Unsupervised Learning of Disentangled and Interpretable Representations from Sequential Data

Wei-Ning Hsu, Yu Zhang, and James Glass
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{wnhsu,yzhang87,glass}@csail.mit.edu

Abstract

We present a factorized hierarchical variational autoencoder, which learns disentangled and interpretable representations from sequential data without supervision. Specifically, we exploit the multi-scale nature of information in sequential data by formulating it explicitly within a factorized hierarchical graphical model that imposes sequence-dependent priors and sequence-independent priors to different sets of latent variables. The model is evaluated on two speech corpora to demonstrate, qualitatively, its ability to transform speakers or linguistic content by manipulating different sets of latent variables; and quantitatively, its ability to outperform an i-vector baseline for speaker verification and reduce the word error rate by as much as 35% in mismatched train/test scenarios for automatic speech recognition tasks.

1 Introduction

Unsupervised learning is a powerful methodology that can leverage vast quantities of unannotated data in order to learn useful representations that can be incorporated into subsequent applications in either supervised or unsupervised fashions. One of the principle approaches to unsupervised learning is probabilistic generative modeling. Recently, there has been significant interest in three classes of deep probabilistic generative models: 1) Variational Autoencoders (VAEs) [24, 36, 23], 2) Generative Adversarial Networks (GANs) [12], and 3) auto-regressive models [31, 41]; more recently, there are also studies combining multiple classes of models [7, 28, 27]. While GANs bypass any inference of latent variables, and auto-regressive models abstain from using latent variables, VAEs jointly learn an inference model and a generative model, allowing them to infer latent variables from observed data.

Despite successes with VAEs, understanding the underlying factors that latent variables associate with is a major challenge. Some research focuses on the supervised or semi-supervised setting using VAEs [22, 18]. There is also research attempting to develop weakly supervised or unsupervised methods to learn disentangled representations, such as DC-IGN [26], InfoGAN [1], and β-VAE [14]. There is yet another line of research analyzing the latent variables with labeled data after the model is trained [35, 16]. While there has been much research investigating static data, such as the aforementioned ones, there is relatively little research on learning from sequential data [9, 4, 2, 10, 8, 19, 38]. Moreover, to the best of our knowledge, there has not been any attempt to learn disentangled and interpretable representations without supervision from sequential data. The information encoded in sequential data, such as speech, video, and text, is naturally multi-scaled; in speech for example, information about the channel, speaker, and linguistic content is encoded in the statistics at the session, utterance, and segment levels, respectively. By leveraging this source of constraint, we can learn disentangled and interpretable factors in an unsupervised manner.

In this paper, we propose a novel factorized hierarchical variational autoencoder, which learns disentangled and interpretable latent representations from sequential data without supervision by

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

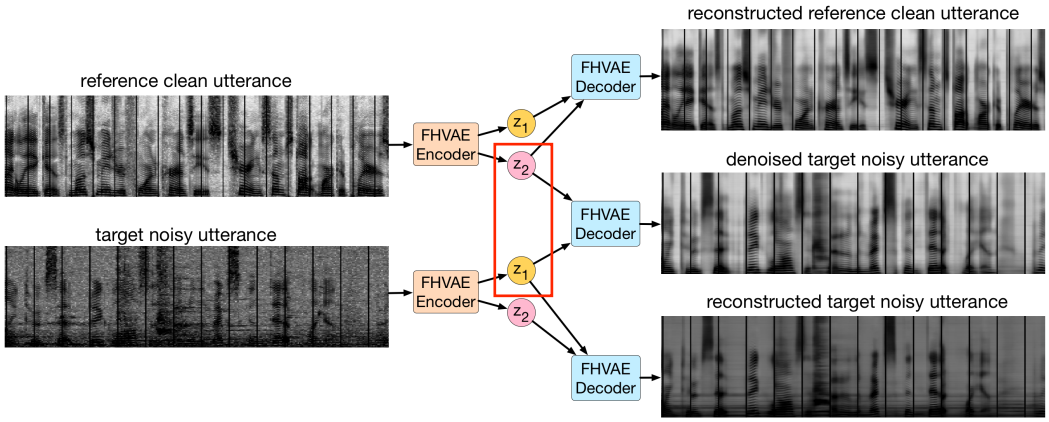


Figure 1: FHVAE ($\alpha = 0$) decoding results of three combinations of *latent segment variables* z_1 and *latent sequence variables* z_2 from two utterances in Aurora-4: a clean one (top-left) and a noisy one (bottom-left). FHVAEs learn to encode local attributes, such as linguistic content, into z_1, and encode global attributes, such as noise level, into z_2. Therefore, by replacing z_2 of a noisy utterance with z_2 of a clean utterance, an FHVAE decodes a denoised utterance (middle-right) that preserves the linguistic content. Reconstruction results of the clean and noisy utterances are also shown on the right. Audio samples are available at `https://youtu.be/naJZITvCfI4`.

explicitly modeling the multi-scaled information with a factorized hierarchical graphical model. The inference model is designed such that the model can be optimized at the segment level, instead of at the sequence level, which may cause scalability issues when sequences become too long. A sequence-to-sequence neural network architecture is applied to better capture temporal relationships. We evaluate the proposed model on two speech datasets. Qualitatively, the model demonstrates an ability to factorize sequence-level and segment-level attributes into different sets of latent variables. Quantitatively, the model achieves 2.38% and 1.34% equal error rate on unsupervised and supervised speaker verification tasks respectively, which outperforms an i-vector baseline. On speech recognition tasks, it reduces the word error rate in mismatched train/test scenarios by up to 35%.

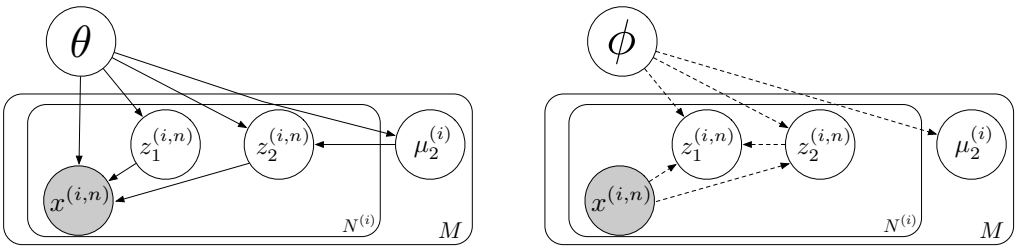
The rest of the paper is organized as follows. In Section 2, we introduce our proposed model, and describe the neural network architecture in Section 3. Experimental results are reported in Section 4. We discuss related work in Section 5, and conclude our work as well as discuss future research plans in Section 6.

2 Factorized Hierarchical Variational Autoencoder

Generation of sequential data, such as speech, often involves multiple independent factors operating at different time scales. For instance, the speaker identity affects fundamental frequency (F0) and volume at the sequence level, while phonetic content only affects spectral contour and durations of formants at the segmental level. This multi-scale behavior results in the fact that some attributes, such as F0 and volume, tend to have a smaller amount of variation within an utterance, compared to between utterances; while other attributes, such as phonetic content, tend to have a similar amount of variation within and between utterances.

We refer to the first type of attributes as *sequence-level attributes*, and the other as *segment-level attributes*. In this work, we achieve disentanglement and interpretability by encoding the two types of attributes into *latent sequence variables* and *latent segment variables* respectively, where the former is regularized by an sequence-dependent prior and the latter by an sequence-independent prior.

We now formulate a generative process for speech and propose our Factorized Hierarchical Variational Autoencoder (FHVAE). Consider some dataset $\mathcal{D} = \{\boldsymbol{X}^{(i)}\}_{i=1}^{M}$ consisting of M i.i.d. sequences, where $\boldsymbol{X}^{(i)} = \{\boldsymbol{x}^{(i,n)}\}_{n=1}^{N^{(i)}}$ is a sequence of $N^{(i)}$ observed variables. $N^{(i)}$ is referred to as the *number of segments* for the i-th sequence, and $\boldsymbol{x}^{(i,n)}$ is referred to as the n-th *segment* of the i-th sequence. Note that here a "segment" refers to a variable of smaller temporal scale compared to the



(a) Generative Model (b) Inference Model

Figure 2: Graphical illustration of the proposed generative model and inference model. Grey nodes denote the observed variables, and white nodes are the hidden variables.

"sequence", which is in fact a sub-sequence. We will drop the index i whenever it is clear that we are referring to terms associated with a single sequence. We assume that each sequence $\boldsymbol{X}$ is generated from some random process involving the latent variables $\boldsymbol{Z}_1 = \{\boldsymbol{z}_1^{(n)}\}_{n=1}^N$, $\boldsymbol{Z}_2 = \{\boldsymbol{z}_2^{(n)}\}_{n=1}^N$, and $\boldsymbol{\mu}_2$. The following generation process as illustrated in Figure 2(a) is considered: (1) a *s-vector* $\boldsymbol{\mu}_2$ is drawn from a prior distribution $p_\theta(\boldsymbol{\mu}_2)$; (2) N i.i.d. *latent sequence variables* $\{\boldsymbol{z}_2^{(n)}\}_{n=1}^N$ and *latent segment variables* $\{\boldsymbol{z}_1^{(n)}\}_{n=1}^N$ are drawn from a sequence-dependent prior distribution $p_\theta(\boldsymbol{z}_2|\boldsymbol{\mu}_2)$ and a sequence-independent prior distribution $p_\theta(\boldsymbol{z}_1)$ respectively; (3) N i.i.d. observed variables $\{\boldsymbol{x}^{(n)}\}_{n=1}^N$ are drawn from a conditional distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z}_1, \boldsymbol{z}_2)$. The joint probability for a sequence is formulated in Eq. 1:

$$p_\theta(\boldsymbol{X}, \boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{\mu}_2) = p_\theta(\boldsymbol{\mu}_2) \prod_{n=1}^{N} p_\theta(\boldsymbol{x}^{(n)}|\boldsymbol{z}_1^{(n)}, \boldsymbol{z}_2^{(n)}) p_\theta(\boldsymbol{z}_1^{(n)}) p_\theta(\boldsymbol{z}_2^{(n)}|\boldsymbol{\mu}_2). \tag{1}$$

Specifically, we formulate each of the RHS term as follows:

$$p_\theta(\boldsymbol{x}|\boldsymbol{z}_1, \boldsymbol{z}_2) = \mathcal{N}(\boldsymbol{x}|f_{\mu_x}(\boldsymbol{z}_1, \boldsymbol{z}_2), diag(f_{\sigma_x^2}(\boldsymbol{z}_1, \boldsymbol{z}_2)))$$
$$p_\theta(\boldsymbol{z}_1) = \mathcal{N}(\boldsymbol{z}_1|\boldsymbol{0}, \sigma_{\boldsymbol{z}_1}^2\boldsymbol{I}), \quad p_\theta(\boldsymbol{z}_2|\boldsymbol{\mu}_2) = \mathcal{N}(\boldsymbol{z}_2|\boldsymbol{\mu}_2, \sigma_{\boldsymbol{z}_2}^2\boldsymbol{I}), \quad p_\theta(\boldsymbol{\mu}_2) = \mathcal{N}(\boldsymbol{\mu}_2|\boldsymbol{0}, \sigma_{\boldsymbol{\mu}_2}^2\boldsymbol{I}),$$

where the priors over the s-vectors $\boldsymbol{\mu}_2$ and the latent segment variables $\boldsymbol{z}_1$ are centered isotropic multivariate Gaussian distributions. The prior over the latent sequence variable $\boldsymbol{z}_2$ conditioned on $\boldsymbol{\mu}_2$ is an isotropic multivariate Gaussian centered at $\boldsymbol{\mu}_2$. The conditional distribution of the observed variable $\boldsymbol{x}$ is the multivariate Gaussian with a diagonal covariance matrix, whose mean and diagonal variance are parameterized by neural networks $f_{\mu_x}(\cdot,\cdot)$ and $f_{\sigma_x^2}(\cdot,\cdot)$ with input $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$. We use θ to denote the set of parameters in the generative model.

This generative model is factorized in a way such that the latent sequence variables $\boldsymbol{z}_2$ within a sequence are forced to be close to $\boldsymbol{\mu}_2$ as well as to each other in Euclidean distance, and therefore are encouraged to encode sequence-level attributes that may have larger variance across sequences, but smaller variance within sequences. The constraint to the latent segment variables $\boldsymbol{z}_1$ is imposed globally, and therefore encourages encoding of residual attributes whose variation is not distinguishable inter and intra sequences.

In the variational autoencoder framework, since the exact posterior inference is intractable, an inference model, $q_\phi(\boldsymbol{Z}_1^{(i)}, \boldsymbol{Z}_2^{(i)}, \boldsymbol{\mu}_2^{(i)}|\boldsymbol{X}^{(i)})$, that approximates the true posterior, $p_\theta(\boldsymbol{Z}_1^{(i)}, \boldsymbol{Z}_2^{(i)}, \boldsymbol{\mu}_2^{(i)}|\boldsymbol{X}^{(i)})$, for variational inference [20] is introduced. We consider the following inference model as Figure 2(b):

$$q_\phi(\boldsymbol{Z}_1^{(i)}, \boldsymbol{Z}_2^{(i)}, \boldsymbol{\mu}_2^{(i)}|\boldsymbol{X}^{(i)}) = q_\phi(\boldsymbol{\mu}_2^{(i)}) \prod_{n=1}^{N^{(i)}} q_\phi(\boldsymbol{z}_1^{(i,n)}|\boldsymbol{x}^{(i,n)}, \boldsymbol{z}_2^{(i,n)}) q_\phi(\boldsymbol{z}_2^{(i,n)}|\boldsymbol{x}^{(i,n)})$$
$$q_\phi(\boldsymbol{\mu}_2^{(i)}) = \mathcal{N}(\boldsymbol{\mu}_2^{(i)}|g_{\mu_{\boldsymbol{\mu}_2}}(i), \sigma_{\tilde{\boldsymbol{\mu}}_2}^2\boldsymbol{I}), \quad q_\phi(\boldsymbol{z}_2|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}_2|g_{\mu_{\boldsymbol{z}_2}}(\boldsymbol{x}), diag(g_{\sigma_{\boldsymbol{z}_2}^2}(\boldsymbol{x})))$$
$$q_\phi(\boldsymbol{z}_1|\boldsymbol{x}, \boldsymbol{z}_2) = \mathcal{N}(\boldsymbol{z}_1|g_{\mu_{\boldsymbol{z}_1}}(\boldsymbol{x}, \boldsymbol{z}_2), diag(g_{\sigma_{\boldsymbol{z}_1}^2}(\boldsymbol{x}, \boldsymbol{z}_2))),$$

where the posteriors over $\boldsymbol{\mu}_2$, $\boldsymbol{z}_1$, and $\boldsymbol{z}_2$ are all multivariate diagonal Gaussian distributions. Note that the mean of the posterior distribution of $\boldsymbol{\mu}_2$ is not directly inferred from $\boldsymbol{X}$, but instead is regarded as

part of inference model parameters, with one for each utterance, which would be optimized during training. Therefore, $g_{\mu_{\boldsymbol{\mu}_2}}(\cdot)$ can be seen as a lookup table, and we use $\tilde{\boldsymbol{\mu}}_2^{(i)} = g_{\mu_{\boldsymbol{\mu}_2}}(i)$ to denote the posterior mean of $\boldsymbol{\mu}_2$ for the i-th sequence; we fix the posterior covariance matrix of $\boldsymbol{\mu}_2$ for all sequences. Similar to the generative model, $g_{\mu_{\boldsymbol{z}_2}}(\cdot)$, $g_{\sigma^2_{\boldsymbol{z}_2}}(\cdot)$, $g_{\mu_{\boldsymbol{z}_1}}(\cdot,\cdot)$, and $g_{\sigma^2_{\boldsymbol{z}_1}}(\cdot,\cdot)$ are also neural networks whose parameters along with $g_{\mu_{\boldsymbol{\mu}_2}}(\cdot)$ are denoted collectively by ϕ. The variational lower bound for this inference model on the marginal likelihood of a sequence $\boldsymbol{X}$ is derived as follows:

$$\mathcal{L}(\theta,\phi;\boldsymbol{X}) = \sum_{n=1}^{N} \mathcal{L}(\theta,\phi;\boldsymbol{x}^{(n)}|\tilde{\boldsymbol{\mu}}_2) + \log p_\theta(\tilde{\boldsymbol{\mu}}_2) + const$$

$$\begin{aligned}\mathcal{L}(\theta,\phi;\boldsymbol{x}^{(n)}|\tilde{\boldsymbol{\mu}}_2) =& \mathbb{E}_{q_\phi(\boldsymbol{z}_1^{(n)},\boldsymbol{z}_2^{(n)}|\boldsymbol{x}^{(n)})}\big[\log p_\theta(\boldsymbol{x}^{(n)}|\boldsymbol{z}_1^{(n)},\boldsymbol{z}_2^{(n)})\big] \\ &- \mathbb{E}_{q_\phi(\boldsymbol{z}_2^{(n)}|\boldsymbol{x}^{(n)})}\big[D_{KL}(q_\phi(\boldsymbol{z}_1^{(n)}|\boldsymbol{x}^{(n)},\boldsymbol{z}_2^{(n)})||p_\theta(\boldsymbol{z}_1^{(n)}))\big] \\ &- D_{KL}(q_\phi(\boldsymbol{z}_2^{(n)}|\boldsymbol{x}^{(n)})||p_\theta(\boldsymbol{z}_2^{(n)}|\tilde{\boldsymbol{\mu}}_2)).\end{aligned}$$

The detailed derivation can be found in Appendix A. Because the approximated posterior of $\boldsymbol{\mu}_2$ does not depend on the sequence $\boldsymbol{X}$, the *sequence variational lower bound* $\mathcal{L}(\theta,\phi;\boldsymbol{X})$ can be decomposed into the sum of $\mathcal{L}(\theta,\phi;\boldsymbol{x}^{(n)}|\tilde{\boldsymbol{\mu}}_2)$, the *conditional segment variational lower bounds*, over segments, plus the log prior probability of $\tilde{\boldsymbol{\mu}}_2$ and a constant. Therefore, instead of sampling a batch at the sequence level to maximize the sequence variational lower bound, we can sample a batch at the segment level to maximize the *segment variational lower bound*:

$$\mathcal{L}(\theta,\phi;\boldsymbol{x}^{(n)}) = \mathcal{L}(\theta,\phi;\boldsymbol{x}^{(n)}|\tilde{\boldsymbol{\mu}}_2) + \frac{1}{N}\log p_\theta(\tilde{\boldsymbol{\mu}}_2) + const. \tag{2}$$

This approach provides better scalability when the sequences are extremely long, such that computing an entire sequence for a batched update is too computationally expensive.

In this paper we only introduce two scales of attributes; however, one can easily extend this model to more scales by simply introducing $\boldsymbol{\mu}_k$ for $k = 2, 3, \cdots$[1] that constrains the prior distribution of latent variables at more scales, such as having session-dependent prior or dataset-dependent prior.

2.1 Discriminative Objective

The idea of having sequence-specific priors for each sequence is to encourage the model to encode the sequence-level attributes and the segment-level attributes into different sets of latent variables. However, when $\boldsymbol{\mu}_2 = 0$ for all sequences, the prior probability of the s-vector is maximized, and the KL-divergence of the inferred posterior of $\boldsymbol{z}_2$ is measured from the same conditional prior for all sequences. This would result in trivial s-vectors $\boldsymbol{\mu}_2$, and therefore $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ would not be factorized to encode sequence and segment attributes respectively.

To encourage $\boldsymbol{z}_2$ to encode sequence-level attributes, we use $\boldsymbol{z}_2^{(i,n)}$, which is inferred from $\boldsymbol{x}^{(i,n)}$, to infer the sequence index i of $\boldsymbol{x}^{(i,n)}$. We formulate the discriminative objective as:

$$\begin{aligned}\log p(i|\boldsymbol{z}_2^{(i,n)}) &= \log p(\boldsymbol{z}_2^{(i,n)}|i) - \log\sum_{j=1}^{M} p(\boldsymbol{z}_2^{(i,n)}|j) \quad (p(i) \text{ is assumed uniform}) \\ &:= \log p_\theta(\boldsymbol{z}_2^{(i,n)}|\tilde{\boldsymbol{\mu}}_2^{(i)}) - \log\big(\sum_{j=1}^{M} p_\theta(\boldsymbol{z}_2^{(i,n)}|\tilde{\boldsymbol{\mu}}_2^{(j)})\big),\end{aligned}$$

Combining the discriminative objective using a weighting parameter α with the segment variational lower bound, the objective function to maximize then becomes:

$$\mathcal{L}^{dis}(\theta,\phi;\boldsymbol{x}^{(i,n)}) = \mathcal{L}(\theta,\phi;\boldsymbol{x}^{(i,n)}) + \alpha\log p(i|\boldsymbol{z}_2^{(i,n)}), \tag{3}$$

which we refer to as the *discriminative segment variational lower bound*.

[1]The index starts from 2 because we do not introduce the hierarchy to $\boldsymbol{z}_1$.

2.2 Inferring S-Vectors During Testing

During testing, we may want to use the s-vector $\boldsymbol{\mu}_2$ of an unseen sequence $\tilde{\boldsymbol{X}} = \{\tilde{\boldsymbol{x}}^{(n)}\}_{n=1}^{\tilde{N}}$ as the sequence-level attribute representation for tasks such as speaker verification. Since the exact maximum a posterior estimation of $\boldsymbol{\mu}_2$ is intractable, we approximate the estimation using the conditional segment variational lower bound as follows:

$$\begin{aligned}
\boldsymbol{\mu}_2^* &= \operatorname*{argmax}_{\boldsymbol{\mu}_2} \log p_\theta(\boldsymbol{\mu}_2|\tilde{\boldsymbol{X}}) = \operatorname*{argmax}_{\boldsymbol{\mu}_2} \log p_\theta(\tilde{\boldsymbol{X}}, \boldsymbol{\mu}_2) \\
&= \operatorname*{argmax}_{\boldsymbol{\mu}_2} \Big(\sum_{n=1}^{\tilde{N}} \log p_\theta(\tilde{\boldsymbol{x}}^{(n)}|\boldsymbol{\mu}_2) \Big) + \log p_\theta(\boldsymbol{\mu}_2) \\
&\approx \operatorname*{argmax}_{\boldsymbol{\mu}_2} \sum_{n=1}^{\tilde{N}} \mathcal{L}(\theta, \phi; \tilde{\boldsymbol{x}}^{(n)}|\boldsymbol{\mu}_2) + \log p_\theta(\boldsymbol{\mu}_2). \qquad (4)
\end{aligned}$$

The closed form solution of $\boldsymbol{\mu}_2^*$ can be derived by differentiating Eq. 4 w.r.t. $\boldsymbol{\mu}_2$ (see Appendix B):

$$\boldsymbol{\mu}_2^* = \frac{\sum_{n=1}^{\tilde{N}} g_{\mu_{\boldsymbol{z}_2}}(\tilde{\boldsymbol{x}}^{(n)})}{\tilde{N} + \sigma_{\boldsymbol{z}_2}^2 / \sigma_{\boldsymbol{\mu}_2}^2}. \qquad (5)$$

3 Sequence-to-Sequence Autoencoder Model Architecture

In this section, we introduce the detailed neural network architectures for our proposed FHVAE. Let a segment $\boldsymbol{x} = x_{1:T}$ be a sub-sequence of $\boldsymbol{X}$ that contains T time steps, and x_t denotes the t-th time step of $\boldsymbol{x}$. We use recurrent network architectures for encoders that capture the temporal relationship among time steps, and generate a summarized fixed-dimension vector after consuming an entire sub-sequence. Likewise, we adopt a recurrent network architecture that generates a frame step by step conditioned on the latent variables $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$. The complete network can be seen as a stochastic sequence-to-sequence autoencoder that encodes $x_{1:T}$ stochastically into $\boldsymbol{z}_1, \boldsymbol{z}_2$, and stochastically decodes from them back to $x_{1:T}$.

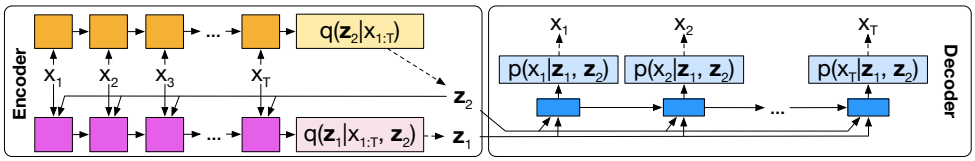


Figure 3: Sequence-to-sequence factorized hierarchical variational autoencoder. Dashed lines indicate the sampling process using the reparameterization trick [24]. The encoders for $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are pink and amber, respectively, while the decoder for $\boldsymbol{x}$ is blue. Darker colors denote the recurrent neural networks, while lighter colors denote the fully-connected layers predicting the mean and log variance.

Figure 3 shows our proposed Seq2Seq-FHVAE architecture.[2] Here we show the detailed formulation:

$$\begin{aligned}
(\boldsymbol{h}_{\boldsymbol{z}_2,t}, \boldsymbol{c}_{\boldsymbol{z}_2,t}) &= \mathrm{LSTM}(x_{t-1}, \boldsymbol{h}_{\boldsymbol{z}_2,t-1}, \boldsymbol{c}_{\boldsymbol{z}_2,t-1}; \phi_{\mathrm{LSTM},\boldsymbol{z}_2}) \\
q_\phi(\boldsymbol{z}_2|x_{1:T}) &= \mathcal{N}(\boldsymbol{z}_2|\,\mathrm{MLP}(\boldsymbol{h}_{\boldsymbol{z}_2,T}; \phi_{\mathrm{MLP}_{\boldsymbol{\mu}},\boldsymbol{z}_2}), \mathrm{diag}(\exp(\mathrm{MLP}(\boldsymbol{h}_{\boldsymbol{z}_2,T}; \phi_{\mathrm{MLP}_{\boldsymbol{\sigma}^2},\boldsymbol{z}_2})))) \\
(\boldsymbol{h}_{\boldsymbol{z}_1,t}, \boldsymbol{c}_{\boldsymbol{z}_1,t}) &= \mathrm{LSTM}([x_{t-1}; \boldsymbol{z}_2], \boldsymbol{h}_{\boldsymbol{z}_1,t-1}, \boldsymbol{c}_{\boldsymbol{z}_1,t-1}; \phi_{\boldsymbol{z}_1}) \\
q_\phi(\boldsymbol{z}_1|x_{1:T}, \boldsymbol{z}_2) &= \mathcal{N}(\boldsymbol{z}_1|\,\mathrm{MLP}(\boldsymbol{h}_{\boldsymbol{z}_1,T}; \phi_{\mathrm{MLP}_{\boldsymbol{\mu}},\boldsymbol{z}_1}), \mathrm{diag}(\exp(\mathrm{MLP}(\boldsymbol{h}_{\boldsymbol{z}_1,T}; \phi_{\mathrm{MLP}_{\boldsymbol{\sigma}^2},\boldsymbol{z}_1})))) \\
(\boldsymbol{h}_{\boldsymbol{x},t}, \boldsymbol{c}_{\boldsymbol{x},t}) &= \mathrm{LSTM}([\boldsymbol{z}_1; \boldsymbol{z}_2], \boldsymbol{h}_{\boldsymbol{x},t-1}, \boldsymbol{c}_{\boldsymbol{x},t-1}; \phi_{\boldsymbol{x}}) \\
p_\theta(x_t|\boldsymbol{z}_1, \boldsymbol{z}_2) &= \mathcal{N}(\boldsymbol{x}_t|\,\mathrm{MLP}(\boldsymbol{h}_{\boldsymbol{x},t}; \phi_{\mathrm{MLP}_{\boldsymbol{\mu}},\boldsymbol{x}}), \mathrm{diag}(\exp(\mathrm{MLP}(\boldsymbol{h}_{\boldsymbol{x},t}; \phi_{\mathrm{MLP}_{\boldsymbol{\sigma}^2},\boldsymbol{x}})))),
\end{aligned}$$

where LSTM refers to a long short-term memory recurrent neural network [15], and MLP refers to a multi-layer perceptron, ϕ_* are the related weight matrices. None of the neural network parameters are shared. We refer to this model as Seq2Seq-FHVAE. Log-likelihood and qualitative comparison with alternative architectures can be found in Appendix D.

[2] Best viewed in color.

4 Experiments

We use speech, which inherently contains information at multiple scales, such as channel, speaker, and linguistic content to test our model. Learning to disentangle the mixed information from the surface representation is essential for a wide variety of speech applications: for example, noise robust speech recognition [44, 40, 39, 17], speaker verification [6], and voice conversion [42, 30, 25].

The following two corpora are used for our experiments: (1) **TIMIT** [11], which contains broadband 16kHz recordings of phonetically-balanced read speech. A total of 6300 utterances (5.4 hours) are presented with 10 sentences from each of 630 speakers, of which approximately 70% are male and 30% are female. (2) **Aurora-4** [33], a broadband corpus designed for noisy speech recognition tasks based on the Wall Street Journal corpus (WSJ0) [32]. Two microphone types, CLEAN/CHANNEL are included, and six noise types are artificially added to both microphone types, which results in four conditions: CLEAN, CHANNEL, NOISY, and CHANNEL+NOISY. Two 14 hour training sets are used, where one is clean and the other is a mix of all four conditions. The same noise types and microphones are used to generate the development and test sets, which both consist of 330 utterances from all four conditions, resulting in 4,620 utterances in total for each set.

All speech is represented as a sequence of 80 dimensional Mel-scale filter bank (FBank) features or 200 dimensional log-magnitude spectrum (only for audio reconstruction), computed every 10ms. Mel-scale features are a popular auditory approximation for many speech applications [29]. We consider a sample $\boldsymbol{x}$ to be a 200ms sub-sequence, which is on the order of the length of a syllable, and implies $T = 20$ for each $\boldsymbol{x}$. For the Seq2Seq-FHVAE model, all the LSTM and MLP networks are one-layered, and Adam [21] is used for optimization. More details of the model architecture and training procedure can be found in Appendix C.

4.1 Qualitative Evaluation of the Disentangled Latent Variables

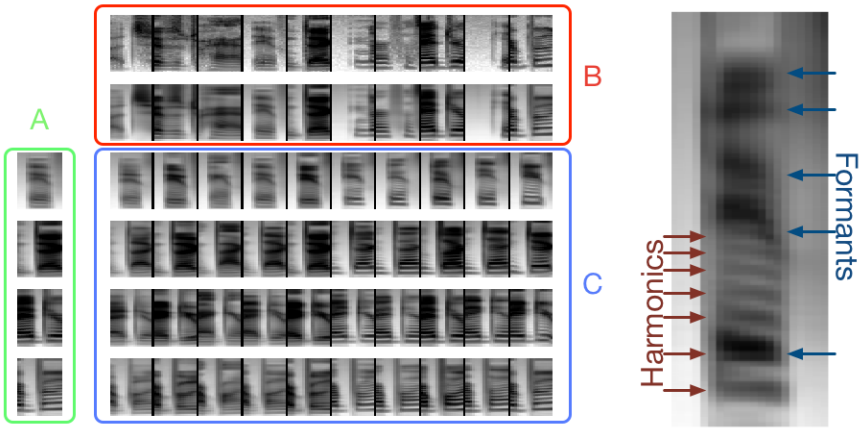


Figure 4: (left) Examples generated by varying different latent variables. (right) An illustration of harmonics and formants in filter bank images. The green block 'A' contains four reconstructed examples. The red block 'B' contains ten original sequences on the first row with the corresponding reconstructed examples on the second row. The entry on the i-th row and the j-th column in the blue block 'C' is the reconstructed example using the latent segment variable $\boldsymbol{z}_1$ of the i-th row from block 'A' and the latent sequence variable $\boldsymbol{z}_2$ of the j-th column from block 'B'.

To qualitatively study the factorization of information between the latent segment variable $\boldsymbol{z}_1$ and the latent sequence variable $\boldsymbol{z}_2$, we generate examples $\boldsymbol{x}$ by varying each of them respectively. Figure 4 shows 40 examples in block 'C' of all the combinations of the 4 latent segment variables extracted from block 'A' and the 10 latent sequence variables extracted from block 'B.' The top two examples from block 'A' and the five leftmost examples from block 'B' are from male speakers, while the rest are from female speakers, which show higher fundamental frequencies and harmonics.[3]

[3]The harmonics corresponds to horizontal dark stripes in the figure; the more widely these stripes are spaced vertically, the higher the fundamental frequency of the speaker is.

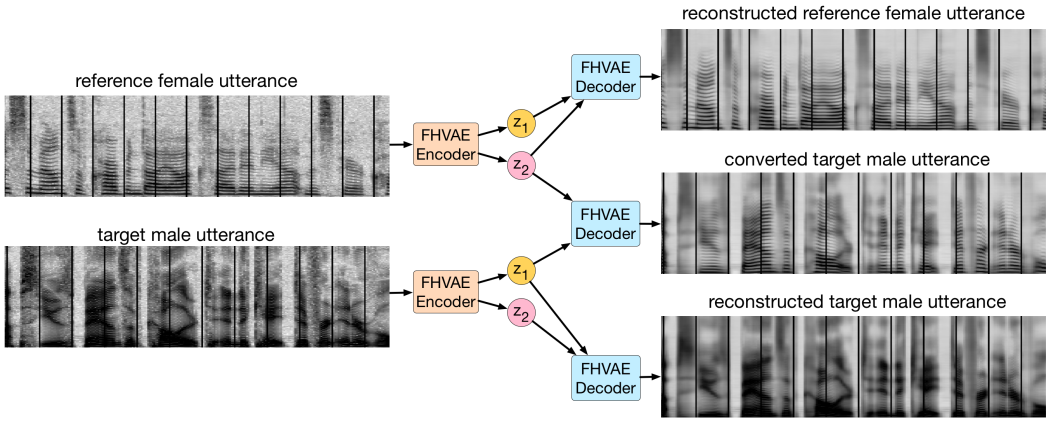


Figure 5: FHVAE ($\alpha = 0$) decoding results of three combinations of *latent segment variables* z_1 and *latent sequence variables* z_2 from one male-speaker utterance (top-left) and one female-speaker utterance (bottom-left) in Aurora-4. By replacing z_2 of a male-speaker utterance with z_2 of a female-speaker utterance, an FHVAE decodes a voice-converted utterance (middle-right) that preserves the linguistic content. Audio samples are available at `https://youtu.be/VMX3IZYWYdg`.

We can observe that along each row in block 'C', the linguistic phonetic-level content, which manifests itself in the form of the spectral contour and temporal position of formants, as well as the relative position between formants, is very similar between elements; the speaker identity however changes (e.g., harmonic structure). On the other hand, for each column we see that the speaker identity remains consistent, despite the change of linguistic content. The factorization of the sequence-level attributes and the segment-level attributes of our proposed Seq2Seq-FHVAE is clearly evident. In addition, we also show examples of modifying an entire utterance in Figure 1 and 5, which achieves denoising by replacing the latent sequence variable of a noisy utterance with those of a clean utterance, and achieves voice conversion by replacing the latent sequence variable of one speaker with that of another speaker. Details of the operations we applied to modify an entire utterance as well as more larger-sized examples of different α values can be found in Appendix E. We also show extra latent space traversal experiments in Appendix H.

4.2 Quantitative Evaluation of S-Vectors – Speaker Verification

To quantify the performance of our model on disentangling the utterance-level attributes from the segment-level attributes, we present experiments on a speaker verification task on the TIMIT corpus to evaluate how well the estimated $\boldsymbol{\mu}_2$ encodes speaker-level information.[4] As a sanity check, we modify Eq. 5 to estimate an alternative s-vector based on latent segment variables $\boldsymbol{z}_1$ as follows: $\boldsymbol{\mu}_1 = \sum_{n=1}^{\tilde{N}} g_{\mu_{\boldsymbol{z}_1}}(\tilde{\boldsymbol{x}}^{(n)})/(\tilde{N} + \sigma^2_{\boldsymbol{z}_1})$. We use the i-vector method [6] as the baseline, which is the representation used in most state-of-the-art speaker verification systems. They are in a low dimensional subspace of the Gaussian mixture model (GMM) mean supervector space, where the GMM is the universal background model (UBM) that models the generative process of speech. I-vectors, $\boldsymbol{\mu}_1$, and $\boldsymbol{\mu}_2$ can all be extracted without supervision; when speaker labels are available during training, techniques such as linear discriminative analysis (LDA) can be applied to further improve the linear separability of the representation. For all experiments, we use the fast scoring approach in [5] that uses cosine similarity as the similarity metric and compute the equal error rate (EER). More details about the experimental settings can be found in Appendix F.

We compare different dimensions for both features as well as different α's in Eq.3 for training FHVAE models. The results in Table 1 show that the 16 dimensional s-vectors $\boldsymbol{\mu}_2$ outperform i-vector baselines in both unsupervised (Raw) and supervised (LDA) settings for all α's as shown in the fourth column; the more discriminatively the FHVAE model is trained (i.e., with larger α), the better speaker

[4]TIMIT is not a standard corpus for speaker verification, but it is a good corpus to show the utterance-level attribute we learned via this task, because the main attribute that is consistent within an utterance is speaker identity, while in Aurora-4 both speaker identity and the background noise are consistent within an utterance.

verification results it achieves. Moreover, with the appropriately chosen dimension, a 32 dimensional $\boldsymbol{\mu}_2$ reaches an even lower EER at 1.34%. On the other hand, the negative results of using $\boldsymbol{\mu}_1$ also validate the success in disentangling utterance and segment level attributes.

Table 1: Comparison of speaker verification equal error rate (EER) on the TIMIT test set

Features	Dimension	α	Raw	LDA (12 dim)	LDA (24 dim)
i-vector	48	-	10.12%	6.25%	5.95%
	100	-	9.52%	6.10%	5.50%
	200	-	9.82%	6.54%	6.10%
$\boldsymbol{\mu}_2$	16	0	5.06%	4.02%	-
	16	10^{-1}	4.91%	4.61%	-
	16	10^{0}	3.87%	3.86%	-
	16	10^{1}	**2.38%**	**2.08%**	-
	32	10^{1}	**2.38%**	**2.08%**	**1.34%**
$\boldsymbol{\mu}_1$	16	10^{0}	22.77%	15.62%	-
	16	10^{1}	27.68%	22.17%	-
	32	10^{1}	22.47%	16.82%	17.26%

4.3 Quantitative Evaluation of the Latent Segment Variables – Domain Invariant ASR

Speaker adaptation and robust speech recognition in automatic speech recognition (ASR) can often be seen as domain adaptation problems, where available labeled data is limited and hence the data distributions during training and testing are mismatched. One approach to reduce the severity of this issue is to extract speaker/channel invariant features for the tasks.

As demonstrated in Section 4.2, the s-vector contains information about domains. Here we evaluate if the latent segment variables contains domain invariant linguistic information by evaluating on an ASR task: (1) train our proposed Seq2Seq-FHVAE using FBank feature on a set that covers different domains. (2) train an LSTM acoustic model [13, 37, 45] on the set that only covers partial domains using mean and log variance of the latent segment variable $\boldsymbol{z}_1$ extracted from the trained Seq2Seq-FHVAE. (3) test the ASR system on all domains. As a baseline, we also train the same ASR models but use the FBank features alone. Detailed configurations are in Appendix G.

For TIMIT we assume that male and female speakers constitute different domains, and show the results in Table 2. The first row of results shows that the ASR model trained on all domains (speakers) using FBank features as the upper bound. When trained on only male speakers, the phone error rate (PER) on female speakers increases by 16.1% for FBank features; however, for $\boldsymbol{z}_1$, despite the slight degradation on male speakers, the PER on the unseen domain, which are female speakers, improves by 6.6% compared to FBank features.

Table 2: TIMIT test phone error rate of acoustic models trained on different features and sets

Train Set and Configuration			Test PER by Set		
ASR	FHVAE	Features	Male	Female	All
Train All	-	FBank	20.1%	16.7%	19.1%
Train Male	-	FBank	**21.0%**	32.8%	25.2%
	Train All, $\alpha = 10$	$\boldsymbol{z}_1$	22.0%	**26.2%**	**23.5%**

On Aurora-4, four domains are considered, which are clean, noisy, channel, and noisy+channel (NC for short). We train the FHVAE on the development set for two purposes: (1) the FHVAE can be considered as a general feature extractor, which can be trained on an arbitrary collection of data that does not necessarily include the data for subsequent applications. (2) the dev set of Aurora-4 contains the domain label for each utterance so it is possible to control which domain has been observed by the FHVAE. Table 3 shows the word error rate (WER) results on Aurora-4, from which we can observe that the FBank representation suffers from severe domain mismatch problems; specifically, the WER

increases by 53.3% when noise is presented in mismatched microphone recordings (NC). In contrast, when the FHVAE is trained on data from all domains, using the latent segment variables as features reduce WER from 16% to 35% compare to baseline on mismatched domains, with less than 2% WER degradation on the matched domain. In addition, β-VAEs [14] are trained on the same data as the FHVAE to serve as the baseline feature extractor, from which we extract the latent variables $\boldsymbol{z}$ as the ASR feature and show the result in the third to the sixth rows. The β-VAE features outperform FBank in all mismatched domains, but are inferior to the latent segment variable $\boldsymbol{z}_1$ from the FHVAE in those domains. The results demonstrate the importance of learning not only disentangled, but also interpretable representations, which can be achieved by our proposed FHVAE models. As a sanity check, we replace $\boldsymbol{z}_1$ with $\boldsymbol{z}_2$, the latent sequence variable and train an ASR, which results in terrible WER performance as shown in the eighth row as expected.

Finally, we train another FHVAE on all domains excluding the combinatory NC domain, and shows the results in the last row in Table 3. It can be observed that the latent segment variable still outperforms the baseline feature with 30% lower WER on noise and channel combined data, even though the FHAVE has only seen noise and channel variation independently.

Table 3: Aurora-4 test word error rate of acoustic models trained on different features and sets

Train Set and Configuration			Test WER by Set				
ASR	{FH-,β-}VAE	Features	Clean	Noisy	Channel	NC	All
Train All	-	FBank	3.60%	7.06%	8.24%	18.49%	11.80%
Train Clean	-	FBank	**3.47%**	50.97%	36.99%	71.80%	55.51%
	Dev, $\beta = 1$	$\boldsymbol{z}$ (β-VAE)	4.95%	23.54%	31.12%	46.21%	32.47%
	Dev, $\beta = 2$	$\boldsymbol{z}$ (β-VAE)	3.57%	27.24%	30.56%	48.17%	34.75%
	Dev, $\beta = 4$	$\boldsymbol{z}$ (β-VAE)	3.89%	24.40%	29.80%	47.87%	33.38%
	Dev, $\beta = 8$	$\boldsymbol{z}$ (β-VAE)	5.32%	34.84%	36.13%	58.02%	42.76%
	Dev, $\alpha = 10$	$\boldsymbol{z}_1$ (FHVAE)	5.01%	**16.42%**	**20.29%**	**36.33%**	**24.41%**
	Dev, $\alpha = 10$	$\boldsymbol{z}_2$ (FHVAE)	41.08%	68.73%	61.89%	86.36%	72.53%
	Dev\NC, $\alpha = 10$	$\boldsymbol{z}_1$ (FHVAE)	5.25%	16.52%	19.30%	40.59%	26.23%

5 Related Work

A number of prior publications have extended VAEs to model structured data by altering the underlying graphical model to dynamic Bayesian networks, such as SRNN [4] and VRNN [10], or to hierarchical models, such as neural statistician [8] and SVAE [19]. These models have shown success in quantitatively increasing the log-likelihood, or qualitatively generating reasonable structured data by sampling. However, it remains unclear whether independent attributes are disentangled in the latent space. Moreover, the learned latent variables in these models are not interpretable without manually inspecting or using labeled data. In contrast, our work presents a VAE framework that addresses both problems by explicitly modeling the difference in the rate of temporal variation of the attributes that operate at different scales.

Our work is also related to β-VAE [14] with respect to unsupervised learning of disentangled representations with VAEs. The boosted KL-divergence penalty imposed in β-VAE training encourages disentanglement of independent attributes, but does not provide interpretability without supervision. We demonstrate in our domain invariant ASR experiments that learning interpretable representations is important for such applications, and can be achieved by our FHVAE model. In addition, the idea of boosting KL-divergence regularization is complimentary to our model, which can be potentially integrated for better disentanglement.

6 Conclusions and Future Work

We introduce the factorized hierarchical variational autoencoder, which learns disentangled and interpretable representations for sequence-level and segment-level attributes without any supervision. We verify the disentangling ability both qualitatively and quantitatively on two speech corpora. For future work, we plan to (1) extend to more levels of hierarchy, (2) investigate adversarial training for disentanglement, and (3) apply the model to other types of sequential data, such as text and videos.

References

[1] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, page 2172–2180, 2016.

[2] Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*, 2016.

[3] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, pages 2067–2075, 2015.

[4] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988, 2015.

[5] Najim Dehak, Reda Dehak, Patrick Kenny, Niko Brümmer, Pierre Ouellet, and Pierre Dumouchel. Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification. In *Interspeech*, volume 9, pages 1559–1562, 2009.

[6] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[7] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[8] Harrison Edwards and Amos Storkey. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

[9] Otto Fabius and Joost R van Amersfoort. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014.

[10] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems*, pages 2199–2207, 2016.

[11] John S Garofolo, Lori F Lamel, William M Fisher, Jonathon G Fiscus, and David S Pallett. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon technical report n*, 93, 1993.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[13] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional LSTM. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pages 273–278. IEEE, 2013.

[14] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] Wei-Ning Hsu, Yu Zhang, and James Glass. Learning latent representations for speech generation and transformation. In *Interspeech*, pages 1273–1277, 2017.

[17] Wei-Ning Hsu, Yu Zhang, and James Glass. Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In *Automatic Speech Recognition and Understanding (ASRU), 2017 IEEE Workshop on*. IEEE, 2017.

[18] Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Controllable text generation. *arXiv preprint arXiv:1703.00955*, 2017.

[19] Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in neural information processing systems*, pages 2946–2954, 2016.

[20] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[21] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[23] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. 2016.

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[25] Tomi Kinnunen, Lauri Juvela, Paavo Alku, and Junichi Yamagishi. Non-parallel voice conversion using i-vector plda: Towards unifying speaker verification and transformation. In *ICASSP*, 2017.

[26] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015.

[27] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[28] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[29] Nelson Mogran, Hervé Bourlard, and Hynek Hermansky. Automatic speech recognition: An auditory perspective. In *Speech processing in the auditory system*, pages 309–338. Springer, 2004.

[30] Toru Nakashika, Tetsuya Takiguchi, Yasuhiro Minami, Toru Nakashika, Tetsuya Takiguchi, and Yasuhiro Minami. Non-parallel training in voice conversion using an adaptive restricted boltzmann machine. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 24(11):2032–2045, November 2016.

[31] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

[32] Douglas B Paul and Janet M Baker. The design for the wall street journal-based csr corpus. In *Proceedings of the workshop on Speech and Natural Language*, pages 357–362. Association for Computational Linguistics, 1992.

[33] David Pearce. *Aurora working group: DSR front end LVCSR evaluation AU/384/02*. PhD thesis, Mississippi State University, 2002.

[34] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[35] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[36] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[37] Hasim Sak, Andrew W Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Interspeech*, pages 338–342, 2014.

[38] Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[39] Dmitriy Serdyuk, Kartik Audhkhasi, Philemon Brakel, Bhuvana Ramabhadran, Samuel Thomas, and Yoshua Bengio. Invariant representations for noisy speech recognition. *CoRR*, abs/1612.01928, 2016.

[40] Yusuke Shunohara. Adversarial multi-task learning of deep neural networks for robust speech recognition. In *Interspeeech*, pages 2369–2372, 2016.

[41] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*, 2016.

[42] Zhizheng Wu, Eng Siong Chng, and Haizhou Li. Conditional restricted boltzmann machine for voice conversion. In *ChinaSIP*, 2013.

[43] Dong Yu, Adam Eversole, Mike Seltzer, Kaisheng Yao, Zhiheng Huang, Brian Guenter, Oleksii Kuchaiev, Yu Zhang, Frank Seide, Huaming Wang, et al. An introduction to computational networks and the computational network toolkit. Technical report, Tech. Rep. MSR, Microsoft Research, 2014, http://codebox/cntk, 2014.

[44] Dong Yu, Michael Seltzer, Jinyu Li, Jui-Ting Huang, and Frank Seide. Feature learning in deep neural networks – studies on speech recognition tasks. *arXiv preprint arXiv:1301.3605*, 2013.

[45] Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass. Highway long short-term memory RNNs for distant speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5755–5759. IEEE, 2016.

A. Derivation of Sequence Variational Lower Bound

The variational lower bound for the marginal probability of a sequence $\boldsymbol{X}$ can be derived as follows:

$$
\begin{aligned}
\log p_\theta(\boldsymbol{X}) \geq & \mathcal{L}(\theta, \phi; \boldsymbol{X}) \\
= & \mathbb{E}_{q_\phi(\boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{\mu}_2 | \boldsymbol{X})}\big[\log \frac{p_\theta(\boldsymbol{\mu}_2)\prod_{n=1}^{N} p_\theta(\boldsymbol{x}^{(n)}|\boldsymbol{z}_1^{(n)}, \boldsymbol{z}_2^{(n)}) p_\theta(\boldsymbol{z}_1^{(n)}) p_\theta(\boldsymbol{z}_2^{(n)}|\boldsymbol{\mu}_2)}{q_\phi(\boldsymbol{\mu}_2)\prod_{n=1}^{N} q_\phi(\boldsymbol{z}_1^{(n)}|\boldsymbol{x}^{(n)}, \boldsymbol{z}_2^{(n)}) q_\phi(\boldsymbol{z}_2^{(n)}|\boldsymbol{x}^{(n)})}\big] \\
= & \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\boldsymbol{z}_1^{(n)}, \boldsymbol{z}_2^{(n)}|\boldsymbol{x}^{(n)})}\big[\log p_\theta(\boldsymbol{x}^{(n)}|\boldsymbol{z}_1^{(n)}, \boldsymbol{z}_2^{(n)})\big] \\
& - \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\boldsymbol{z}_2^{(n)}|\boldsymbol{x}^{(n)})}\big[D_{KL}(q_\phi(\boldsymbol{z}_1^{(n)}|\boldsymbol{x}^{(n)}, \boldsymbol{z}_2^{(n)})||p_\theta(\boldsymbol{z}_1^{(n)}))\big] \\
& - \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\boldsymbol{\mu}_2)}\big[D_{KL}(q_\phi(\boldsymbol{z}_2^{(n)}|\boldsymbol{x}^{(n)})||p_\theta(\boldsymbol{z}_2^{(n)}|\boldsymbol{\mu}_2))\big] \qquad (6) \\
& - D_{KL}(q_\phi(\boldsymbol{\mu}_2)||p_\theta(\boldsymbol{\mu}_2)). \qquad (7)
\end{aligned}
$$

The expected KL-divergence in Eq. 6 of two Gaussian distributions, $q_\phi(\boldsymbol{z}_2^{(n)}|\boldsymbol{x}^{(n)})$ and $p_\theta(\boldsymbol{z}_2^{(n)}|\boldsymbol{\mu}_2)$, over a Gaussian $q_\phi(\boldsymbol{\mu}_2) = \mathcal{N}(\boldsymbol{\mu}_2|\tilde{\boldsymbol{\mu}}_2, \sigma^2_{\tilde{\boldsymbol{\mu}}_2}\boldsymbol{I})$ can be computed analytically. Let J be the dimensionality of $\boldsymbol{z}_2$. Let $\hat{\boldsymbol{\mu}}_{\boldsymbol{z}_2}$ and $\hat{\boldsymbol{\sigma}}_{\boldsymbol{z}_2}$ denote the variational mean and standard deviation evaluated at $\boldsymbol{x}^{(n)}$, and let $\mu_{2,j}$, $\tilde{\mu}_{2,j}$, $\hat{\mu}_{\boldsymbol{z}_2,j}$ and $\hat{\sigma}_{\boldsymbol{z}_2,j}$ denote the j-th element of these vectors. We have:

$$
\begin{aligned}
& \mathbb{E}_{q_\phi(\boldsymbol{\mu}_2)}\big[D_{KL}(q_\phi(\boldsymbol{z}_2^{(n)}|\boldsymbol{x}^{(n)})||p_\theta(\boldsymbol{z}_2^{(n)}|\boldsymbol{\mu}_2))\big] \\
& = \mathbb{E}_{q_\phi(\boldsymbol{\mu}_2)}\big[D_{KL}(\mathcal{N}(\hat{\boldsymbol{\mu}}_{\boldsymbol{z}_2}, \hat{\boldsymbol{\sigma}}^2_{\boldsymbol{z}_2})||\mathcal{N}(\boldsymbol{\mu}_2, \sigma^2_{\boldsymbol{z}_2}\boldsymbol{I}))\big] \\
& = \mathbb{E}_{q_\phi(\boldsymbol{\mu}_2)}\big[-\frac{1}{2}\sum_{j=1}^{J}(1 + \log\frac{\hat{\sigma}^2_{\boldsymbol{z}_2,j}}{\sigma^2_{\boldsymbol{z}_2}} - \frac{(\hat{\mu}_{\boldsymbol{z}_2,j} - \mu_{2,j})^2 + \hat{\sigma}^2_{\boldsymbol{z}_2,j}}{\sigma^2_{\boldsymbol{z}_2}})\big] \\
& = -\frac{1}{2}\sum_{j=1}^{J}(1 + \log\frac{\hat{\sigma}^2_{\boldsymbol{z}_2,j}}{\sigma^2_{\boldsymbol{z}_2}} - \frac{\hat{\sigma}^2_{\boldsymbol{z}_2,j}}{\sigma^2_{\boldsymbol{z}_2}} - \mathbb{E}_{q_\phi(\boldsymbol{\mu}_2)}\big[\frac{(\hat{\mu}_{\boldsymbol{z}_2,j} - \mu_{2,j})^2}{\sigma^2_{\boldsymbol{z}_2}}\big]) \\
& = D_{KL}(\mathcal{N}(\hat{\boldsymbol{\mu}}_{\boldsymbol{z}_2}, \hat{\boldsymbol{\sigma}}^2_{\boldsymbol{z}_2})||\mathcal{N}(\tilde{\boldsymbol{\mu}}_2, \sigma^2_{\boldsymbol{z}_2})) + \frac{J}{2}\frac{\sigma^2_{\tilde{\boldsymbol{\mu}}_2}}{\sigma^2_{\boldsymbol{z}_2}} \\
& = D_{KL}(q_\phi(\boldsymbol{z}_2^{(n)}|\boldsymbol{x}^{(n)})||p_\theta(\boldsymbol{z}_2^{(n)}|\tilde{\boldsymbol{\mu}}_2)) + \frac{J}{2}\frac{\sigma^2_{\tilde{\boldsymbol{\mu}}_2}}{\sigma^2_{\boldsymbol{z}_2}} \qquad (8)
\end{aligned}
$$

The KL-divergence in Eq. 7 can also be computed analytically and rewritten as follows:

$$
\begin{aligned}
& D_{KL}(q_\phi(\boldsymbol{\mu}_2)||p_\theta(\boldsymbol{\mu}_2)) \\
& = D_{KL}(\mathcal{N}(\tilde{\boldsymbol{\mu}}_2, \sigma^2_{\tilde{\boldsymbol{\mu}}_2}\boldsymbol{I})||\mathcal{N}(0, \sigma^2_{\boldsymbol{\mu}_2}\boldsymbol{I})) \\
& = -\frac{1}{2}\sum_{j=1}^{J}(1 + \log\frac{\sigma^2_{\tilde{\boldsymbol{\mu}}_2}}{\sigma^2_{\boldsymbol{\mu}_2}} - \frac{(\tilde{\mu}_{2,j} - 0)^2 + \sigma^2_{\tilde{\boldsymbol{\mu}}_2}}{\sigma^2_{\boldsymbol{\mu}_2}}) \\
& = -\frac{1}{2}\sum_{j=1}^{J}(1 + \log\sigma^2_{\tilde{\boldsymbol{\mu}}_2}) - \frac{1}{2}\log 2\pi - \log p_\theta(\tilde{\boldsymbol{\mu}}_2) \qquad (9)
\end{aligned}
$$

By replacing Eq. 6 and 7 with Eq. 8 and 9 respectively, we rewrite the variational lower bound for a sequence $\boldsymbol{X}$ as follows:

$$
\begin{aligned}
\mathcal{L}(\theta, \phi; \boldsymbol{X}) = & \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\boldsymbol{z}_1^{(n)}, \boldsymbol{z}_2^{(n)} | \boldsymbol{x}^{(n)})} \big[\log p_\theta(\boldsymbol{x}^{(n)} | \boldsymbol{z}_1^{(n)}, \boldsymbol{z}_2^{(n)}) \big] \\
& - \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\boldsymbol{z}_2^{(n)} | \boldsymbol{x}^{(n)})} \big[D_{KL}(q_\phi(\boldsymbol{z}_1^{(n)} | \boldsymbol{x}^{(n)}, \boldsymbol{z}_2^{(n)}) || p_\theta(\boldsymbol{z}_1^{(n)})) \big] \\
& - \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\boldsymbol{\mu}_2)} \big[D_{KL}(q_\phi(\boldsymbol{z}_2^{(n)} | \boldsymbol{x}^{(n)}) || p_\theta(\boldsymbol{z}_2^{(n)} | \boldsymbol{\mu}_2)) \big] \\
& - D_{KL}(q_\phi(\boldsymbol{\mu}_2) || p_\theta(\boldsymbol{\mu}_2)). \\
= & \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\boldsymbol{z}_1^{(n)}, \boldsymbol{z}_2^{(n)} | \boldsymbol{x}^{(n)})} \big[\log p_\theta(\boldsymbol{x}^{(n)} | \boldsymbol{z}_1^{(n)}, \boldsymbol{z}_2^{(n)}) \big] \\
& - \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\boldsymbol{z}_2^{(n)} | \boldsymbol{x}^{(n)})} \big[D_{KL}(q_\phi(\boldsymbol{z}_1^{(n)} | \boldsymbol{x}^{(n)}, \boldsymbol{z}_2^{(n)}) || p_\theta(\boldsymbol{z}_1^{(n)})) \big] \\
& - \sum_{n=1}^{N} D_{KL}(q_\phi(\boldsymbol{z}_2^{(n)} | \boldsymbol{x}^{(n)}) || p_\theta(\boldsymbol{z}_2^{(n)} | \tilde{\boldsymbol{\mu}}_2)) - \frac{J}{2} \frac{\sigma^2_{\tilde{\boldsymbol{\mu}}_2}}{\sigma^2_{\boldsymbol{z}_2}} \\
& + \frac{1}{2} \sum_{j=1}^{J} (1 + \log \sigma^2_{\tilde{\boldsymbol{\mu}}_2}) + \frac{1}{2} \log 2\pi + \log p_\theta(\tilde{\boldsymbol{\mu}}_2) \\
= & \sum_{n=1}^{N} (\mathcal{L}(\theta, \phi; \boldsymbol{x}^{(n)} | \tilde{\boldsymbol{\mu}}_2) - \frac{J}{2} \frac{\sigma^2_{\tilde{\boldsymbol{\mu}}_2}}{\sigma^2_{\boldsymbol{z}_2}}) + \frac{1}{2} \sum_{j=1}^{J} (1 + \log \sigma^2_{\tilde{\boldsymbol{\mu}}_2}) + \frac{1}{2} \log 2\pi + \log p_\theta(\tilde{\boldsymbol{\mu}}_2) \\
= & \sum_{n=1}^{N} \mathcal{L}(\theta, \phi; \boldsymbol{x}^{(n)} | \tilde{\boldsymbol{\mu}}_2) + \log p_\theta(\tilde{\boldsymbol{\mu}}_2) + const
\end{aligned}
$$

B. Derivation of the Inferred S-Vector

As described in Section 2.2, inference of the s-vector $\boldsymbol{\mu}_2$ of an unseen utterance $\tilde{\boldsymbol{X}} = \{\tilde{\boldsymbol{x}}^{(n)}\}_{n=1}^{\tilde{N}}$ is cast as an approximated maximum a posterior estimation problem, which uses the conditional segment variational lower bound, $\mathcal{L}(\theta, \phi; \tilde{\boldsymbol{x}}^{(n)} | \boldsymbol{\mu}_2)$, to approximate the conditional likelihood of a segment, $\log p_\theta(\tilde{\boldsymbol{x}}^{(n)} | \boldsymbol{\mu}_2)$. Let J be the dimensionality of $\boldsymbol{z}_2$. Let $\hat{\boldsymbol{\mu}}_{\boldsymbol{z}_2}^{(n)}$ denote the variational mean of $\boldsymbol{z}_2$ evaluated at $\boldsymbol{x}^{(n)}$, and let $\mu_{2,j}$ and $\hat{\mu}_{\boldsymbol{z}_2,j}^{(n)}$ denote the j-th element of these vectors. The optimal $\boldsymbol{\mu}_2^*$ can be derived as follows:

$$
\begin{aligned}
\boldsymbol{\mu}_2^* = & \operatorname*{argmax}_{\boldsymbol{\mu}_2} \sum_{n=1}^{\tilde{N}} \mathcal{L}(\theta, \phi; \tilde{\boldsymbol{x}}^{(n)} | \boldsymbol{\mu}_2) + \log p_\theta(\boldsymbol{\mu}_2) \\
= & \operatorname*{argmax}_{\boldsymbol{\mu}_2} \sum_{n=1}^{\tilde{N}} -D_{KL}(q_\phi(\boldsymbol{z}_2^{(n)} | \tilde{x}^{(n)}) || p_\theta(\boldsymbol{z}_2^{(n)} | \boldsymbol{\mu}_2)) + \log p_\theta(\boldsymbol{\mu}_2) \\
= & \operatorname*{argmax}_{\boldsymbol{\mu}_2} \sum_{n=1}^{\tilde{N}} \sum_{j=1}^{J} \frac{-(\hat{\mu}_{\boldsymbol{z}_2,j}^{(n)} - \mu_{2,j})^2}{\sigma^2_{\boldsymbol{z}_2}} + \sum_{j=1}^{J} \frac{-(\mu_{2,j} - 0)^2}{\sigma^2_{\boldsymbol{\mu}_2}} \\
= & \operatorname*{argmax}_{\boldsymbol{\mu}_2} f(\boldsymbol{\mu}_2),
\end{aligned}
$$

where $f(\cdot)$ is a concave quadratic function that has only one maximum point. We then have:

$$\frac{\partial f(\boldsymbol{\mu}_2)}{\partial \boldsymbol{\mu}_2}\Big|_{\boldsymbol{\mu}_2=\tilde{\boldsymbol{\mu}}_2^*} = 0$$

$$\boldsymbol{\mu}_2^* = \frac{\sum_{n=1}^{\tilde{N}} \hat{\mu}_{\boldsymbol{z}_2,j}^{(n)}}{\tilde{N} + \sigma_{\boldsymbol{z}_2}^2/\sigma_{\boldsymbol{\mu}_2}^2}$$

C. FHVAE Model and Training Configurations

For the Seq2Seq-FHVAE model, each $LSTM$ network consists of one layer with 256 hidden units, while each MLP network is one layer with the output dimension equal to the variable whose mean or log variance the MLP parameterizes, and variances $\sigma_{\boldsymbol{z}_1}^2 = \sigma_{\boldsymbol{\mu}_2}^2 = 1$, $\sigma_{\boldsymbol{z}_2}^2 = 0.25$. We experiment with various dimensions for the latent variable $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$. All models were trained with stochastic gradient descent using a mini-batch size of 256 to minimize the negative discriminative segment variational lower bound plus an $L2$-regularization with weight 10^{-4}. The Adam [21] optimizer is used with $\beta_1 = 0.95$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and initial learning rate of 10^{-3}. Training continues for 500 epochs unless the segment variational lower bound on the development set does not improve for 50 epochs. The $\boldsymbol{\mu}_2$ for the sequences in the development set and the test set is estimated using the closed form solution in Section 2.2.

D. Comparison of Seq2Seq-FHVAE and Alternative Architectures

Here we study the performance of our proposed architecture by replacing the LSTM module with three baseline architectures: a fully-connected feed-forward network (FC), a vanilla recurrent neural network (RNN), and a gated recurrent neural network (GRU) [3]. All the models have one hidden layer with 16 dimensions for both $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, and are trained with $\alpha = 0$. For the FC model, the entire segment is flattened and feed to the fully-connected layers; therefore the temporal structure is simply ignored.

Table 4 shows the segment variational lower bound on the TIMIT test set. We can see that the recurrent models (RNN, GRU, LSTM) outperform the feed-forward model using fewer parameters, which demonstrates the importance of considering the temporal structure within a segment. Figure 6 shows the reconstruction results using the FC model and the LSTM model. The LSTM model reconstructs sharper images that preserves more speech detail, and, in particular, presents superior high frequency harmonic structure that does the FC model, as highlighted in the red boxes.

Table 4: TIMIT test set segment variational lower bound results on different model architectures.

Models	#Hidden Units	#Params	$\mathcal{L}(\theta, \phi; \boldsymbol{x}^{(n)})$
FC	512	3.3M	-348.63
RNN	256	0.3M	-261.19
GRU	256	0.8M	-158.42
LSTM	256	1.1M	**-143.80**

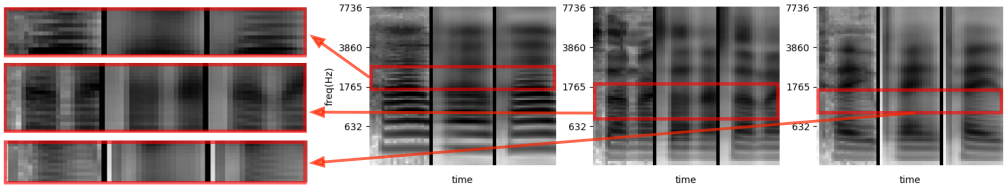


Figure 6: Three examples from different speakers. Within each example, from left to right are 1) the original segment, 2) FC reconstructed segment, and 3) LSTM reconstructed segment. The leftmost images show expanded views of the higher frequency harmonic structure (horizontal dark bands) of the spectrogram suggesting that the LSTM reconstruction is superior to the FC model.

E. Transformation of Speaker and Noise Conditions

Figure 7 shows the zoomed-in version of the left part in Figure 4, from which we can observe the harmonic patterns more clearly. In Figure 8, we illustrate the results of the same experiments, but use the model trained on the Aurora-4 corpus instead. In particular, we sample two speakers, 441 and 443, from the test set and choose four noise conditions: clean, car, babble, and restaurant, without the microphone channel effect. Furthermore, since the noise is artificially added to each clean utterance in the test set, we can actually choose the corresponding segment in different noise conditions for a given speaker. Same eight examples are used in both block 'A' and block 'B', which results in 64 combinations of latent segment variables and latent sequence variables in total. It can be observed that the latent sequence variables capture not only the speaker information, but also the noise information, which are both sequence-level attributes. Therefore, when modifying the latent sequence variables, we can not only transform speaker identities, but also carry out denoising or noise corruption. Moreover, the disentanglement is evident for both the model trained without discriminative training ($\alpha = 0$) and the model trained with discriminative training ($\alpha = 10$).

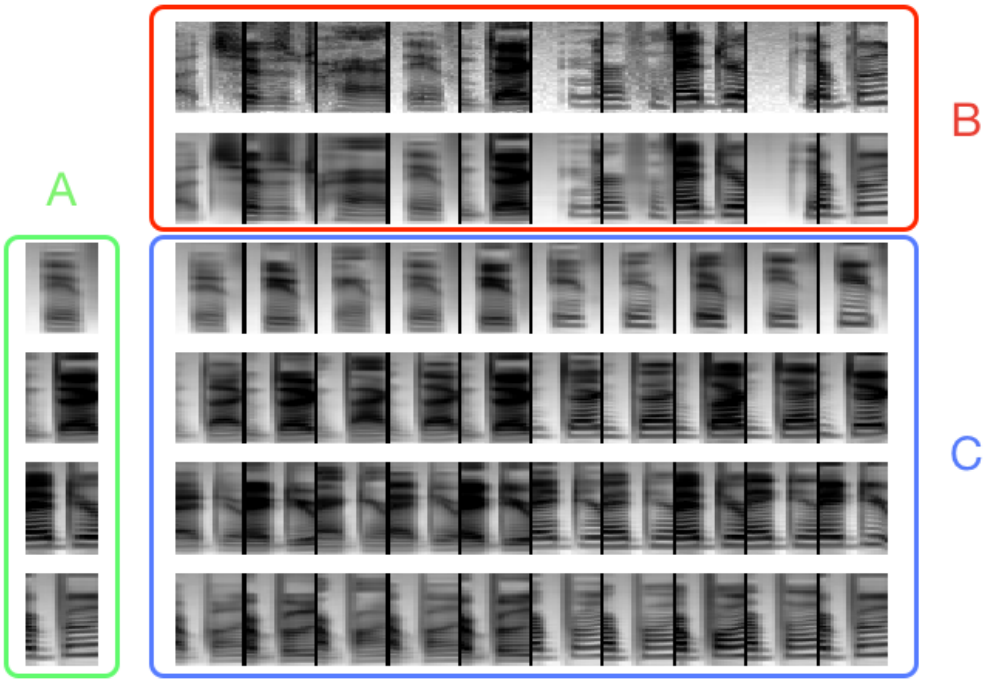


Figure 7: Examples generated by varying different latent variables of a FHVAE model trained with $\alpha = 10$ on TIMIT dataset. The green block 'A' contains four reconstructed examples. The red block 'B' contains ten original examples on the first row and the corresponding reconstructed examples on the second row. The entry on the i-th row and the j-th column in the blue block 'C' is the reconstructed example using the latent segment variable $\boldsymbol{z}_1$ of the i-th row from the block 'A' and the latent sequence variable $\boldsymbol{z}_2$ of the j-th column from the block 'B.'

In addition to transforming a single segment, one may also be interested in transforming a target sequence $\boldsymbol{X}_{tar}$ to be of a different speaker or a different noise condition of a reference sequence $\boldsymbol{X}_{ref}$. Mathematically, it means mapping the distribution of the latent sequence variable from that of $\boldsymbol{X}_{tar}$ to that of $\boldsymbol{X}_{ref}$. Since the distributions are both Gaussian with the same covariance matrices, centered at their own s-vectors, $\boldsymbol{\mu}_{2,tar}$ and $\boldsymbol{\mu}_{2,ref}$, a simple solution is to shift the latent sequence variable by the s-vector difference $\Delta\boldsymbol{\mu}_2 = \boldsymbol{\mu}_{2,ref} - \boldsymbol{\mu}_{2,tar}$. Therefore, we transform a target utterance given a reference utterance by shifting the $\boldsymbol{z}_2$ of each segment from the target utterance by $\Delta\boldsymbol{\mu}_2$, and then decode-and-concatenate each segment using the unmodified $\boldsymbol{z}_1$ and the modified $\boldsymbol{z}_2$. Figure 1, 5, 9, and 10 shows examples of modifying entire utterances, which achieves voice conversion and denoising respectively.

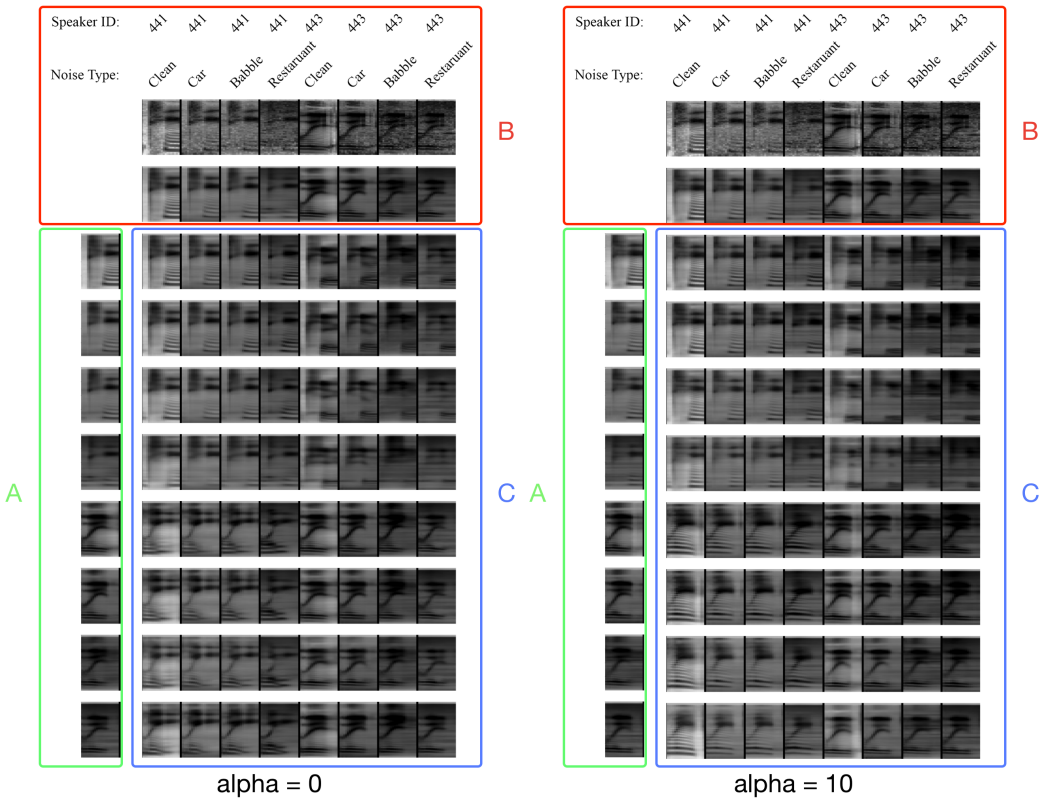


Figure 8: Examples generated by varying z_1 and z_2 of two FHVAE models trained with $\alpha = 0$ and $\alpha = 10$ respectively on Aurora-4 dataset. The green block 'A' and the red block 'B' contains the same eight examples from the test set. In block 'B,' original examples are shown on the first row and the corresponding reconstructed examples are shown on the second row. The entry on the i-th row and the j-th column in the blue block 'C' is the reconstructed example using the latent segment variable $\boldsymbol{z}_1$ of the i-th row from the block 'A' and the latent sequence variable $\boldsymbol{z}_2$ of the j-th column from the block 'B.'

F. More Details about the Speaker Verification Experiments

Verification performance is reported in terms of equal error rate (EER), where the false rejection rate equals the false acceptance rate. For our baseline system, we use the i-vectors [6] provided by Kaldi [34], which are extracted using Mel-frequency cepstral coefficients (MFCCs), plus delta and delta-delta after voice activity detection (VAD). A full-covariance gender-independent UBM with 2048 mixtures was trained on the training set and the i-vector dimensionality is tuned on the development set. The verification pairs were created from the test set as target/non-target. There are in total 24 speakers and 18,336 pairs for testing. For all the Seq2Seq-FHVAE model, $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ have the same dimension, and we use the closed form solution of the inferred s-vector as mentioned in Section 2.2 to represent each utterance for verification.

G. More Details about the Domain Invariant ASR Experiments

The Gaussian mixture model-hidden Markov models (GMM-HMM) systems are built first to generate the senone (tied triphone HMM state) alignments for the later neural network acoustic model training, which replaces the GMM acoustic model. In both tasks (TIMIT and Aurora-4), the GMM-HMM system is built with Kaldi [34] using standard recipes. We use the LSTM [13] for the acoustic model in our hybrid DNN-HMM system, which are implemented using the CNTK [43] toolkit. Our training recipe follows [45]. The baseline uses 80-dimensional FBank features as input. The model has 3 LSTM-projection layers [37], where each layer has 1024 cells and the output is projected to a

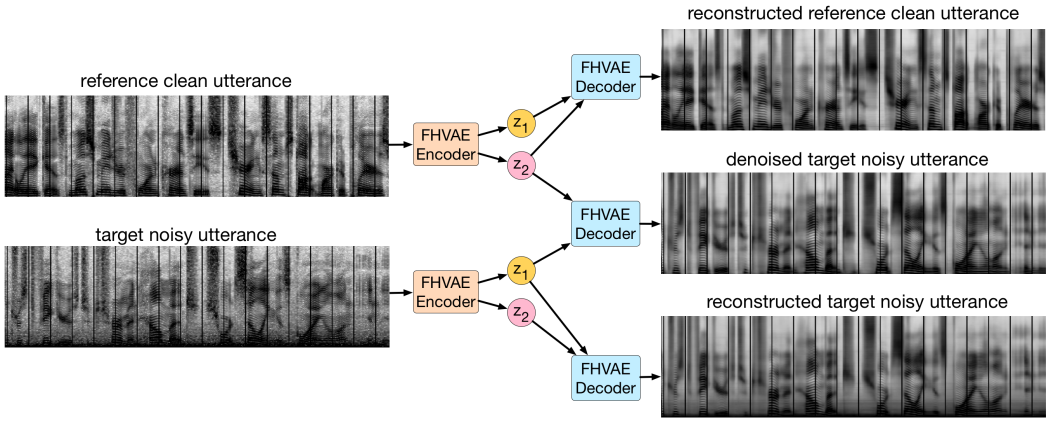


Figure 9: FHVAE ($\alpha = 0$) decoding results of three combinations of *latent segment variables* z_1 and *latent sequence variables* z_2 from one clean utterance (top-left) and one utterance with car noise (bottom-left) in Aurora-4. By replacing z_2 of a noisy utterance with z_2 of a clean utterance, an FHVAE decodes a denoised utterance (middle-right) that preserves the linguistic content. Audio samples are available at `https://youtu.be/pOP2DVZWRjM`.

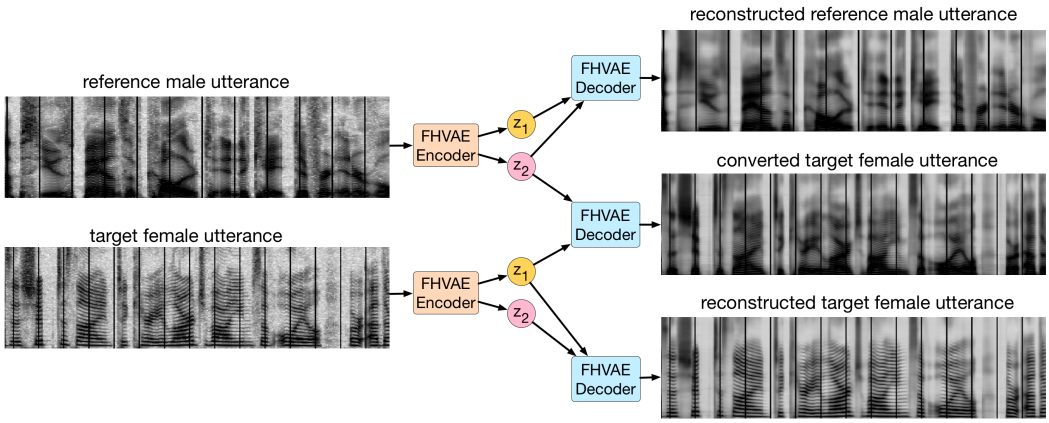


Figure 10: FHVAE ($\alpha = 0$) decoding results of three combinations of *latent segment variables* z_1 and *latent sequence variables* z_2 from one female-speaker utterance (top-left) and one male-speaker utterance (bottom-left) in Aurora-4. By replacing z_2 of a female-speaker utterance with z_2 of a male-speaker utterance, an FHVAE decodes a voice-converted utterance (middle-right) that preserves the linguistic content. Audio samples are available at `https://youtu.be/Rurj2ByNRs8`.

512 dimensional space. The truncated BPTT is used to train the LSTM that unrolls 20 frames; 40 utterances are processed in parallel to form a mini-batch. For the Seq2Seq-FHVAE model, we use the same configuration as the one that achieved the best result on the speaker verification task: both $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ are 32 dimensional, and the weight $\alpha = 10$ for discriminative training. For the VAE model, the dimension of the latent variable $\boldsymbol{z}$ is 64, and the number of hidden units of the LSTM encoder is 512. We doubled both the latent variable dimension and the number of hidden units for the encoder compared to the FHVAE model because the VAE model only has one set of latent variables and one encoder. Therefore, both the FHVAE and VAE models would have a comparable number of parameters as well as latent space dimensionalities.

H. FHVAE Latent Space Traversal

In this section, we present a qualitative analysis of traversing a single latent sequence variable or latent segment variable over the range $[-3, 3]$, while keeping the remaining latent variables fixed.

Each row corresponds to a different seed $(\boldsymbol{z}_1, \boldsymbol{z}_2)$ pair, inferred from some seed segment randomly drawn from the test set. The leftmost column in each figure shows the seed segments for each row. We use the same five seed segments for traversing each latent variable. The FHVAE model is trained on TIMIT with $\alpha = 0$, and a 200 dimensional log-magnitude spectrum is used for frame feature representations.

Figures 11 and 12 show examples of traversing four different latent segment variables, $\boldsymbol{z}_1$, while keeping the latent sequence variables fixed. It can be observed that these latent segment variables encode the information of segment-level attributes in speech data, such as rising/falling F2, back vowel/front vowel, vowel/fricative, and closure/non-closure.

In contrast, Figures 13 and 14 illustrate examples for traversing four different latent sequence variables, $\boldsymbol{z}_2$, while keeping the latent segment variables fixed. It can be seen the spectral contour, temporal position, and relative frequency-axis position of formants remain almost intact when traversing these latent sequence variables. The attributes being changed when traversing these latent sequence variables are more related to sequence-level attributes, such as harmonic patterns (F0), volume, offsets of formant frequencies. The results again demonstrate the ability of our proposed FHVAE to not only learn disentangled representations, but also enable interpretation of the information captured by different sets of latent variables.

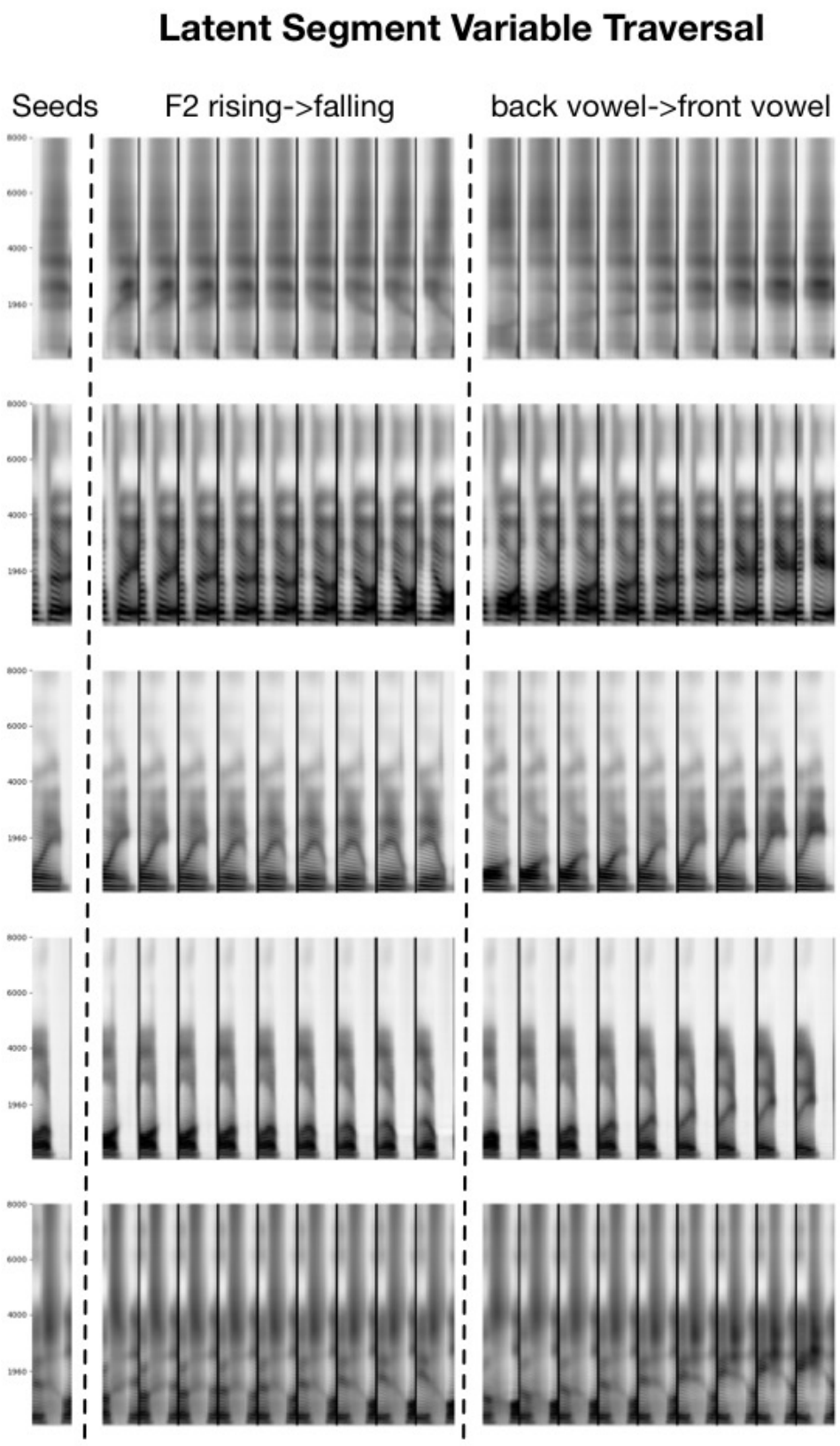


Figure 11: Traversing two different latent segment variables with five seed segments from the TIMIT test set using an FHVAE model trained on TIMIT with $\alpha = 0$.

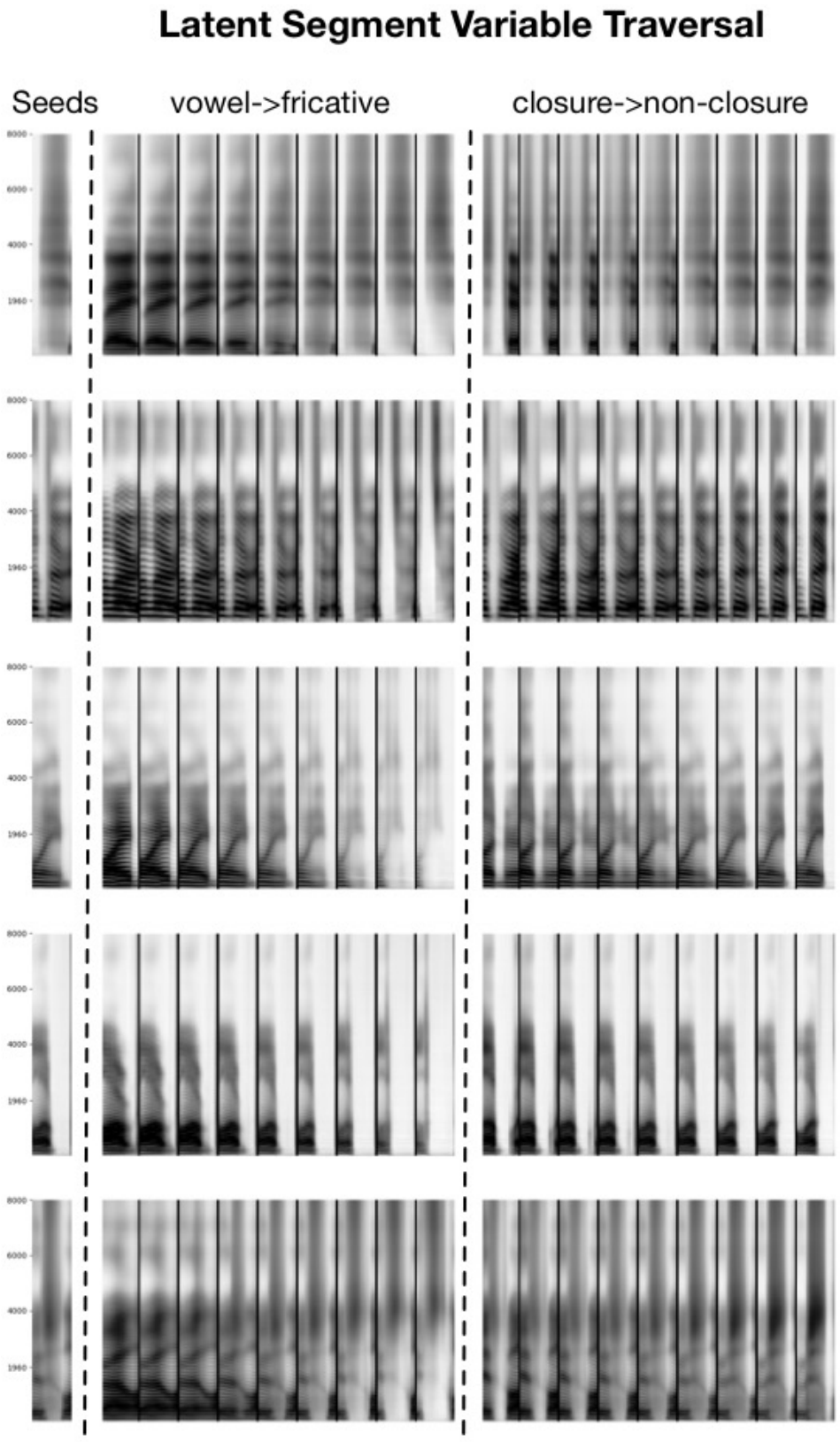


Figure 12: Traversing another two different latent segment variables with five seed segments from the TIMIT test set using an FHVAE model trained on TIMIT with $\alpha = 0$.

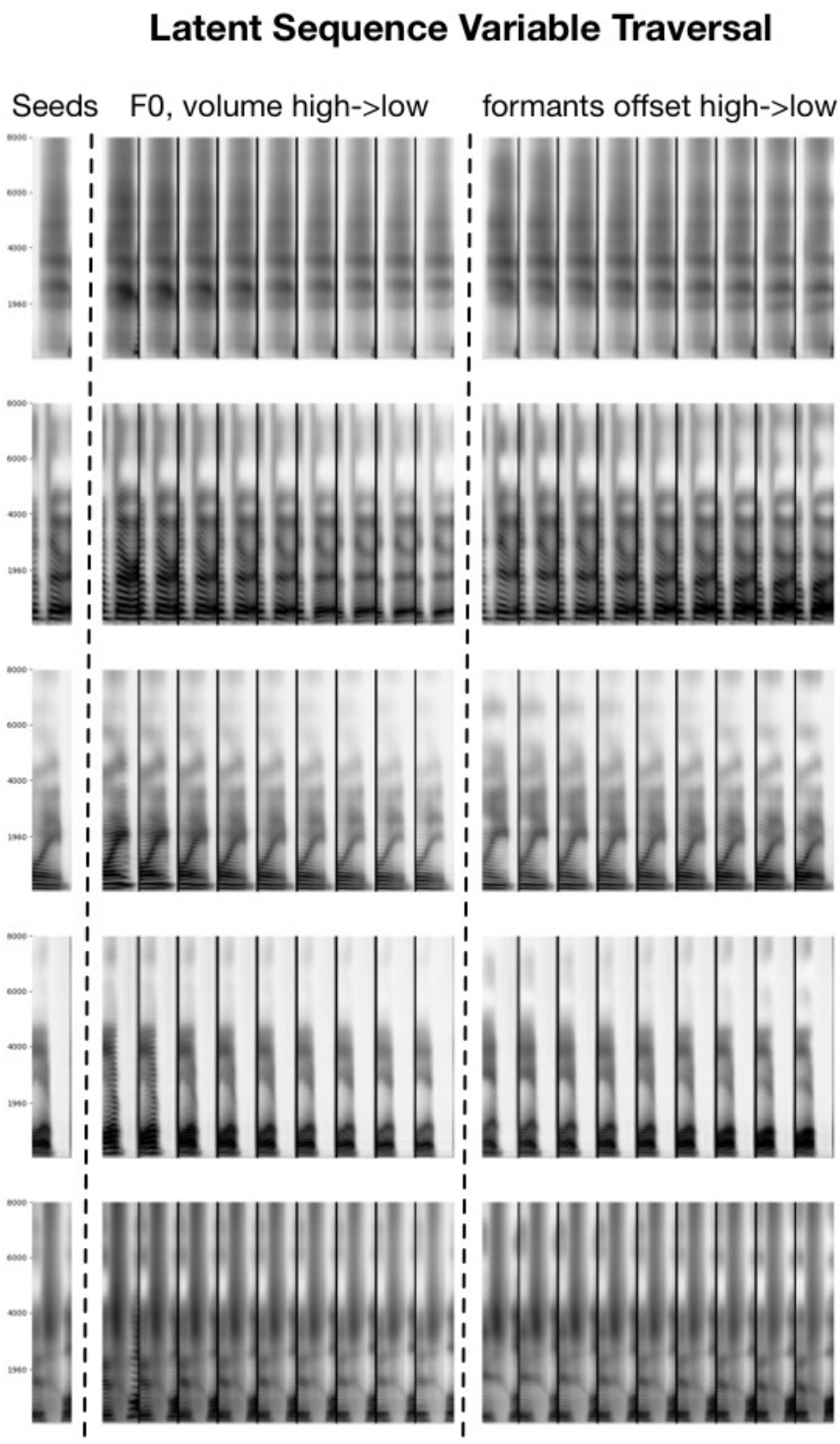


Figure 13: Traversing two different latent sequence variables with five seed segments from the TIMIT test set using an FHVAE model trained on TIMIT with $\alpha = 0$.

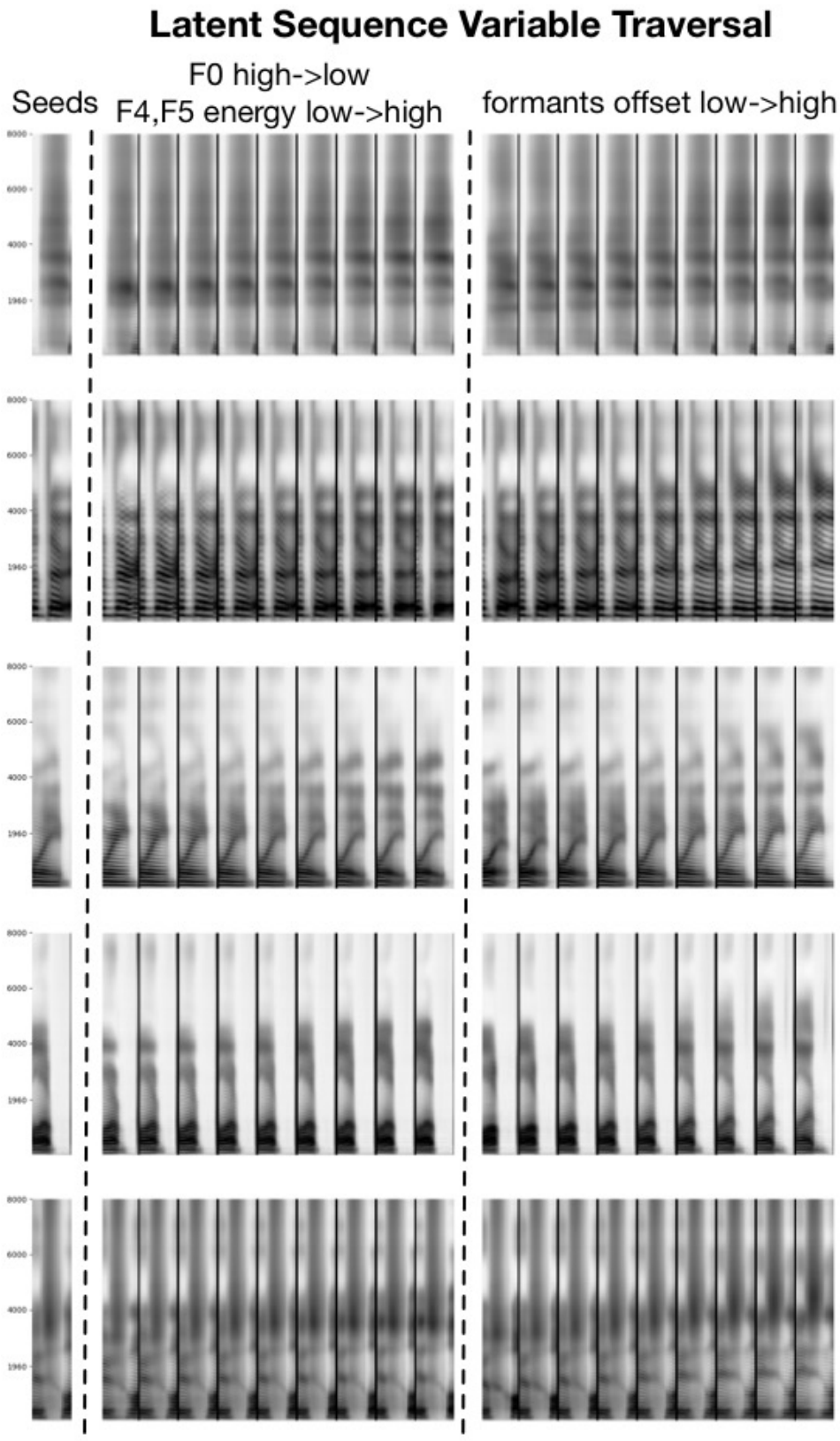


Figure 14: Traversing another two different latent sequence variables with five seed segments from the TIMIT test set using an FHVAE model trained on TIMIT with $\alpha = 0$.