Unsupervised Learning of a Disentangled Speech Representation for Voice Conversion

Tobias Gburrek[1], *Thomas Glarner*[1], *Janek Ebbers*[1], *Reinhold Haeb-Umbach*[1], *Petra Wagner*[2]

[1]Paderborn University, Department of Communications Engineering, Paderborn, Germany
[2]Bielefeld University, Bielefeld, Germany

{gburrek,glarner,ebbers,haeb}@nt.upb.de
petra.wagner@uni-bielefeld.de

Abstract

This paper presents an approach to voice conversion, which does neither require parallel data nor speaker or phone labels for training. It can convert between speakers which are not in the training set by employing the previously proposed concept of a factorized hierarchical variational autoencoder. Here, linguistic and speaker induced variations are separated upon the notion that content induced variations change at a much shorter time scale, i.e., at the segment level, than speaker induced variations, which vary at the longer utterance level. In this contribution we propose to employ convolutional instead of recurrent network layers in the encoder and decoder blocks, which is shown to achieve better phone recognition accuracy on the latent segment variables at frame-level due to their better temporal resolution. For voice conversion the mean of the utterance variables is replaced with the respective estimated mean of the target speaker. The resulting log-mel spectra of the decoder output are used as local conditions of a WaveNet which is utilized for synthesis of the speech waveforms. Experiments show both good disentanglement properties of the latent space variables, and good voice conversion performance.

Index Terms: voice conversion, direct waveform modeling, disentanglement

1. Introduction

Voice conversion (VC) is concerned with modifying the speech waveform to convert non- or paralinguistic traits while preserving the linguistic contents. Such a technology can enable people with vocal disorders to use speech interfaces to devices. Other applications are in the field of computer games to give computer animated characters (avatars) a desired set of extralinguistic traits.

VC is typically carried out in three stages: Analysis, conversion, and synthesis. In the analysis stage, an appropriate parametric representation of the input speech is computed, e.g., log-mel spectra or a parametric representation based on the source-filter model of speech. Those parameters are then converted to the target signal by employing a non-linear mapping function, while in the synthesis stage the speech waveform is generated. Recently, neural network-based techniques, such as the WaveNet [1], have achieved remarkable naturalness in waveform generation [2].

The key component is arguably the converter. This conversion task can be formulated as a regression problem. In supervised training the mapping is learnt from utterance pairs of source and target speech. In the case of speaker conversion considered here, such parallel data consist of utterances of the source and the target speaker, where both speak the same text. Many approaches have been proposed to learn the mapping, such as vector quantization based [3], Gaussian mixture model based [4], exemplar-based [5], or deep learning based [6] methods.

Nonetheless, the need for parallel data is clearly a nuisance. It would be much less restrictive if any speech probes of a target speaker were sufficient to learn the mapping instead of requiring the target speaker to utter exactly the same text as the source speaker. However, unsupervised voice conversion, which does not require parallel data, is much more difficult.

There are basically two approaches to non-parallel data voice conversion. The first is to generate pseudo parallel data from non-parallel data through frame alignment, e.g., [7], and the second is to factorize the speech representation into the traits to be converted and the rest, e.g., [8]. Current techniques achieve this by means of generative neural models, such as the concept of GANs [9, 10] or variational autoencoders [11, 12, 13].

This contribution is concerned with the latter. A disentangled representation of the input speech signal is developed, where speaker characteristics are captured in one set of latent parameters, and content related variations in another. The starting point of our research is the factorized hierarchical variational autoencoder (FHVAE), which was developed in [14]. In this model two key assumptions are made. First, the sources of variation can be much better disentangled in a nonlinear low-dimensional latent space than in the observed data. This leads to the use of a variational autoencoder (VAE). The second key assumption is that content induced properties of the speech signal vary at a much faster rate than those induced by the speaker or other extra-linguistic factors. This assumption is represented by a corresponding probabilistic graphical model in the latent space, where a series of so-called *segment variables* captures short-term variations, supposedly caused by the linguistic content, and a series of *utterance variables*, which captures variations at a larger time-scale, supposedly caused by the speaker or environment characteristics present in the speech signal [14].

In this contribution we extend that work in several important ways: First, we propose a convolutional neural network (CNN) based VAE encoder/decoder architecture, which, compared to the recurrent neural network (RNN) and fully connected network (FCN) used in [14], allows to model short-term variations at a more fine-grained level of detail. This will be demonstrated by better phoneme recognition accuracy achieved on the segment variables. Second, we conduct waveform generation with the WaveNet, using the converted log-mel spectral features as local conditions, which results in much better speech quality compared to the Griffin-Lim reconstruction done in [14]. Compared to the voice conversion proposed in [13, 11], which also employs a WaveNet for synthesis, our WaveNet

10.21437/SSW.2019-15

component is trained independently of the target speaker. This allows to train the WaveNet without knowledge of the target speaker, for which it is later supposed to generate speech.

The paper is organized as follows: In Section 2, after briefly reviewing the foundations of VAEs, we give a summary of the FHVAE introduced in [14]. Furthermore, a brief explanation of the WaveNet architecture is given. In Section 3, we present our proposed model, which is based on the FHVAE, and in Section 4, we introduce the experimental setup and discuss the obtained results.

2. Theoretical Background

2.1. Variational Autoencoder

This section discusses the VAE, which forms the basis of the FHVAE, and in consequence, our proposed voice conversion system. The VAE framework assumes that a complicated observation distribution can be modeled by learning a nonlinear mapping from a latent representation $\mathbf{z}$, called the code, to the parameters of a normal distribution $p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; f(\mathbf{z};\delta), \sigma^2)$, which serves as the observation likelihood. This mapping $f(\mathbf{z};\delta)$ is given by a neural network (NN), called the decoder network, with learnable network parameters δ. In contrast to the default model of a Gaussian VAE, the variance of the decoder distribution is held fixed in the models appearing here. This is further discussed later in this Section.

In the case of a basic Gaussian VAE as proposed in [15], a standard normal distribution $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ is used for the code prior.

The inherently nonlinear relationship between the latent code and the parameters leads to an analytically intractable posterior and thus prevents exact inference. Therefore, variational inference (VI) is used to learn a variational posterior

$$q(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_{\mathbf{z}}, \operatorname{diag} \boldsymbol{\sigma}^2_{\mathbf{z}})$$

by maximizing a lower bound $\mathcal{L}$ on the evidence marginal, $p(\mathbf{x})$, called evidence lower bound (ELBO) [16]. If the observations $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are assumed to be independent and identically distributed (i.i.d.) and if each observation $\mathbf{x}_n$ has a corresponding latent representation $\mathbf{z}_n$, the ELBO decomposes into a sum $\mathcal{L} = \sum_{n=1}^{N} \mathcal{L}_n$ of independent terms with

$$\mathcal{L}_n = \mathbb{E}_{q(\mathbf{z}_n)}\left[\frac{p(\mathbf{x}_n|\mathbf{z}_n)p(\mathbf{z}_n)}{q(\mathbf{z}_n)}\right].$$

The posterior parameters $\boldsymbol{\mu}_{\mathbf{z}_n}, \log\big(\boldsymbol{\sigma}^2_{\mathbf{z}_n}\big) = g(\mathbf{x}_n;\phi)$ are modeled by another NN with learnable parameters ϕ, which is commonly called the encoder network. Note that the covariance matrix is usually chosen to be diagonal for simplicity.

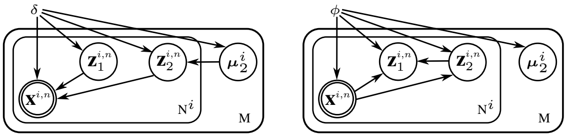
This leads to the following cost function:

$$J_n^{(\mathrm{VAE})} = \frac{\mathbb{E}_{q(\mathbf{z}_n;\phi)}\, \|\mathbf{x}_n - f(\mathbf{z}_n;\delta)\|^2}{2\sigma^2} + \mathrm{KL}(q(\mathbf{z}_n;\phi)\|p(\mathbf{z}_n)),$$

where the sign is flipped because the training of NNs is usually stated as a minimization problem. The loss decomposes into the sum of two terms which are commonly called the reconstruction loss and the Kullback-Leibler (KL) loss. Ideally, a joint minimization of both loss terms leads to a solution that is able to produce a reasonable reconstruction of the observation space while reducing the amount of information in the code at the same time.

With the shortcut $\beta = 2\sigma^2$, the model becomes an instance of the β-VAE as proposed in [17]. Furthermore, the constant variance lowers the expressiveness of the decoder which leads to a reduction of the problem of posterior collapse [18].

2.2. Factorized Hierarchical Variational Autoencoder



(a) *generative model* (b) *inference model*

Figure 1: *Graphical model of an FHVAE*

The task of voice conversion requires a disentanglement of speaker information and linguistic content, which is not provided by the vanilla VAE. A sensible way to introduce a structure capable of this disentanglement, called the FHVAE, was proposed in [14]. An utterance is represented as a sequence of frame-level input feature vectors. For the FHVAE, this sequence is divided into non-overlapping segments which can span multiple frames. The FHVAE is designed up-front to include disentanglement by exploiting that utterances show variations on different time scales: The speaker and environmental characteristics show only slow changes and can thus be assumed as constant for a single utterance. On the other hand, the linguistic content leads to variations on short time scales, i.e., the segment level. This is exploited by factorizing the VAE latent code vector as a concatenation $\mathbf{z}^{i,n} = [\mathbf{z}_1^{i,n}; \mathbf{z}_2^{i,n}]$ of two kinds of latent vectors:

- A segment variable, called $\mathbf{z}_1^{i,n}$, with a standard normal prior
- An utterance variable, called $\mathbf{z}_2^{i,n}$, which is assumed to follow a normal distribution with a mean that is constant for the whole utterance.

The corresponding graphical model of the generative process and the associated inference model are shown in Figure 1. Here, M denotes the number of utterances, and N^i the number of segments within an utterance. In our work, the mean $\boldsymbol{\mu}_2^i$ of the utterance variable, called the s-vector in [14], models the speaker characteristics.

Equivalently to the VAE, VI is used to learn the posteriors of the latent variables. Therefore, the variational lower bound is maximized by drawing batches of segments, whereby an additional discriminative objective $\log p(i|\mathbf{z}_2^{i,n})$ was introduced in [14] to encourage different s-vectors for different utterances:

$$\begin{aligned}\mathcal{L}_{i,n} =& \mathbb{E}_{q(\mathbf{z}_1^{i,n}, \mathbf{z}_2^{i,n}|\mathbf{x}^{i,n};\phi)}[\log p(\mathbf{x}^{i,n}|\mathbf{z}_1^{i,n}, \mathbf{z}_2^{i,n};\delta)] \\ &- \mathbb{E}_{q(\mathbf{z}_2^{i,n}|\mathbf{x}^{i,n};\phi)}[\mathrm{KL}(q(\mathbf{z}_1^{i,n}|\mathbf{x}^{i,n}, \mathbf{z}_2^{i,n};\phi)||p(\mathbf{z}_1^{i,n}))] \\ &- \mathrm{KL}(q(\mathbf{z}_2^{i,n}|\mathbf{x}^{i,n};\phi)||p(\mathbf{z}_2^{i,n}|\tilde{\boldsymbol{\mu}}_2^i)) \\ &+ \frac{1}{N^i}\log p(\tilde{\boldsymbol{\mu}}_2^i) + \alpha \log p(i|\mathbf{z}_2^{i,n}) + \mathit{const},\end{aligned}$$

where i denotes the index of the utterance, n the index of the segment, $\mathbf{x}^{i,n}$ the feature vectors within one segment, N^i the segment length and $\tilde{\boldsymbol{\mu}}_2^i$ the i-th entry of a lookup table for the s-vectors. In [14] it was proposed to realize the FHVAE by using long short-term memory (LSTM) cells for the encoders and the decoder, whereby the FHVAE consumes one segment and encodes this into one latent segment variable and one latent utterance variable. The decoder uses this pair of latent variables to reconstruct the corresponding segment.

2.3. WaveNet

The FHVAE decoder outputs log-mel spectra. To synthesize the speech waveform from those, we make use of the WaveNet model introduced in [1].

The WaveNet is a generative model which allows the synthesis of audio on the level of time domain samples. It implements a nonlinear autoregressive process of order T with a stack of one-dimensional causal CNN layers. Modeling the conditional distribution of the next sample given all previous T samples $p(x_t|x_{t-1}, \dots, x_{t-T})$ allows a sample-wise generation of new samples by feeding back previously generated samples. To increase the length T of the receptive field, i.e. the number of previous samples affecting the next output sample, while keeping the number of layers low, dilated convolutional layers are used, where only every 2^n-th sample is used as input for the n-th layer. This leads to a scheme where every sample of the receptive field affects the output through only a single path. Typically, several blocks of stacked layers are used, whereby the dilation scheme is reset after every block.

The conditional distribution is modeled as a categorical distribution of 256 values. This requires the samples to be discretized first. The authors propose to apply μ-law companding to deal with the nonuniform distribution of speech amplitudes.

To allow a guided synthesis of a desired waveform through a given set of additional inputs $\mathbf{h}$, a distribution $p(x_t|x_{t-1}, \dots, x_{t-T}, \mathbf{h})$ conditioned on these values $\mathbf{h}$, which can consist of local and global conditions, needs to be modeled. The conditional WaveNet architecture often utilizes upsampling because the frame rate of the local conditions most likely differs from the sampling rate.

3. Proposed Model

3.1. Convolutional Factorized Hierarchical Variational Autoencoder

The long length of the segments, where one segment consists of multiple frames, is a major drawback of the realization of an FHVAE as proposed in [14]. Thus, the resolution of the learned representation of the linguistic content is closer to the syllable time scale than the phone time scale. In order to provide a good phonetic representation at fine time scale, e.g. at frame level, it is necessary to use segments with a length of only a few frames. However, preliminary experiments have shown that this results in a degraded disentanglement for the LSTM-based architecture.

To overcome this restriction we propose to realize the FHVAE with convolutional instead of recurrent network layers, as shown in in Figure 2.[1] Therefore, the length of one segment corresponds to the length of the receptive field of the resulting CNN, which covers multiple frames. Nevertheless the stride of the receptive field equals only a few frames and causes a better temporal resolution of the latent variables.

We propose a CNN as basis for both the decoder and the encoders. The structure of the CNNs is chosen to be equal to the structure of the encoder proposed in [13], with both encoders being identical. As in [13], we use strided convolutions in the CNNs of the encoders, which causes the frequency at which the latent variables are extracted to be smaller than the temporal resolution of the log-mel spectra. In consequence the decoder has to do an upsampling in order to be able to reconstruct the log-mel spectra. Thus, all strided convolutions that are used in the encoders are replaced by transposed convolutions with the same stride in the decoder.

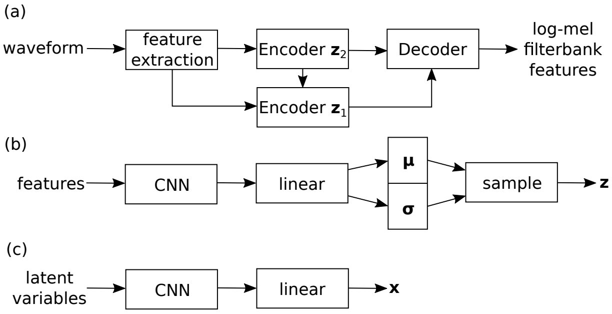


Figure 2: *Convolutional FHVAE: (a) full architecture (b) proposed architecture of the encoder of* $\mathbf{z}_1$ *and the encoder of* $\mathbf{z}_2$ *(c) proposed architecture of the decoder*

Another effect of the convolutional layers used in the proposed realization of an FHVAE is that there is no one-to-one mapping of the segments and the latent variables anymore. Although both encoders generate one latent variable for each segment, the decoder does not map from one pair of latent segment and utterance variables back to the corresponding segment. Instead of this the decoder takes a series of latent variables to reconstruct single frames of the log-mel spectra.

Due to this fact, the conditional probabilities of the generative model and the inference model have to be adjusted compared to the original FHVAE. Each variable is now conditioned on a series of corresponding variables $\mathbf{X}^i$, $\mathbf{Z}_1^i$ and $\mathbf{Z}_2^i$, respectively, instead of conditioning them on the variables which represent the corresponding segment. The particular series results from the variables lying in the receptive field of the CNN. Furthermore, this also causes different time scales for the different parts of the lower bound. There is one time scale for the frames of the log-mel spectra and a second coarser time scale for the latent variables. Consequently, it is not possible to draw a batch of segments and calculate the variational lower bound as proposed in [14]. Rather, it is necessary to draw a batch of series of segments and calculate the lower bound from that:

$$\begin{aligned}\mathcal{L}_i = &\sum_{t=1}^{T^i} \mathbb{E}_{q(\mathbf{z}_1^{i,n}, \mathbf{z}_2^{i,n}|\mathbf{X}^i;\phi)}[\log p(\mathbf{x}^{i,t}|\mathbf{Z}_1^i, \mathbf{Z}_2^i; \delta)] \\ &- \sum_{n=1}^{\tilde{N}^i} \mathbb{E}_{q(\mathbf{z}_2^{i,n}|\mathbf{X}^i;\phi)}[\mathrm{KL}(q(\mathbf{z}_1^{i,n}|\mathbf{X}^i, \mathbf{z}_2^{i,n}; \phi)||p(\mathbf{z}_1^{i,n}))] \\ &+ \sum_{n=1}^{\tilde{N}^i} \mathrm{KL}(q(\mathbf{z}_2^{i,n}|\mathbf{X}^i; \phi)||p(\mathbf{z}_2^{i,n}|\tilde{\boldsymbol{\mu}}_2^i)) \\ &+ \log p(\tilde{\boldsymbol{\mu}}_2^i) + \sum_{n=1}^{\tilde{N}^i} \alpha \log p(i|\mathbf{z}_2^{i,n}) + const,\end{aligned}$$

whereby $\tilde{N}^i$ denotes the number of considered segments and T^i the number of frames of the considered series of segments.

[1] The feature extraction has to be seen as fixed signal processing layer in both encoders to keep the point of view of an autoencoder.

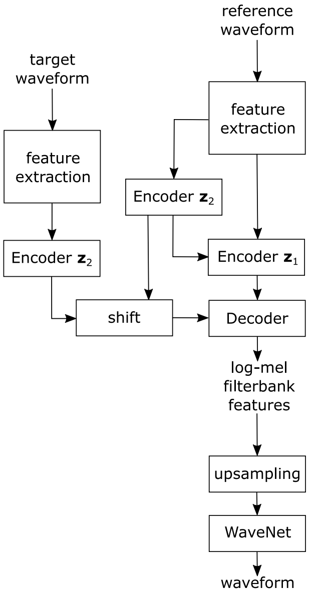


Figure 3: *Block diagram of the proposed voice conversion system*

3.2. Voice Conversion System

The proposed voice conversion system is shown in Figure 3. This model combines the modified FHVAE with a WaveNet used for speech synthesis. First, 80 dimensional log-mel spectra are computed from the input speech signal. Those plus their first and second order derivatives are then encoded into the latent segment and utterance variables $\mathbf{z}_1$ and $\mathbf{z}_2$, respectively. To do speaker conversion, the utterance variables of the source speaker $\mathbf{z}_2^{(src)}$ are shifted in the direction of the utterance variables of the target speaker. This is done in the same way as it was proposed for the FHVAE in [14]. Firstly, the s-vectors of the source speaker $\boldsymbol{\mu}_2^{(src)}$ and the target speaker $\boldsymbol{\mu}_2^{(tar)}$ are estimated from the corresponding utterance. These are used to get the latent utterance variables $\mathbf{z}_2^{(tar)}$, which are used in the decoder:

$$\mathbf{z}_2^{(\mathrm{tar})} = \mathbf{z}_2^{(\mathrm{src})} - \boldsymbol{\mu}_2^{(\mathrm{src})} + \boldsymbol{\mu}_2^{(\mathrm{tar})}.$$

After this the decoder (re)constructs the log-mel spectra, and, after upsampling, the WaveNet, conditioned on the log-mel spectra, synthesizes the speech waveform.

Due to the fact that the WaveNet is a powerful autoregressive generative model the occurrence of latent space collapse is very probable for joint training of the latent space model and the WaveNet, where the constrained latent space variables are completely ignored in the speech reconstruction, as has been discussed in [13]. Similarly here, the strict constraints for the latent variables of the FHVAE can cause the latent space collapse, if those were used directly as local conditions of the WaveNet and both models were trained jointly. In this case the WaveNet would use only the past speech samples for the generation of a waveform ignoring the latent variables, whereby the model cannot conduct voice conversion.

The approach proposed here solves this issue by training the FHVAE and the WaveNet separately. First, the WaveNet is removed from the model and the FHVAE is trained on log-mel spectra. In the second step the WaveNet is trained while the FHVAE is kept fixed, using the upsampled log-mel spectra calculated from the raw waveform as local conditions. For the upsampling of the (reconstructed) log-mel spectra we employ a transposed convolutional layer.

4. Experiments

4.1. Dataset

For evaluation we consider the TIMIT dataset, which contains recordings of 630 speakers, each reading ten phonetically rich sentences [19]. The FHVAE and the WaveNet are trained on the respective training set. In order to show the ability of our system to work with speakers, which were not seen during training, we evaluate our system on the corresponding test set, which consists of new speakers.

4.2. Model Configuration

If not otherwise stated, we use the following configuration for our model. We make use of 80 dimensional log-mel spectra as features for our FHVAE. These are extracted every $10\,\mathrm{ms}$ using a $25\,\mathrm{ms}$ long window. Additionally we extend them by their first and second derivative.

As already mentioned above, we use the architecture of the encoder proposed in [13] for the CNNs in the encoders and the decoder of our FHVAE. So each CNN consists of 9 layers, whereby each layer utilizes 256 units and rectified linear unit (ReLU) activation. First, there are 2 convolutional layers with filter size 3 and stride 1 followed by a convolutional layer with stride 2 and filter size 3 and another 2 convolutional layers with filter size 3 and stride 1. In the decoder we replace the strided convolutional layer by a transposed convolution with the same stride. This block of convolutional layers is followed by 4 feedforward layers. Furthermore, residual connections are used for those layers, which do not change the temporal resolution. Finally, there is a linear layer, whereby the number of units of these layers equals the dimension of the corresponding output vector. With regard to the best results obtained in [14] the dimension of $\mathbf{z}_1$ and $\mathbf{z}_2$ is chosen to be 32 and the discriminative objective is weighted with $\alpha = 10$.

The WaveNet architecture is selected to incorporate 2 blocks of 8 dilated convolutional layers with 64 units. In the layers of the postprocessing network of the WaveNet 256 units are utilized.

4.3. Quantitative Evaluation of Disentanglement

The ability of our proposed version of an FHVAE to disentangle the linguistic content and the information about the speaker is measured by carrying out a phone recognition and a speaker recognition task on either of the two latent variables. Thereby, the phone recognition is done frame-wise to prove the better temporal resolution of the information contained in the latent variables of our proposed version of an FHVAE. The results obtained for our version of the FHVAE are compared to the results obtained for the FHVAE proposed in [14]. Additionally, we investigate different values for the number of frames per embedding of the FHVAE.

For speaker recognition, the test set is split into one set containing the sx-utterances and another set containing the si- and sa-utterances, whereby the set of sx-utterances is used for training. Phone recognition should be done in a speaker independent way. Therefore, the test set is split into a training set for the classifier, which contains 80 $\%$ of the speakers, and an evaluation set, which contains the remaining speakers.

It is desirable that the latent utterance variables $\mathbf{z}_2$ capture information about the speaker, and as little information about the linguistic content as possible, while for the segment variables $\mathbf{z}_1$ it should be just the opposite. To get the features for the speaker recognition task we separately encode each utterance into a sequence of latent segment variables and a sequence of latent utterance variables. Then we take the average over the whole series of latent variables for each utterance and use the resulting values as features for speaker recognition. For phone recognition the series of latent variables are used directly.

The speaker recognition is done by a feedforward network with 2048 units. Due to the fact, that the temporal resolution of the considered versions of the FHVAE are quite different, we use the classifier architectures similar to the corresponding decoder architectures (with adjusted output layers) for the phone recognition in order to be able to compare both versions in a fair way.

Table 1: *Speaker recognition accuracy and frame-wise phone recognition accuracy on the latent variables of the different FHVAE-architectures on TIMIT*

Architecture	**Features**	**Accuracy**	
		Speaker	**Phone**
-	log-mel spectra	93.51 %	52.17 %
CNN	$\mathbf{z}_1$	11.95 %	61.54 %
LSTM	$\mathbf{z}_1$	8.05 %	42.31 %
CNN	$\mathbf{z}_2$	93.37 %	26.44 %
LSTM	$\mathbf{z}_2$	95.86 %	40.60 %

The corresponding results for the disentanglement obtained on TIMIT are shown in Table 1. It can be seen that the utterance variables $\mathbf{z}_2$ by far outperform the segment variables $\mathbf{z}_1$ in terms of accuracy for speaker recognition, confirming that speaker information is gathered in $\mathbf{z}_2$. This holds for both the proposed convolutional FHVAE and the FHVAE using LSTM layers proposed in [14], whereby the results for the FHVAE proposed in [14] are slightly better.

For frame-wise phone recognition the proposed FHVAE-architecture outperforms the architecture proposed in [14]. Nevertheless, a quite high portion of linguistic content is still encoded into the latent utterance variables $\mathbf{z}_2$. This can be explained by the fact that also the information about the style and therefore a small portion of the linguistic content is encoded into the utterance variables so that in particular silence can be recognized from those.

We define the embedding rate as T/M, where T denotes the number of frames per utterance. This rate can be viewed as the downsampling factor between the frame- and segment-level representations. Table 2 shows the effect of the embedding rate on the proposed realization of an FHVAE and the FHVAE proposed in [14]. As mentioned above, we found that the performance of the FHVAE in terms of disentanglement strongly depends on this key measure: For the same embedding rate, our model outperforms the version proposed in [14]. It becomes clear that for a small embedding rate, resulting in very short segments, the realization proposed in [14] is prone to encode nearly all information, even the information about the linguistic content, into the latent utterance variables. Furthermore, the proposed realization suffers from either very high or very low embedding rates. Consequently, there is a trade-off between a good embedding rate and a good disentanglement.

Table 2: *Quality of disentanglement for varying embedding rates of the latent variables on TIMIT*

Architecture	T/M	**Features**	**Accuracy**	
			Speaker	**Phone**
CNN	1	$\mathbf{z}_1$	14.08 %	56.11 %
CNN	2	$\mathbf{z}_1$	11.95 %	61.54 %
CNN	8	$\mathbf{z}_1$	1.07 %	48.06 %
LSTM	8	$\mathbf{z}_1$	8.52 %	45.08 %
LSTM	20	$\mathbf{z}_1$	8.05 %	42.31 %
CNN	1	$\mathbf{z}_2$	89.82 %	28.84 %
CNN	2	$\mathbf{z}_2$	93.37 %	26.44 %
CNN	8	$\mathbf{z}_2$	92.90 %	39.06 %
LSTM	8	$\mathbf{z}_2$	96.56 %	40.42 %
LSTM	20	$\mathbf{z}_2$	95.86 %	40.60 %

4.4. Phonetic Evaluation of Speaker Conversion

In order to assess the success of the voice conversion, we first performed a detailed phonetic analysis focusing on the segmental quality, suprasegmental pitch contours, and aspects of general voice quality.

In order to obtain a qualitative impression of how well the segmental content, or the phone sequence, was realized in the target voices, we manually segmented three source utterances,[2] which were realized both in a male and a female source voice each, and carried out a fine-grained phonetic comparison. The used utterances and the corresponding source voices came from the test set and were not seen during training to be able to evaluate the out-of dataset performance of our model. This analysis yielded a high degree of correspondence between source and target. The segmental material was fully reproduced in most details, including variations in plosive aspiration, phone deletions, or assimilations. However, on very few occasions, the segmental quality of the source utterance is not convincingly preserved. One instance is the perceptual impression of a sibilant as a non-sibilant, probably due to a lower intensity of the higher frequencies that characterizes the target speaker. Also, voicing features of the source are not preserved in all cases.

Pitch contours may be a crucial indicator for the success of our conversion approach, as they transport linguistic content, shifting dynamically in course of the utterance, but also characterize the global voice of an individual speaker. Ideally, the contour shape of the source utterance should therefore be preserved, while the pitch level should be realized in the voice range of the target voice. We compared the time normalized pitch contours of source voices and voice targets with the conversion results. Our analysis shows that the pitch levels of the target voices are mimicked very successfully, while the local pitch contours show occasional deviations from the local contours of the source utterance (cf. Fig. 4).

To also analyze whether the voice characteristics of the target speakers were suitably mimicked after conversion, we calculated long term average spectra (LTAS) on the voiced parts of the various utterances, and compared the results of sources and target voices with the conversion result. Ideally, the converted voices should mimick the LTAS of the target voices. The results indicate that in most cases, the LTAS of the converted utterances are closer to those of the target voices than to those of the source voice, but for some voices, the result is inconclusive (cf. Fig. 5)

[2] available on `http://go.upb.de/vcex`

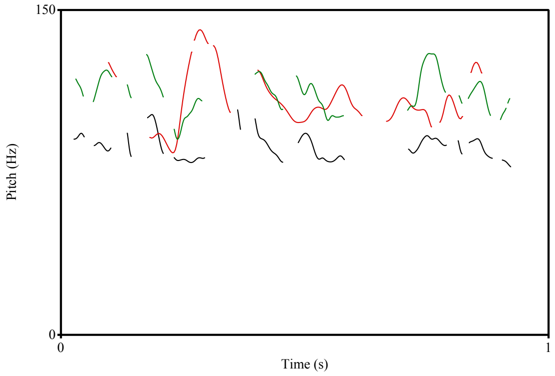


Figure 4: *Example of a time normalized source pitch contour (black), a contour of the target speaker (red), and the converted utterance (green). The local contour dynamics of the source is preserved as well as the global pitch height and range.*

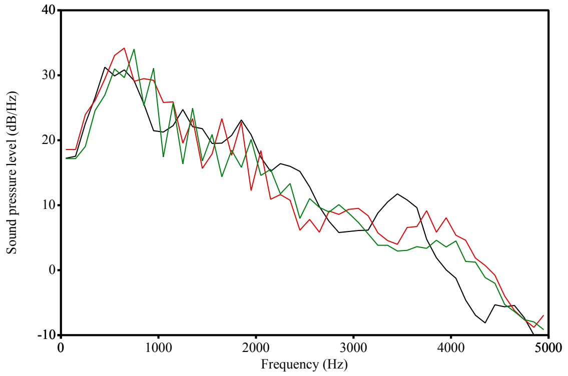


Figure 5: *Example of an LTAS of the source (black), the target speaker (red), and the converted utterance (green). The LTAS of the converted utterance mostly follows the pattern of the target speaker.*

5. Conclusions

In this contribution we proposed a voice conversion system, which works even with out-of dataset speakers. This system combines a new network architecture for the FHVAE, which uses convolutional network layers in both the encoders and the decoder, with a WaveNet for speech synthesis. We have shown that the proposed version of an FHVAE is able to model the linguistic content of speech at a higher temporal resolution compared to the original FHVAE, which employs LSTM network layers. Furthermore, the proposed system shows good results for voice conversion.

6. Acknowledgements

This work was in part supported by DFG under contract no. Ha3455/15-1 within the Research Unit FOR 2457 (acoustic sensor networks).

7. References

[1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[2] K. Kobayashi, S. Takamichi, S. Nakamura, and T. Toda, "The nu-naist voice conversion system for the voice conversion challenge 2016," in *INTERSPEECH*, 2016.

[3] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," in *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*, April 1988, pp. 655–658 vol.1.

[4] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, March 1998.

[5] Z. Wu, T. Virtanen, E. S. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1506–1521, Oct 2014.

[6] L. Chen, Z. Ling, L. Liu, and L. Dai, "Voice conversion using deep neural networks with layer-wise generative training," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1859–1872, Dec 2014.

[7] D. Sundermann, H. Höge, A. Bonafonte, H. Ney, A. Black, and S. Narayanan, "Text-independent voice conversion based on unit selection," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 1, May 2006, pp. I–I.

[8] C. Hsu, H. Hwang, Y. Wu, Y. Tsao, and H. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Dec 2016, pp. 1–6.

[9] T. Kaneko and H. Kameoka, "Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks," *arXiv e-prints*, p. arXiv:1711.11293, Nov 2017.

[10] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 266–273.

[11] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," *CoRR*, vol. abs/1711.00937, 2017. [Online]. Available: http://arxiv.org/abs/1711.00937

[12] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Acvae-vc: Non-parallel many-to-many voice conversion with auxiliary classifier variational autoencoder," *arXiv preprint arXiv:1808.05092*, 2018.

[13] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *CoRR*, vol. abs/1901.08810, 2019. [Online]. Available: http://arxiv.org/abs/1901.08810

[14] W. Hsu, Y. Zhang, and J. R. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," *CoRR*, vol. abs/1709.07902, 2017. [Online]. Available: http://arxiv.org/abs/1709.07902

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

[16] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[17] 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," 2016.

[18] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, "Variational lossy autoencoder," *CoRR*, vol. abs/1611.02731, 2016. [Online]. Available: http://arxiv.org/abs/1611.02731

[19] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM," 1993.