INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Unsupervised Learning of Discrete Latent Representations with Data-Adaptive Dimensionality from Continuous Speech Streams

Shun Takahashi, Sakriani Sakti

Japan Advanced Institute of Science and Technology, Japan
{shun-tak,ssakti}@jaist.ac.jp

Abstract

This work presents a novel deep generative model for unsupervised learning of sparse binary feature representations with data-adaptive dimensionality directly from continuous speech streams. Sharing the critical assumption of unbounded latent dimensionality with previously proposed Bayesian nonparametric approaches, our proposed model can capture the much richer, non-Markovian dependencies between its latent representations. The present work focuses on an investigation of our proposed model's performance in learning linguistically meaningful representations under challenging, realistic scenarios. We train our model with highly speaker-imbalanced datasets and evaluate it on the ABX phone discriminability test. Our model achieves a promising, competitive performance to the state-of-the-art model, despite its huge disadvantage: limited or no access to speaker information during training.

Index Terms: unsupervised learning, representation learning, zero resource, deep non-parametric Bayes, deep generative model, speech recognition

1. Introduction

Unsupervised learning of linguistically meaningful representations directly from speech signals is of fundamental interest to inclusive spoken language processing technology for the world's languages, as well as for computational approaches to developmental linguistic/cognitive studies [1].

Early representative models for unsupervised discrete representation learning for speech were based on Bayesian nonparametric (BNP) approaches [2, 3, 4], where, the latent representations and their number/dimensionality are inferred from the data. However, the latent structures in the previously proposed BNP models were too restrictive for modeling speech: e.g., only short-term Markovian dependencies or none at all. They were difficult to apply to larger and/or realistic datasets due to the restrictive assumptions and slow inference.

In recent years, a number of models have been proposed through the *ZeroSpeech Challenges*, especially from 2015 to 2020 [5, 6, 7, 8]. Among them, state-of-the-art approaches are characterized by the use of *Vector-Quantization* (VQ). VQ is the online clustering of neural networks' internal representations. originally proposed as a bottleneck of a discrete autoencoder [9] and later optimized for unsupervised speech representation learning under zero-resource scenarios [10, 11]. Unlike earlier BNP approaches, the number/dimensionality of VQ's latent representations must be set as hyperparameters. Furthermore, since it is a clustering approach, its inference is not based on the temporal structure of speech, leading to the need for post-hoc sequential modeling on learned representations [12].

Our work provides the following three main contributions. (1) We propose a highly flexible deep generative model that inherits the key aspect of the previously proposed BNP approaches, which has been missing in such state-of-the-art approaches as VQ-based ones: inference of the dimensionality of the latent representations. (2) We address the intractability of posterior inference by revisiting a neural-based sequential Monte Carlo method that has received little attention compared to other variational approaches. (3) By extensive comparison with the state-of-the-art VQ-based model, we show that our model achieves a promising, competitive performance.

2. Proposed Approach

In this work, we approach unsupervised discrete representation learning as posterior inference of a deep generative model. Our model assumes latent representations, which underlie speech streams, have sparse binary features with dynamic dimensionality. It is trained with a recognition model (encoder) in an autoencoder-like fashion to learn and infer the latent representations from continuous speech streams.

Fig. (1) shows examples where our model reconstructed log-Mel spectrograms with its learned discrete representations.

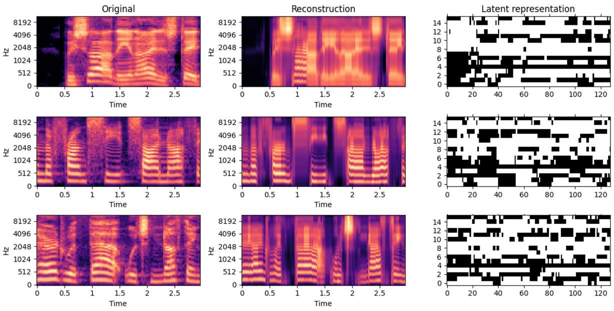


Figure 1: *Original log-Mel Spectrograms, their reconstructions and latent representations generated by our model, trained with* `LibriLight` *(120h) with no speaker information. The sample data were obtained from the validation data (*`LibriSpeech dev-clean`*). In the latent representation, black and white blocks represent* 1 *and* 0*. The latent dimension is clipped at 15, the largest active dimension.*

2.1. Generative Model

We define a sequential generative model parameterized by deep neural networks for observation sequence $\mathbf{x}_{1:T} \in \mathbb{R}^D$ and latent

10.21437/Interspeech.2023-1321

K-dimensional binary feature vector sequence $\mathbf{z}_{1:T} \in \{0,1\}^K$:

$$p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}) = p_\theta(\mathbf{x}_1, \mathbf{z}_1) \prod_{t=2}^{T} p_\theta(\mathbf{x}_t, \mathbf{z}_t \mid \mathbf{z}_{1:t-1})$$

$$= p_\theta(\mathbf{x}_1 \mid \mathbf{z}_1) p_\theta(\mathbf{z}_1) \times \prod_{t=2}^{T} p_\theta(\mathbf{x}_t \mid \mathbf{z}_t) p_\theta(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}) \quad (1)$$

where current observation $\mathbf{x}_t$ is assumed to be independent of its past states given its current latent variable $\mathbf{z}_t$ and $\mathbf{z}_t$ to be dependent on its entire past trajectory which is represented by the internal states of deterministic recurrent neural networks.

Our proposed model assumes uncertainty in $\mathbf{z}$'s dimensionality $K \in \mathbb{N}_0$ and probability of each feature's activation $\nu_i \in [0,1]$ and i for $0,1,2,\cdots,K$. As prior knowledge, inspired by *distinctive feature* established in the context of phonological analysis [13], our model assumes $\mathbf{z}_{1:T}$ be low-dimensional and sparse. Based on such an assumption, the latent features' dimensionality and probabilities are factorized to another set of latent variables. The following is our further factorized predictive prior at time step t:

$$p_\theta(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}) = \left\{ \prod_{i=0}^{K} p_\theta(z_{t,i} \mid \pi_{t,i}, K_t, \mathbf{z}_{1:t-1}) \right.$$

$$\left. \times\, p_\theta(\nu_{t,i} \mid K_t, \mathbf{z}_{1:t-1}) \right\} \times p_\theta(K_t \mid \mathbf{z}_{1:t-1}) \quad (2)$$

where $\pi_{t,i}$ is a probability weighted by the stick-breaking process [14]: $\pi_{t,i} = \prod_{j=0}^{i} \nu_j$. We use $\pi_{t,i}$ to denote stick-breaking weights throughout the paper. For the initial state at $t = 1$, the dependency on $\mathbf{z}_{1:t-1}$ is omitted, and hyperparameters are set for the priors on π and K. In the following, to avoid notational clutter, we represent $\{z_{t,i}, \pi_{t,i}, \nu_{t,i}, K_t\}$ by $\mathbf{z_t}$.

Our proposed model can be viewed as a dynamic variant of an Indian Buffet process (IBP) [15]. However, for dimensionality, instead of using implicit prior and biased posterior approximation [16], we propose a learnable prior and approximate posterior with unbounded support, which allows for data-adaptive dimensionality.

2.2. Our Choice of Posterior Inference Method

We consider online inference for our model's posterior: $p_\theta(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T})$. Although a variational approach is typically taken for deep generative models [17, 18], it requires approximate distribution to have differentiable sampling or score function-based gradient estimation [19] which suffers from high variance [20]. Such an approach is unsuitable for our model, primarily due to its discrete structured latent variables.

We propose using an alternative, importance sampling approach [21]. Such a Representative method is known as *Reweighted Wake-Sleep* (RWS) [22], where neural networks are trained as a *recognition model* to construct *adaptive proposal distribution* $q_\phi(\mathbf{z} \mid \mathbf{x})$ by optimizing expected forward KL divergence from the true posterior: $\mathrm{KL}[p_\theta(\mathbf{z} \mid \mathbf{x}) || q_\phi(\mathbf{z} \mid \mathbf{x})]$. Since our model is sequentially structured, we use an independently proposed variant of RWS, called *Neural-adaptive Sequential Monte Carlo* (NASMC) [23].

In this method, at each time step, we obtain N samples (called particles), which represent N possible latent states, from the proposal distribution parameterized by neural networks with parameters ϕ (recognition model): $\{\mathbf{z}_t^{(n)}\}_{n=\{1,2,\cdots,N\}} \sim q_\phi(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}^{A^{(n)}}, \mathbf{x}_{1:t})$, and calculate the following weights by evaluating $\mathbf{z}_t^{(n)}$ with the generative and recognition models:

$$w_t^{(n)} = \frac{p_\theta(\mathbf{x}_t, \mathbf{z}_t^{(n)} \mid \mathbf{z}_{1:t-1}^{A^{(n)}})}{q_\phi(\mathbf{z}_t^{(n)} \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t-1}^{A^{(n)}})}, \quad \hat{w}_t^{(n)} = \frac{w_t^{(n)}}{\sum_n w_t^{(n)}}, \quad (3)$$

where $\mathbf{z}_{1:t-1}^{A^{(n)}}$ are the ancestor latent states' trajectories $\{\mathbf{z}_{1:t-1}^{a_{1:t-1}^{(n)}}\}$. Each particle is resampled at each step with $a_l^n \sim \text{Categorical}(\hat{w}_l^{(n)}) \quad (l \in \{1,2,\cdots,t-1\})$: for subsequent step t, the particles with smaller weights are more likely to be replaced with those with larger weights. The idea, which resembles the probabilistic analog of beam search [24], allows for the practical inference of non-Markovian structures.

Using $\hat{w}_t^{(n)}$, we obtain online gradient estimates at time step t for $\nabla_\phi \mathrm{KL}[p(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}) || q_\phi(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T})]$ with which to update the recognition model' parameters ϕ:

$$\nabla_\phi \mathrm{KL}[p_\theta(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}) || q_\phi(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T})]$$

$$\simeq -\Sigma_t \Sigma_n \hat{w}_t^{(n)} \nabla_\phi \log q_\phi(\mathbf{z}_t^{(n)} \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t-1}^{A^{(n)}}). \quad (4)$$

Likewise, using the obtained particle weights, the generative model's parameters, θ, are updated with the following gradient estimates of Evidence Lower Bound (ELBO):

$$\nabla_\theta \mathrm{KL}[p_\theta(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}) || q_\phi(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T})]$$

$$\simeq \Sigma_t \Sigma_n \hat{w}_t^{(n)} \nabla_\theta \log p_\theta(\mathbf{x}_t, \mathbf{z}_t^{(n)} \mid \mathbf{z}_{1:t-1}^{A^{(n)}}). \quad (5)$$

2.3. Recognition Model

We propose the following proposal distribution which leverages the structure of our generative model:

$$q_\phi(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}) = q_\phi(\mathbf{z}_1 \mid \mathbf{x}_1) \prod_{t=2}^{T} q_\theta(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) \quad (6)$$

which is further factorized at each time step:

$$q_\phi(\mathbf{z}_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) = \left\{ \prod_{i=0}^{K} q_\phi(z_{t,i} \mid \pi_{t,i}, K_t, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) \right.$$

$$\left. \times\, q_\phi(\nu_{t,i} \mid K_t, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) \right\} q_\phi(K_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}). \quad (7)$$

Our proposal distribution resembles the predictive prior (2). However, one crucial difference is that it has access to the current observation in addition to the past ones $\mathbf{x}_{1:t}$ through internal states of deterministic recurrent neural networks.

2.4. Implementation Details

There are many choices for the architecture of neural networks, the probability distributions as well as how to parameterize them in our model. We specify a case for our model used for the following experiments. Fig. (2) illustrates an implementation of our model during both the training and inference phases. Here MLP and RNN refer to multi-layered perceptrons and recurrent neural networks. In the following, we present the specification of the distributions and how to parameterize them in the generative and recognition models. Note that we use the same variable notations in Fig. (2).

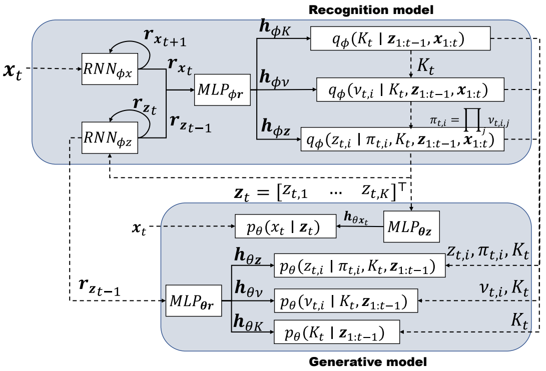

Recognition model
$\boldsymbol{x}_t$
$RNN_{\phi x}$
$\boldsymbol{r}_{\boldsymbol{x}_{t+1}}$
$\boldsymbol{r}_{\boldsymbol{x}_t}$
$MLP_{\phi \boldsymbol{r}}$
$RNN_{\phi z}$
$\boldsymbol{r}_{\boldsymbol{z}_t}$
$\boldsymbol{r}_{\boldsymbol{z}_{t-1}}$
$\boldsymbol{h}_{\phi K}$
$q_\phi(K_t \mid \boldsymbol{z}_{1:t-1}, \boldsymbol{x}_{1:t})$
K_t
$\boldsymbol{h}_{\phi \nu}$
$q_\phi(\nu_{t,i} \mid K_t, \boldsymbol{z}_{1:t-1}, \boldsymbol{x}_{1:t})$
$\pi_{t,i} = \prod_j \nu_{t,i,j}$
$\boldsymbol{h}_{\phi z}$
$q_\phi(z_{t,i} \mid \pi_{t,i}, K_t, \boldsymbol{z}_{1:t-1}, \boldsymbol{x}_{1:t})$
$\mathbf{z}_t = [z_{t,1} \quad \cdots \quad z_{t,K}]^\top$
$\boldsymbol{x}_t$
$p_\theta(x_t \mid \boldsymbol{z}_t)$
$\boldsymbol{h}_{\theta x_t}$
$MLP_{\theta \boldsymbol{z}}$
$\boldsymbol{h}_{\theta z}$
$p_\theta(z_{t,i} \mid \pi_{t,i}, K_t, \boldsymbol{z}_{1:t-1})$
$z_{t,i}, \pi_{t,i}, K_t$
$\boldsymbol{r}_{\boldsymbol{z}_{t-1}}$
$MLP_{\theta \boldsymbol{r}}$
$\boldsymbol{h}_{\theta \nu}$
$p_\theta(\nu_{t,i} \mid K_t, \boldsymbol{z}_{1:t-1})$
$\nu_{t,i}, K_t$
$\boldsymbol{h}_{\theta K}$
$p_\theta(K_t \mid \boldsymbol{z}_{1:t-1})$
K_t
Generative model

Figure 2: *An implementation of our proposed model during training and inference: the arrows represent input and output flows. Additionally, the solid/dashed ones represent where the gradients flow back/where they do not during training.*

2.4.1. Generative Model

We specified the distributions and their parameterization as:

$$p_\theta(K_t \mid \mathbf{z}_{1:t-1}) = \text{Geometric}(\sigma(\mathbf{w}_{\theta_K}^\top \mathbf{h}_{\theta K_t})), \tag{8}$$

$$p_\theta(\nu_{t,i} \mid K_t, \mathbf{z}_{1:t-1}) = \text{Beta}\left(\frac{\alpha - (i+1) * d}{K_t + 1}, 1 - d\right),$$
$$\text{where } \alpha = \gamma(\mathbf{w}_{\theta_\nu}^\top \mathbf{h}_{\theta \nu_t}) \text{ and } d = \sigma(\mathbf{w}_{\theta_K}^\top \mathbf{h}_{\theta \nu_t}), (\alpha > -d), \tag{9}$$

$$p_\theta(z_{t,i} \mid \pi_{t,i}, K_t, \mathbf{z}_{1:t-1}) = \text{Bernoulli}(\lambda_i),$$
$$\text{where } \lambda_i = \sigma\left(\mathbf{w}_{\theta_{\mathbf{z}} i}^\top \mathbf{h}_{\theta \mathbf{z}_t} + \text{logit}(\pi_{t,i})\right), \tag{10}$$

$$p_\theta(\mathbf{x}_t \mid \mathbf{z_t}) = \text{MultivariteNormal}(\mu_{\mathbf{x}}, \Sigma_{\mathbf{x}}),$$
$$\text{where } \mu_{\mathbf{x}} = \mathbf{W}_{\theta \mathbf{x}} \mathbf{h}_{\mathbf{x}_t} \text{and } \Sigma_{\mathbf{x}} = \gamma(\mathbf{W}_{\theta \mathbf{x}} \mathbf{h}_{\mathbf{x}_t}) \mathbf{I}_D \tag{11}$$

where $\gamma(y) = \log(1 - e^{-|y|}) + \max(y, 0)$ [25], $\sigma(y) = \text{sigmoid}(y)$, $\mathbf{I}_D$ is an D-dimensional identity matrix and the variables subscripted with i denotes to elements with index $i \leq K$ and other elements with $i > K$ are masked to 0.

In Eq. (8), a geometric prior is chosen for the dimensionality, as it puts large probability mass on the lower dimensions and spreads exponentially decreasing, thin masses to the larger dimensions with no bound. In Eq. (9), Pitman-Yor parameterization proposed in [14] was used, which allows the discount parameter d to control the decrease of probability weights $\pi_{t,i}$ in the stick-breaking process (10). The first parameter of $\text{Beta}(.,.)$ is dependent on K in such a way that growing dimensionality may induce sparsity in the resulting $\mathbf{z}_t$.

2.4.2. Recognition Model

Except for $q(K_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})$, all the other distributions follow the same parameterization as those of the generative model (2.4.1). For $q(K_t \mid \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})$, we propose using *discretized Gamma distribution* [26]. This is because other typical discrete distributions such as geometric and Poisson do not allow for flexible variance control. Other continuous distributions such as Gumbel may be used, but we leave the exploration of such alternatives as future work.

3. Experimental Setup

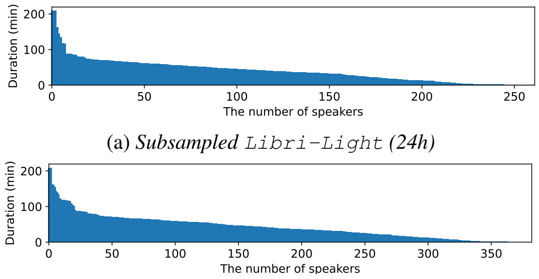
We experimentally evaluated our model's representation learning performance by comparing it with the state-of-the-art model. The *ABX phone discriminability test* [27, 28] with the cosine similarity-based measure was used as a metric.

3.1. Dataset Preparation

We use `Libri-Light` [29] dataset, which offers 600 to 60,000 hours of read English speech audio data. Although `Libri-Light` contains a massive amount of data, based on practical applications on zero-/low-resource languages for which the amount of available data is likely to be limited, we randomly sub-sampled 24 hours and 120 hours of data from the smallest 600-hour sub-dataset (`unlab-600`). The speaker statistics of our datasets are shown in Figs. (3a) and (3b). They are characterized by imbalanced speaker distributions, making representation learning challenging and realistic.

We segmented each speech sequence of the datasets into approximately minute-long, smaller sub-sequences. Then, using `librosa` (0.9.2) [30], we converted them into log-Mel magnitude spectrograms with the FFT window length of 2048, a hop length of 160, a window length of 400, and 80 Mel bands. We did not remove any silent parts from the data, since they only comprised a negligible proportion in the original dataset [31].

For the evaluation, we used both `test-clean` and `test-other`. The latter contains noisy data, so the inference is expected to be more difficult.



(a) *Subsampled* `Libri-Light` *(24h)*

(b) *Subsampled* `Libri-Light` *(120h)*

Figure 3: *Two speaker-imbalanced small- and medium-sized datasets we created for this experiment: we randomly subsampled (a) 24 hours and (b) 120 hours respectively from* `Libri-Light` *[29].*

3.2. Hyperparameter settings and other details

Model. Our model is implemented entirely on `Pytorch` (1.12.1) [32]. All the MLP's consisted of four layers, each containing 512 neurons, followed by layer normalization [33] and a ReLU activation function. For all the RNN's, *Independently Recurrent Neural Networks* (IndRNN) [34] was used with Layer normalization. Three layers of IndRNN's were stacked for the observation trajectory ($\mathbf{x}_{1:t}$), and a single layer for the latent trajectory ($\mathbf{z}_{1:t}$) with the number of neurons set to 512. The weights were initialized by their respective standard methods. IndRNN was chosen because it was stable throughout the training while other RNN variants, including LSTM and GRU, were susceptible to gradient explosion. The parameters of the priors at the initial time step ($t = 1$) were fixed as $p(K_1) = \text{Geometric}(0.1)$ and $p(\nu_{1,i}) = \text{Beta}(2/(K_1 + 1), 1)$.

Training and Inference. Leveraging the fact that our model's inference is independent of future observations, we trained the model in a multi-stream fashion, where only *frame-by-frame*, a batch of B speech sequences was incrementally loaded and fed into the model so that we could avoid loading the entire observation sequences. Instead, we increased the batch and the particle sizes with the saved memory space. The batch size was set to $B = 256$ and the particle size was set to $N = 10$. An Adam optimizer [35] was used for both the generative and recognition models with learning rates set to 0.0001 for both. The number of iterations was set to 2 million steps, but using `LibriLight dev-clean`, we monitored the models' convergence by ELBO averaged over randomly sampled consecutive time steps ($T = 5000$), batch size $B = 256$, and particle size $N = 10$, and stopped the training when ELBO converged. In the inference, the particle size was set to $N = 1024$, and at each time step, a particle with the largest weight was obtained as an approximate MAP estimate. The resulting sequences were used for the ABX evaluation. A single NVIDIA A-100 GPU was used for both the model's training and inference.

3.3. Baseline and Topline Models

We compared our model to Vector Quantized Contrastive Predictive Coding (VQ-CPC) with a previously proposed implementation and hyperparameter configuration [11]. VQ-CPC is a deterministic discrete representation learning model, which is currently considered as the state-of-the-art [11].

The latent discrete representations of VQ-CPC are vector embeddings called a codebook. Their size and dimension ($|S|$ and K) must be set as hyperparameters. We consider four models with $|S| = 64, 128, 256, 512$ and $K = 64$.

As our baseline and topline models, we considered two versions of VQ-CPC with different training methods: those trained with random negative sampling and random training data sampling (*VQ-CPC-RS*); those trained with within-speaker negative sampling and speaker-balanced training data sampling (*VQ-CPC-WS*). The key difference between them is that, during training, while *VQ-CPC-RS* does not use any speaker information, *VQ-CPC-WS* does exploit speaker identity to sample and structure the input data in a way that induces speaker-invariance. This sampling method assumes speaker information in training data to be available. In total, eight models were run.

To make the comparison fairer, we trained two different versions of our model similar to the above: one with random training data sampling (denoted as *random*) and another with speaker-balanced training data sampling (denoted as *balanced*). Although our speaker information utilization was more limited than the above, we expected the latter to mitigate the bias from speaker imbalance.

3.4. Experimental Results and Discussion

Tables (1) and (2) show the ABX results of all the models trained on `LibriLight` (24h) and `LibriLight` (120h).

All the VQ-CPC's including *VQ-CPC-WS* and *VQ-CPC-RS* improved the representation quality consistently with the increase of the codebook size. Notably, VQ-CPC models degraded their performance quite drastically when the speaker information was not provided during training (*VQ-CPC-RS*). These results are consistent with the observations in [11]. The performance of VQ-CPC is significantly dependent on the availability of speaker information in training.

Despite no (or limited) access to speaker information, our model, including both *random* and *balanced*, significantly outperformed the *VQ-CPC-RS* group, which, we emphasize, is trained under the similar condition to our model. Speaker-balanced sampling alone could not induce speaker-invariance. The inconsistent performances between *clean-/other-within* and *clean-/other-across* also suggest that our model could not discard correlated speaker-specific features, arguably due to the lack of inductive bias for speaker-invariance as well.

Table 1: *Cosine similarity-based ABX evaluation results for models trained with* `LibriLight` *(24h).*

		Cosine similarity-based ABX error rates (%)			
		clean-within	*clean-across*	*other-within*	*other-across*
Topline (*VQ-CPC-WS*)	$\lvert S\rvert = 512$	12.75	16.87	15.84	23.66
	$\lvert S\rvert = 256$	13.06	16.98	16.41	23.92
	$\lvert S\rvert = 128$	13.51	17.30	16.82	23.62
	$\lvert S\rvert = 64$	15.13	19.60	18.44	25.94
Baseline (*VQ-CPC-RS*)	$\lvert S\rvert = 512$	33.97	42.37	36.02	44.20
	$\lvert S\rvert = 256$	34.44	42.98	36.79	44.72
	$\lvert S\rvert = 128$	39.24	48.12	41.47	51.22
	$\lvert S\rvert = 64$	43.66	51.38	43.44	59.22
Ours	*Random*	17.95	28.30	22.35	36.13
	Balanced	17.81	28.79	22.68	37.10

Table 2: *Cosine similarity-based ABX evaluation results for models trained with* `LibriLight` *(120h).*

		Cosine similarity-based ABX error rates (%)			
		clean-within	*clean-across*	*other-within*	*other-across*
Topline (*VQ-CPC-WS*)	$\lvert S\rvert = 512$	9.904	12.80	12.98	19.92
	$\lvert S\rvert = 256$	10.28	13.45	13.41	20.25
	$\lvert S\rvert = 128$	11.88	15.44	14.92	21.88
	$\lvert S\rvert = 64$	13.78	17.41	16.82	23.42
Baseline (*VQ-CPC-RS*)	$\lvert S\rvert = 512$	32.52	40.08	33.75	40.80
	$\lvert S\rvert = 256$	33.37	41.48	34.71	43.21
	$\lvert S\rvert = 128$	38.02	47.15	40.41	49.32
	$\lvert S\rvert = 64$	43.16	51.16	43.89	51.06
Ours	*Random*	18.58	28.78	23.33	37.00
	Balanced	18.71	28.98	23.38	35.91

In addition to the absence of speaker information, the large margins to the topline group may be accounted for by the difference between the two models' representations. The representations of VQ-CPC are dense real vectors with their values learned from the data. In contrast, those of our model are sparsity-enforced binary vectors that take the values either 0 or 1. This difference is not considered in the current ABX metrics.

4. Conclusions and Future Work

We presented a novel deep generative model for learning discrete representations with a data-adaptive dimensionality and introduced an efficient sequential Monte Carlo method for its online inference of latent representations. Since this work is especially focused on our model's performance on unsupervised representation learning under realistic scenarios, we trained our model with two speaker-imbalanced small-sized and mid-sized datasets. We evaluated it on an ABX discriminability test to compare with the state-of-the-art model. The experimental results suggest our model is promising yet clearly missing inductive bias for speaker-invariance, which the state-of-the-art model successfully supplements by its training method.

In future work, we aim to explore effective methods to incorporate speaker information in the training of our model. Furthermore, we will investigate other aspects of our model, such as the interpretability of the learned representations and application to spoken language modeling directly on speech [36, 12].

5. Acknowledgement

Part of this work is supported by JSPS KAKENHI Grant Numbers JP21H05054 and JP21H03467.

6. References

[1] E. Dupoux, "Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner," *Cognition*, vol. 173, pp. 43–59, Apr. 2018.

[2] C.-y. Lee and J. Glass, "A nonparametric Bayesian approach to acoustic model discovery," in *Proceedings of the 50th annual meeting of the ACL (volume 1: Long papers)*. ACL, Jul. 2012, pp. 40–49.

[3] M. Heck, S. Sakti, and S. Nakamura, "Unsupervised linear discriminant analysis for supporting DPGMM clustering in the zero resource scenario," *Procedia Computer Science*, vol. 81, pp. 73–79, 2016.

[4] L. Ondel, P. Godard, L. Besacier *et al.*, "Bayesian Models for Unit Discovery on a Very Low Resource Language," in *2018 ICASSP*. IEEE, Apr. 2018, pp. 5939–5943.

[5] M. Versteegh, R. Thiollière, T. Schatz, X. N. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015," in *Interspeech*, vol. 2015-January. ISCA, 2015.

[6] E. Dunbar, X. N. Cao, J. Benjumea *et al.*, "The zero resource speech challenge 2017," in *2017 IEEE ASRU Workshop*. IEEE, Dec. 2017, pp. 323–330.

[7] E. Dunbar, R. Algayres, J. Karadayi *et al.*, "The Zero Resource Speech Challenge 2019: TTS Without T," in *Interspeech 2019*. ISCA, Sep. 2019, pp. 1088–1092.

[8] E. Dunbar, J. Karadayi, M. Bernard, X.-N. Cao, R. Algayres, L. Ondel, L. Besacier, S. Sakti, and E. Dupoux, "The Zero Resource Speech Challenge 2020: Discovering Discrete Subword and Word Units," in *Interspeech 2020*. ISCA, Oct. 2020, pp. 4831–4835.

[9] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural Discrete Representation Learning," in *Proceedings of the 31st NeurIPS*, ser. NIPS'17. Curran Associates Inc., 2017, pp. 6309–6318.

[10] A. Tjandra, B. Sisman, M. Zhang, S. Sakti, H. Li, and S. Nakamura, "VQVAE Unsupervised Unit Discovery and Multi-Scale Code2Spec Inverter for Zerospeech Challenge 2019," in *Interspeech 2019*. ISCA, Sep. 2019, pp. 1118–1122.

[11] B. v. Niekerk, L. Nortje, and H. Kamper, "Vector-Quantized Neural Networks for Acoustic Unit Discovery in the ZeroSpeech 2020 Challenge," in *Interspeech 2020*. ISCA, Oct. 2020, pp. 4836–4840.

[12] K. Lakhotia, E. Kharitonov, Hsu *et al.*, "On generative spoken language modeling from raw audio," *TACL*, vol. 9, 2021.

[13] R. Jakobson and M. Halle, *Fundamentals of Language*. De Gruyter, Dec. 1980.

[14] Y. W. Teh, D. Grür, and Z. Ghahramani, "Stick-breaking Construction for the Indian Buffet Process," in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*. PMLR, Mar. 2007, pp. 556–563, iSSN: 1938-7228.

[15] Z. Ghahramani and T. Griffiths, "Infinite latent feature models and the Indian buffet process," in *Advances in Neural Information Processing Systems*, vol. 18. MIT Press, 2005.

[16] R. Singh, J. Ling, and F. Doshi-Velez, "Structured Variational Autoencoders for Beta-Bernoulli Processes," in *NeurIPS Workshop on Advances in Approximate Bayesian Inference*, 2017, p. 1.

[17] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," May 2014, arXiv:1312.6114 [cs, stat].

[18] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st ICML*, ser. Proceedings of machine learning research, E. P. Xing and T. Jebara, Eds., vol. 32. PMLR, Jun. 2014, pp. 1278–1286.

[19] A. Mnih and D. Rezende, "Variational inference for monte carlo objectives," in *Proceedings of the 33rd ICML*, ser. Proceedings of machine learning research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. PMLR, Jun. 2016, pp. 2188–2196.

[20] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte carlo gradient estimation in machine learning," *JMLR*, vol. 21, no. 1, Jun. 2022.

[21] T. A. Le, A. R. Kosiorek, N. Siddharth, Y. W. Teh, and F. Wood, "Revisiting Reweighted Wake-Sleep for Models with Stochastic Control Flow," in *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, ser. Proceedings of Machine Learning Research, R. P. Adams and V. Gogate, Eds., vol. 115. PMLR, Jul. 2020, pp. 1039–1049.

[22] J. Bornschein and Y. Bengio, "Reweighted Wake-Sleep," Apr. 2015, arXiv:1406.2751 [cs].

[23] S. S. Gu, Z. Ghahramani, and R. E. Turner, "Neural Adaptive Sequential Monte Carlo," in *Advances in Neural Information Processing Systems*, vol. 28. Curran Associates, Inc., 2015.

[24] G. Dubbin and P. Blunsom, "Unsupervised Part of Speech Inference with Particle Filters," in *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*. ACL, Jun. 2012, pp. 47–54.

[25] J.-B. Leger, "Parametrization Cookbook: A set of Bijective Parametrizations for using Machine Learning methods in Statistical Inference," Jan. 2023, arXiv:2301.08297 [stat].

[26] S. Chakraborty and D. Chakravarty, "Discrete Gamma Distributions: Properties and Parameter Estimations," *Communications in Statistics - Theory and Methods*, vol. 41, no. 18, pp. 3301–3324, Sep. 2012.

[27] T. Schatz, V. Peddinti, F. Bach, A. Jansen, H. Hermansky, and E. Dupoux, "Evaluating speech features with the minimal-pair ABX task: Analysis of the classical MFC/PLP pipeline," in *Interspeech*. ISCA, 2013, pp. 1781 – 1785.

[28] T. Schatz, V. Peddinti, X.-N. Cao, F. Bach, H. Hermansky, and E. Dupoux, "Evaluating speech features with the minimal-pair ABX task (II): resistance to noise," in *Interspeech 2014*. ISCA, Sep. 2014, pp. 915–919.

[29] J. Kahn, M. Riviere, W. Zheng *et al.*, "Libri-Light: A Benchmark for ASR with Limited or No Supervision," in *ICASSP 2020*. IEEE, May 2020, pp. 7669–7673.

[30] B. McFee, C. Raffel, D. Liang *et al.*, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, vol. 8, 2015.

[31] M. Riviere and E. Dupoux, "Towards Unsupervised Learning of Speech Features in the Wild," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, Jan. 2021, pp. 156–163.

[32] A. Paszke, S. Gross, F. Massa *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[33] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," Jul. 2016, arXiv:1607.06450 [cs, stat] version: 1.

[34] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 5457–5466, iSSN: 2575-7075.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd ICLR 2015, san diego, CA, USA, may 7-9, 2015, conference track proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[36] E. Dunbar, M. Bernard, N. Hamilakis, T. A. Nguyen, M. d. Seyssel, P. Rozé, M. Rivière, E. Kharitonov, and E. Dupoux, "The Zero Resource Speech Challenge 2021: Spoken Language Modelling," in *Interspeech 2021*. ISCA, Aug. 2021, pp. 1574–1578.