Interspeech 2022
18-22 September 2022, Incheon, Korea

A Sparsity-promoting Dictionary Model for Variational Autoencoders

Mostafa Sadeghi, Paul Magron

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

mostafa.sadeghi@inria.fr, paul.magron@inria.fr

Abstract

Structuring the latent space in probabilistic deep generative models, e.g., variational autoencoders (VAEs), is important to yield more expressive models and interpretable representations, and to avoid overfitting. One way to achieve this objective is to impose a sparsity constraint on the latent variables, e.g., via a Laplace prior. However, such approaches usually complicate the training phase, and they sacrifice the reconstruction quality to promote sparsity. In this paper, we propose a simple yet effective methodology to structure the latent space via a sparsity-promoting dictionary model, which assumes that each latent code can be written as a sparse linear combination of a dictionary's columns. In particular, we leverage a computationally efficient and tuning-free method, which relies on a zero-mean Gaussian latent prior with learnable variances. We derive a variational inference scheme to train the model. Experiments on speech generative modeling demonstrate the advantage of the proposed approach over competing techniques, since it promotes sparsity while not deteriorating the output speech quality.
Index Terms: generative models, variational autoencoders, sparsity, dictionary model, speech spectrogram modeling.

1. Introduction

Unsupervised representation learning [1] is defined as the task of automatically extracting useful information from unlabeled data, in the form of a *feature* or *representation* vector, which can be subsequently used in downstream tasks. To that end, one successful approach, which has gained much attention in the past years, is based on variational autoencoders (VAEs) [2]. These models explicitly consider a latent vector which encapsulates some information about the data. A VAE model consists of a stochastic *encoder* (a recognition network), which transforms the input data into a latent space whose dimension is usually much lower than the original data space, and a stochastic *decoder* (a generative network), which produces data back in the original space from the encoded latent representation. VAEs have been successfully exploited in a variety of speech processing related tasks, such as speech enhancement [3], source separation [4], speech recognition [5], and speech generation and transformation [6].

A common practice in a VAE framework consists in assuming a standard normal distribution as the prior distribution of the latent space. However, this is not effective, as it might not efficiently capture the underlying distribution of the data. Structuring the latent space by imposing some meaningful constraints and distributions is thus of paramount importance to obtain more expressive and interpretable representations. A promising approach is to favor parsimonious or *sparse* latent representations, that is, feature vectors with mostly zeroes or small magnitude entries. Indeed, sparsity is known to promote a variety of benefits, such as increasing interpretability of the latent features and reducing the risk of overfitting [7]. It has proven very successful for feature learning in many fields such as computer vision [8] or natural language processing [9].

In this regard, a sparse prior has been proposed in [10] in the form of a mixture of two Gaussian distributions with fixed variances, where one of them is very small in order to encourage sparsity. A mixing parameter balances the contribution of the two distributions, and thus the amount of sparsity. Tonolini *et al.* [11] proposed a variational sparse coding (VSC) framework, where they consider a Spike-and-Slab distribution [12] for the encoder, which is a mixture of a Gaussian distribution, as in a standard VAE, and a sparsity-promoting Delta function. The prior distribution has the same form as the encoder but instead of directly processing the original data as input, some (unknown) pseudo-input data are defined and learned jointly with the VAE parameters. The sparsity level is tuned via a user-defined parameter, which governs the prior distribution of the mixing variable. Following a different strategy, [13] proposed a *deterministic* dimension selector function, so that some dimensions of the latent vector are deactivated via a point-wise multiplication. The amount of sparsity is then monitored via an entropy-based regularization term. Similarly, in [14] a *stochastic* per-feature masking variable is applied to the latent representation via point-wise multiplication to zero out some dimensions of the latent vector. A hierarchical Spike-and-Slab Lasso prior is assumed for the masking variable, consisting of two Laplace distributions with different parameters. A set of hyper parameters controls the desired amount of sparsity. Even though these approaches have shown promising performance, they often involve several user-defined hyperparameters and non-Gaussian distributions or discrete latent variables, which complicates the learning process.

In this paper, we adopt a different standpoint. We propose to structure the latent space via the usage of a sparsity-promoting dictionary model: each latent vector is assumed to have a sparse representation over a dictionary, i.e., it can be written as a weighted sum of only few columns of the dictionary. As a sparsity regularizer, we adopt a zero-mean Gaussian prior distribution with learnable variances: this is inspired by the relevance vector machines [15], which has proven effective for promoting sparsity. Furthermore, in contrast to non-Gaussian priors, it does not complicate the learning procedure, and does not introduce any extra sparsity-adjusting parameter. We derive an efficient variational inference scheme to learn the parameters, with a negligible extra computational burden compared to a plain VAE. Our experimental results on speech generative modeling demonstrate the effectiveness of the proposed model in promoting sparsity while preserving the reconstructed speech quality.

The rest of the paper is organized as follows. Section 2 reviews the standard VAE model and inference. Section 3 presents the proposed approach and its application to speech spectrogram modeling. Experiments are conducted in Section 4. Finally, Section 5 concludes the paper.

10.21437/Interspeech.2022-237

2. Background on VAE

2.1. Generative modeling

Let $\mathbf{s} = \{\mathbf{s}_1, \dots, \mathbf{s}_N\}$ denote a set of training data with $\mathbf{s}_i \in \mathbb{R}^n$. The core idea in a generative modeling context is to encode the data via some latent variables, denoted $\mathbf{z} = \{\mathbf{z}_1, \dots, \mathbf{z}_N\}$ with $\mathbf{z}_i \in \mathbb{R}^m$, where usually $m \ll n$. Then, the goal is to model the joint distribution $p(\mathbf{s}, \mathbf{z}) = p(\mathbf{s}|\mathbf{z}) \cdot p(\mathbf{z})$. To this end, the variational autoencoding framework [2] assumes some parametric forms for the generative distribution $p(\mathbf{s}|\mathbf{z})$, which is also referred to as the *decoder*, and for the prior distribution $p(\mathbf{z})$, which are typically expressed as Gaussian distributions:

$$\begin{cases} p_\theta(\mathbf{s}|\mathbf{z}) = \mathcal{N}(\boldsymbol{\mu}_\theta(\mathbf{z}), \text{diag}(\boldsymbol{\sigma}^2_\theta(\mathbf{z})), \\ p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I}), \end{cases} \tag{1}$$

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\mathbf{I}$ denotes the identity matrix of appropriate size, and the power is applied element-wise. Furthermore, $\boldsymbol{\mu}_\theta(.)$ and $\boldsymbol{\sigma}_\theta(.)$ are some non-linear functions denoting the mean and standard deviation, which are implemented via some deep neural networks (DNNs). To learn the model parameters, i.e., θ, one would need to compute the posterior distribution of the latent codes, that is $p_\theta(\mathbf{z}|\mathbf{s})$, which is, unfortunately, intractable to compute. The VAE framework proposes to approximate this distribution with a parametric Gaussian distribution, called the *encoder*, as follows:

$$q_\psi(\mathbf{z}|\mathbf{s}) = \mathcal{N}(\boldsymbol{\mu}_\psi(\mathbf{s}), \text{diag}(\boldsymbol{\sigma}^2_\psi(\mathbf{s})), \tag{2}$$

where $\boldsymbol{\mu}_\psi$ and $\boldsymbol{\sigma}_\psi$ are also implemented using some DNNs with parameters ψ.

2.2. Parameter estimation

The whole set of parameters $\Phi = \{\theta, \psi\}$ is learned using the following variational procedure. Instead of directly optimizing the data log-likelihood $\log p_\theta(\mathbf{s})$, which is intractable, a lower bound $\mathcal{L}$ is targeted such that $\mathcal{L}(\Phi; \mathbf{s}) \leq \log p_\theta(\mathbf{s})$. To that end, the evidence lower bound (ELBO) of the data log-likelihood is considered, which is defined as:

$$\mathcal{L}(\Phi; \mathbf{s}) = \mathbb{E}_{q_\psi(\mathbf{z}|\mathbf{s})}[\log p_\theta(\mathbf{s}|\mathbf{z})] - \mathcal{D}_{\text{KL}}(q_\psi(\mathbf{z}|\mathbf{s}) \| p(\mathbf{z})), \tag{3}$$

where $\mathcal{D}_{\text{KL}}(q\|p)$ represents the Kullback–Leibler (KL) divergence between q and p. The KL divergence acts as a regularizer in the above formula, whereas the first term measures the reconstruction quality of the model. Then, the overall training phase consists of optimizing $\mathcal{L}(\Phi; \mathbf{s})$ over Φ using a gradient-based optimizer together with the application of the *reparametrization trick*, which enables backpropagation through the encoder parameters [2].

3. Proposed framework

3.1. Model

The main idea underlying our proposal consists in structuring the latent space by considering a sparse dictionary model for each latent code $\mathbf{z}_i$ as follows:

$$\mathbf{z}_i = \mathbf{D}\mathbf{a}_i, \quad \forall i, \tag{4}$$

where $\mathbf{D} \in \mathbb{R}^{m \times k}$ is a dictionary with unit-norm columns (to avoid scale ambiguity), and $\mathbf{a}_i \in \mathbb{R}^k$ denotes a (sparse) representation. In this work, we consider that $\mathbf{D}$ is a fixed dictionary,

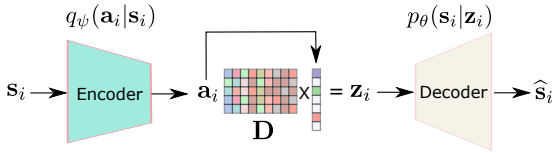

Figure 1: *Schematic diagram of the proposed VAE with a sparse dictionary model.*

even though it could also be learned jointly with the code vector $\mathbf{a}_i$. Note that one way to achieve a sparser representation is to consider an overcomplete dictionary, that is, such that $k > m$. The proposed model is illustrated in Fig. 1.

The motivation behind the model in (4) is to keep the reconstruction quality of VAEs intact, while at the same time promoting interpretability of the latent space by enforcing sparsity of the code vector $\mathbf{a}_i$. To impose such a constraint, one possible approach is to leverage some sparsity-promoting prior, such as based on the Laplace distribution [16]. However, such distributions involve non-smooth or complicated forms, which makes parameter inference and training more cumbersome. Instead, we propose to use the following zero-mean Gaussian prior:

$$p(\mathbf{a}_i; \boldsymbol{\gamma}_i) = \mathcal{N}(\mathbf{0}, \text{diag}(\boldsymbol{\gamma}_i)), \tag{5}$$

in which $\boldsymbol{\gamma}_i$ is a vector of learnable variances. This prior has already proven successful in obtaining sparse solutions for linear regression and classification tasks [15, 17]. Indeed, as discussed in [17], if one considers an inverse Gamma hyperprior on $\boldsymbol{\gamma}_i$ in a Bayesian setting, then the prior distribution, which is given by:

$$p(\mathbf{a}_i) = \int p(\mathbf{a}_i|\boldsymbol{\gamma}_i) \cdot p(\boldsymbol{\gamma}_i) \text{d}\boldsymbol{\gamma}_i, \tag{6}$$

becomes a Student's t-distribution, which has been shown to promote sparsity due to its sharp peak at zero [15]. Even though we do not consider such a hyperprior explicitly, we still observe experimentally (see Section 4) that the model given by (4) and (6) promotes sparsity. We refer the reader to [17] for more detailed discussions.

3.2. Inference

The inference phase follows the standard procedure of VAEs described in Section 2.2, with the main difference being that the encoder is now used to encode each $\mathbf{s}_i$ into $\mathbf{a}_i$, then compute $\mathbf{z}_i$ via the dictionary model in (4), and finally pass it to the decoder to reconstruct $\mathbf{s}_i$. Therefore, $\mathbf{z}_i$ only implicitly participates in the inference. Then, the cost function to optimize is:

$$\begin{aligned} \mathcal{L}(\Phi, \boldsymbol{\gamma}; \mathbf{s}) &= \mathbb{E}_{q_\psi(\mathbf{a}|\mathbf{s})}[\log p_\theta(\mathbf{s}|\mathbf{z})] - \mathcal{D}_{\text{KL}}(q_\psi(\mathbf{a}|\mathbf{s}) \| p(\mathbf{a}; \boldsymbol{\gamma})), \\ &\text{with} \quad \mathbf{z}_i = \mathbf{D}\mathbf{a}_i \quad \forall i. \end{aligned} \tag{7}$$

It should be noted that Φ are called *amortized* parameters, as they are shared among all the training data, while $\boldsymbol{\gamma} = \{\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_N\}$ are parameters specific to each data point.

We use an alternating minimization approach to update Φ and $\boldsymbol{\gamma}$ by alternately fixing one variable and optimizing the cost function over the other one. This corresponds to solving the following two sub-problems iteratively:

$$\begin{cases} \text{Update } \boldsymbol{\gamma}: \quad \boldsymbol{\gamma} \leftarrow \text{argmin}_{\boldsymbol{\gamma}} \ \mathcal{D}_{\text{KL}}(q_\psi(\mathbf{a}|\mathbf{s}) \| p(\mathbf{a}; \boldsymbol{\gamma})) & \text{(8a)} \\ \text{Update } \Phi: \quad \Phi \leftarrow \text{argmax}_{\Phi} \ \mathcal{L}(\Phi, \boldsymbol{\gamma}; \mathbf{s}). & \text{(8b)} \end{cases}$$

Since the KL divergence between two Gaussian distributions can be obtained in closed-form, it is straightforward to

Algorithm 1 SDM-VAE

1: **Input:** Training data $\mathbf{s} = \{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$, $\mathbf{D}$.
2: **Initialize:** $\Phi = \{\theta, \psi\}$ with random entries.
3: **While** stopping criterion not met **do**:
4: **For** each mini-batch (b):
▷ $\boldsymbol{\gamma}^{(b)}$ **- update:** Using (9)
▷ **Reparametrization:**
▷ $\epsilon^{(b)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
▷ $\mathbf{a}^{(b)} = \boldsymbol{\mu}_\psi(\mathbf{s}^{(b)}) + \boldsymbol{\sigma}_\psi(\mathbf{s}^{(b)}) \odot \boldsymbol{\epsilon}^{(b)}$
▷ $\mathbf{z}^{(b)}$ **- update:** Using $\mathbf{z}^{(b)} = \mathbf{D}\mathbf{a}^{(b)}$
▷ $\boldsymbol{\Phi}$ **- update:** Using one-step gradient ascent on (10)
5: **End for**
6: **End while**

solve (8a), which yields the following update rule:

$$\boldsymbol{\gamma}_i = \mathbb{E}_{q_\psi(\mathbf{a}_i|\mathbf{s})}[\mathbf{a}_i^2] = \mu_\psi^2(\mathbf{s}_i) + \sigma_\psi^2(\mathbf{s}_i), \quad \forall i. \tag{9}$$

Then, we resort to the reparametrization trick [2] to approximate the expectation in (7), and thus reformulate subproblem (8b). Specifically, we approximate it using a single sample given by $\mathbf{a}_i = \boldsymbol{\mu}_\psi(\mathbf{s}_i) + \boldsymbol{\sigma}_\psi(\mathbf{s}_i) \odot \boldsymbol{\epsilon}_i$, where $\odot$ denotes the element-wise multiplication, and $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We then compute $\mathbf{z}_i = \mathbf{D}\mathbf{a}_i$ for each data sample i in order to finalize computation of the following approximated loss:

$$\hat{\mathcal{L}}(\Phi, \boldsymbol{\gamma}; \mathbf{s}) = \log p_\theta(\mathbf{s}|\mathbf{z}) - \mathcal{D}_{\text{KL}}(q_\psi(\mathbf{a}|\mathbf{s}) \| p(\mathbf{a}; \boldsymbol{\gamma})). \tag{10}$$

Then, we optimize (10) by a single gradient ascent step, as in the standard VAE learning framework, which yields the update on Φ. The proposed procedure, called the sparsity-promoting dictionary model VAE (SDM-VAE) is summarized in Algorithm 1. Note that for practical implementation, the training data is split into mini-batches, and a stochastic gradient algorithm is applied sequentially to each mini-batch.

3.3. Application to speech modeling

In this paper, we evaluate the effectiveness of the proposed approach for the modeling of speech signals, that is, speech *analysis-resynthesis* [18]. To that end, we first compute the short-time Fourier transform (STFT) of the time-domain signals, which yields the complex-valued data $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, where N is the total number of time frames. In practice, the VAE processes and retrieves power spectrograms $\mathbf{s} = |\mathbf{x}|^2$. In terms of probabilistic modeling, this boils down to assuming that the data follows a circularly-symmetric zero-mean complex-valued Gaussian distribution, which is a common assumption when modeling speech signals with VAEs [3].

Once the VAE model is trained on the power spectrograms, we perform speech analysis-resynthesis on the test data, similarly to [18]. More precisely, once the input power spectrogram is encoded to obtain the posterior parameters, the latent's posterior mean is fed to the decoder to reconstruct the input spectrogram. Then, the magnitude spectrogram estimate $\hat{\mathbf{s}}^{1/2}$ is combined with the input STFT's phase $\angle\mathbf{x}$ in order to obtain a complex-valued STFT: $\hat{\mathbf{x}} = \hat{\mathbf{s}}^{1/2} \odot \exp(j\angle\mathbf{x})$. Finally, time-domain estimates are retrieved by applying the inverse STFT to $\hat{\mathbf{x}}$. Note that, as such, the VAE ignores the phase information, which is only exploited for re-synthesizing time-domain signals. We leave modeling the phase in VAEs [19] or extending the proposed method to complex-valued autoencoders, e.g., by overcoming the circularly-symmetric assumption [20], to future work.

As for the dictionary, we consider a discrete cosine transform (DCT) matrix with k atoms, that is, the r-th column of the DCT dictionary $\mathbf{D}$ is defined as follows [21]:

$$\mathbf{d}_r = [\cos((\ell - 1)\pi r/k)]_{\ell=1}^m, \tag{11}$$

followed by mean subtraction and normalization to ensure unit-norm columns.

4. Experiments

In this section, we assess the potential of our proposed VAE model in terms of speech modeling. The code implementing the proposed VAE model is available online for reproducibility.[1]

4.1. Protocol

4.1.1. Data

We use the speech data in the TCD-TIMIT corpus [22] for training and evaluating the model. It includes speech utterances from 56 English speakers with an Irish accent, uttering 98 different sentences, each with an approximate length of 5 seconds, and sampled at 16 kHz. The total speech data duration is about 8 hours. 39 speakers are used for training, 8 for validation, and the remaining 9 speakers for testing. The STFT parameters are as follows. The STFT is computed with a 1024 samples-long (64 ms) sine window, 75% overlap and no zero-padding, which yields STFT frames of length $n = 513$.

4.1.2. Methods & model architecture

As baseline methods, we compare the performance of a standard VAE[2] [3], the VSC model[3] [11], and the proposed SDM-VAE model. All the VAE models follow the same simple encoder-decoder architecture as the one proposed in [3] in order to focus our experiments on the latent space structure. More precisely, both the encoder and decoder are fully-connected networks using a single hidden layer with 128 nodes and hyperbolic tangent activation functions. The dimension of the latent space is $m \in \{32, 64\}$. The dictionary in SDM-VAE is a DCT matrix, as defined in (11), which contains $k \in \{32, 64\}$ atoms. These parameters values are chosen according to prior studies [3, 11], where they have shown good performance. For comparison, we also consider the identity matrix as the dictionary $\mathbf{D} = \mathbf{I}$. In this setting, the model becomes similar to a classical VAE, except the latent prior now is $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \text{diag}(\boldsymbol{\gamma}_i))$ instead of the standard normal prior $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

4.1.3. Parameter settings & training

All the VAE variants are implemented in PyTorch [23], and trained with mini-batch stochastic gradient descent using the Adam optimizer [24] with a learning rate equal to 0.0001, and a batch size of 128. As stopping criterion we use early stopping on the validation set with a patience of 20 epochs, meaning that the training stops if the validation loss does not improve after 20 consecutive epochs. For the sparsity parameter of the VSC model, denoted α, we have experimented three different values: $\{0.05, 0.5, 0.9\}$, where a lower value corresponds to a sparser representation.

[1] https://gitlab.inria.fr/smostafa/sdm-vae
[2] https://gitlab.inria.fr/smostafa/avse-vae
[3] https://github.com/Alfo5123/Variational-Sparse-Coding

Table 1: *Reconstruction quality and sparsity measure for various VAE-based methods in terms of PESQ, STOI, and Hoyer scores. The results are averaged over the test set (the standard deviation for all methods is in the order of* 0.01 *for PESQ, and* 0.001 *for STOI and the Hoyer score).*

Dimension of $\mathbf{z}$		$m = 32$			$m = 64$		
		PESQ	STOI	Hoyer	PESQ	STOI	Hoyer
VAE [3]		3.29	0.85	0.40	3.26	0.85	0.56
VSC [11]	$\alpha = 0.05$	3.00	0.81	0.57	3.25	0.84	0.51
	$\alpha = 0.5$	3.25	0.84	0.54	3.32	0.85	0.65
	$\alpha = 0.9$	3.25	0.84	0.47	3.26	0.85	0.60
SDM-VAE	$\mathbf{I}$	3.33	0.86	0.64	**3.45**	**0.87**	0.73
	DCT ($k = 32$)	3.37	0.86	0.66	3.28	0.84	0.66
	DCT ($k = 64$)	3.32	0.86	**0.87**	3.33	0.86	0.76

4.1.4. Evaluation

To evaluate the compared methods in terms of reconstruction quality, we compute the short-term objective intelligibility (STOI) measure [25] and the perceptual evaluation of speech quality (PESQ) score [26]. These metrics respectively belong to the $[-0.5, 4.5]$ and $[0, 1]$ range, and higher is better. We also compute the Hoyer metric [27] in order to evaluate the sparsity of the latent codes. For a vector $\mathbf{z} \in \mathbb{R}^m$, the Hoyer metric is defined as follows:

$$\text{Hoyer}(\mathbf{z}) = \frac{\sqrt{m} - \|\mathbf{z}\|_1 / \|\mathbf{z}\|_2}{\sqrt{m} - 1}, \tag{12}$$

where $\|.\|_1$ and $\|.\|_2$ respectively denote the ℓ_1 and ℓ_2 norms. The Hoyer metric ranges in $[0, 1]$, and a higher score corresponds to a more sparse latent feature. It should be emphasized that for the proposed SDM-VAE model, the sparsity is measured on $\mathbf{a}$, since it is the interpretable latent code in this case.

4.2. Results

From the results, averaged over all the test samples, and displayed in Table 1, we can draw several conclusions. First, we observe that for the classical VAE, the PESQ decreases when going from $m = 32$ to 64. Overall, the opposite behavior is observed for the sparse VAE models, for which increasing the latent space dimension does not sacrifice the reconstruction quality (except for SDM-VAE when using a DCT dictionary with $k = 32$ atoms, which is expected as the dictionary becomes undercomplete for $m = 64$). These results confirm a known advantage of sparsity as a regularizer for the model to avoid overfitting.

For VSC, we see that increasing sparsity (i.e., decreasing α) results in sacrifying the reconstruction quality in terms of PESQ when $m = 32$, although a different trend is observed for $m = 64$. Our proposed SDM-VAE model using the DCT dictionary is less sensitive to the sparsity level, since it exhibits more stable STOI values, and an increasing PESQ when the sparsity score increases.

We also observe that increasing the number of atoms in the dictionary results in an increased sparsity score. In particular, the highest sparsity score is obtained for $m = 32$ and when using an overcomplete dictionary ($k > m$). Nonetheless, this setting does not improve reconstruction quality compared to using a complete dictionary. As a result, using $k = m$ would be an appropriate and simple guideline, since it maximizes reconstruction quality and avoid fine-tuning the dictionary size.

Interestingly, we observe that the best SDM-VAE results are obtained when using an identity dictionary rather than the DCT. This shows that a simple change in the prior structure compared to the baseline VAE (as detailed in Section 4.1.2) is an efficient way to both improve performance and promote sparsity, rather than resorting to more involved models such as in VSC. Nevertheless, the DCT dictionary remains the best performing option when $m = 32$. This motivates a future investigation on the design of more optimal dictionaries, or the joint learning of the dictionary along with other parameters based on dictionary learning algorithms from the literature [21, 28].

Finally, we remark that SDM-VAE yields the best results in terms of both reconstruction quality (PESQ and STOI) and sparsity (Hoyer score). This outlines the potential of our proposal, which does not rely on a complicated inference scheme or extra-parameter tuning, and exhibits a stable performance while enabling interpretability and avoiding overfitting.

5. Conclusions

In this work, we have proposed a novel approach for promoting sparsity in VAEs based on structuring the latent space using a sparse dictionary model. We derived a simple inference scheme, which does not require any fine-tuning of hyperparameters. Experiments conducted on speech signals modeling have demonstrated the potential of this technique compared to other existing sparse approaches. In particular, the proposed method improves speech reconstruction quality, and efficiently exploits sparsity to improve interpretability of the latent representation while reducing the risk of overfitting. Future work will focus on learning the dictionary along with other parameters, possibly with some mutual coherence constraint [28]. Extending the developed model to dynamical VAEs [18], and exploiting it in speech enhancement [3] and source separation applications [4] are other future directions to pursue.

6. Acknowledgements

Experiments presented in this paper were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr).

7. References

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *EEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, p. 1798–1828, aug 2013.

[2] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. International Conference on Learning Representations (ICLR)*, April 2014.

[3] S. Leglaive, L. Girin, and R. Horaud, "A variance modeling framework based on variational autoencoders for speech enhancement," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, September 2018.

[4] V.-N. Nguyen, M. Sadeghi, E. Ricci, and X. Alameda-Pineda, "Deep variational generative models for audio-visual speech separation," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, August 2021.

[5] S. Tan and K. C. Sim, "Learning utterance-level normalisation using variational autoencoders for robust automatic speech recognition," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, December 2016, pp. 43–49.

[6] W.-N. Hsu, Y. Zhang, and J. R. Glass, "Learning latent representations for speech generation and transformation," in *Proc. Interspeech*, August 2017.

[7] A. Asperti, "Sparsity in variational autoencoders," *arXiv preprint arXiv:1812.07238*, 2019.

[8] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," in *Proc. International Conference on Machine Learning (ICML)*, July 2015.

[9] V. Prokhorov, Y. Li, E. Shareghi, and N. Collier, "Learning sparse sentence encoding without supervision: An exploration of sparsity in variational autoencoders," in *Proc. Workshop on Representation Learning for NLP*, August 2021, p. 34–46.

[10] E. Mathieu, T. Rainforth, N. Siddharth, and Y. W. Teh, "Disentangling disentanglement in variational autoencoders," in *Proc. International Conference on Machine Learning (ICML)*, June 2019.

[11] F. Tonolini, B. S. Jensen, and R. Murray-Smith, "Variational sparse coding," in *Proc. conference on Uncertainty in Artificial Intelligence (UAI)*, August 2020.

[12] M. R. Andersen, O. Winther, and L. K. Hansen, "Bayesian inference for structured spike and slab priors," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, vol. 27, December 2014.

[13] N. Miao, E. Mathieu, N. Siddharth, Y. W. Teh, and T. Rainforth, "On incorporating inductive biases into VAEs," in *Proc. International Conference on Learning Representations (ICLR)*, May 2021.

[14] G. E. Moran, D. Sridhar, Y. Wang, and D. M. Blei, "Identifiable variational autoencoders via sparse decoding," *arXiv preprint arXiv:2110.10804*, 2021.

[15] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of machine learning research*, vol. 1, no. Jun, pp. 211–244, 2001.

[16] S. Mohamed, K. A. Heller, and Z. Ghahramani, "Bayesian and L1 approaches for sparse unsupervised learning," in *Proc. International Conference on Machine Learning (ICML)*, 2012.

[17] D. P. Wipf and B. D. Rao, "Sparse bayesian learning for basis selection," *IEEE Transactions on Signal processing*, vol. 52, no. 8, pp. 2153–2164, 2004.

[18] X. Bie, L. Girin, S. Leglaive, T. Hueber, and X. Alameda-Pineda, "A benchmark of dynamical variational autoencoders applied to speech spectrogram modeling," in *Proc. Interspeech*, August 2021.

[19] A. A. Nugraha, K. Sekiguchi, and K. Yoshii, "A deep generative model of speech complex spectrograms," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 905–909.

[20] P. Magron, R. Badeau, and B. David, "Phase-dependent anisotropic gaussian model for audio source separation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 531–535.

[21] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[22] N. Harte and E. Gillen, "TCD-TIMIT: An audio-visual corpus of continuous speech," *IEEE Transactions on Multimedia*, vol. 17, no. 5, pp. 603–615, 2015.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. International Conference on Learning Representations ICLR*, May 2015.

[25] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time–frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, February 2011.

[26] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2001.

[27] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, p. 1457–1469, dec 2004.

[28] M. Sadeghi and M. Babaie-Zadeh, "Dictionary learning with low mutual coherence constraint," *Neurocomputing*, vol. 407, pp. 163–174, 2020.