representations*, 2017.

[26] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] Y. LeCun, C. Cortes, and C. Burges, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist/, 1998.

[30] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in Neural Information Process. Systems*, 2015.