

VARIATIONAL SPARSE CODING

ICLR 2019 REPRODUCIBILITY CHALLENGE REPORT

Alfredo de la Fuente

Department of Statistical Learning Theory
Skolkovo Institute of Science and Technology
Moscow, Russia
alfredo.delafuente@skoltech.ru

Robert Aduviri

Artificial Intelligence Research Group
Pontifical Catholic University of Peru
Lima, Peru
robert.aduviri@pucp.edu.pe

ABSTRACT

In this work we replicate and validate the experiments described in the paper “Variational Sparse Coding” submitted to the ICLR 2019 Conference (Tonolini et al., 2019). Aiming to reproduce the results obtained by the authors, we implemented from scratch the variational auto-encoder architecture as described in the submitted paper to learn sparse representations for MNIST, Fashion-MNIST and CelebA datasets. We run the experiments and propose further suggestions to improve model performance at generating samples from low-dimensional vectors, traversing latent space, and using the sparse representations for classification tasks. We conclude that reproducibility, given the paper description, is overall possible. The code repository can be found in the following link:

<https://github.com/Alfo5123/Variational-Sparse-Coding>

1 INTRODUCTION

The paper (Tonolini et al., 2019) proposes an improvement over the Variational Auto-Encoder (VAE) architecture (Rezende et al., 2014; Kingma & Welling, 2013) by explicitly modelling sparsity in the latent space with a Spike and Slab prior distribution and drawing ideas from sparse coding theory. The main motivation behind their work lies in the ability to infer truly sparse representations from generally intractable non-linear probabilistic models, simultaneously addressing the problem of lack of interpretability of latent features. Moreover, the proposed model improves the classification accuracy using the low-dimensional representations obtained, and significantly adds robustness while varying the dimensionality of latent space.

The authors of the paper derive an analytic expression for the evidence lower bound (ELBO) of the VAE model by choosing the sparsity-inducing Spike and Slab prior distribution for the latent variables, which is later optimized using standard gradient methods to recover the encoding and decoding mappings. After training on well-known datasets, the Variational Sparse Coding (VSC) model is able to recover sparse, informative and interpretable representations given a fixed number of latent dimensions, which authors claim to be advantageous over standard VAE representations for classification tasks.

In this reproducibility report we study in detail the VSC model to implement the architectures described in the paper, run the experiments (detailed in Section 2), provide insights and suggestions for replicating results, and analyze the results obtained in comparison with the ones reported by the authors of the paper (Section 3). Furthermore, we add modifications to improve the model performance and propose some possible future work directions (Section 4).

1.1 RELATED WORK

Variational Auto-Encoders have been extensively studied (Doersch, 2016) and widely modified in the recent years in order to encourage certain behavior of the latent space variables (Nalisnick & Smyth, 2016; Rolfe, 2016; Casale et al., 2018) or to be further applied for particular tasks (Chen et al., 2016; Walker et al., 2016; Kusner et al., 2017; Jin et al., 2018). Regarding the sparsity of the latent space for VAEs, previous work in the literature has focused on either explicitly incorporating

a regularization term to benefit sparsity (Louizos et al., 2017), or fixing a prior distribution, such as Rectified Gaussians by Salimans (2016), discrete distributions by van den Oord et al. (2017), student-t distribution for Variational Information Bottleneck by Chalk et al. (2016) and Stick Breaking Processes by Nalisnick & Smyth (2016).

Nonetheless, previous works have not allowed to explicitly model sparsity by incorporating linear sparse coding to non-linear probabilistic generative models. The paper we aim to reproduce, offers a connection between both areas through the Spike-and-Slab distribution, chosen as a prior distribution for the latent variables. Although this distribution has been commonly used for modeling sparsity (Goodfellow et al., 2012), it has rarely been applied to generative models. Moreover, since sparse coding imposes efficient data representations (Ishwaran et al., 2005; Titsias & Lázaro-Gredilla, 2011; Bengio et al., 2013), the authors demonstrate qualitatively how the sparse learned representations can capture subjectively understandable sources of variation.

Following the line of latent features interpretability, we can observe that the authors’ idea is closely related to the Epitomic VAE by Yeung et al. (2016), which learns the latent dimensions the recognition function should exploit. Many recent approaches, mostly related to disentangled representations, such as β -VAE (Higgins et al., 2016; Burgess et al., 2018) or Factor-VAE by Kim & Mnih (2018), focus on learning interpretable factorized representations of the independent data generative factors via generative models. However, these approaches although explicitly induce interpretation of the latent features, do not directly produce sparse representations in contrast with the VSC model. Hence, the authors’ aim is to develop a model that directly induces sparsity in a continuous latent space, which in addition, results into a higher expectation of interpretability in large latent spaces.

Our work, as part of the reproducibility challenge, will contribute in clarifying the implementation details of the VSC model, corroborate the results, and assess a few concerns of the reviewers by adding small modifications to the model and experiments based on the insights obtained during the reproducibility challenge.

1.2 TARGET QUESTIONS

In order to assess the reproducibility of the paper and validate its conclusions, the main questions we will focus our efforts on answering are:

- CAN WE ACTUALLY VALIDATE THE REPORTED RESULTS?
- IS IT POSSIBLE TO INTERPRET THE LATENT FEATURES LEARNED BY THE VSC MODEL?
- HOW THE PROPOSED MODEL CAN BE FURTHER IMPROVED?

2 EXPERIMENTAL METHODOLOGY

Within the experiments described in the paper, we focus on replicating the precise settings for:

- *ELBO evaluation*: to observe ELBO drop while optimizing the model loss.
- *Classification Performance*: to use the learned presentations for classification tasks.
- *Interpretation of sparse codes*: to visually inspect the role of learned latent features.
- *Visualization / Traversal of Latent Space*: to qualitatively evaluate the reconstructed images.

We found that the paper was well written and moderately amenable for reproduction. Although the authors did not make the code available, writing the code from scratch was not as challenging as we anticipated. Thus, in this section, we describe in detail how our implementation of the model was carried out, clarify the adjustments considered and display the results obtained.

2.1 DATASETS

We test the VSC model in two commonly used image datasets: MNIST (LeCun et al., 1998) and Fashion-MNIST (Xiao et al., 2017), both composed of 28×28 grey scale images of handwritten digits and pieces of clothing respectively. Following the paper description, we run most of the

experiments with these datasets. In addition to this, CelebA faces (Liu et al., 2015) dataset was used to showcase qualitative results. We include in our repository routines to download and preprocess the datasets.

OBSERVATIONS

- For the CelebA dataset, we used a subset of 100K samples for training and 20K samples for testing, which were center cropped and downsampled to a size of 32×32 using all 3 RGB channels, as described in the paper.
- We applied a 0-1 Min-Max scaling to all the datasets. We observed that normalizing vectors to unitary norm, as the paper suggests, produced lower quality image reconstruction.

2.2 IMPLEMENTATION DETAILS

We decided to replicate the architecture described in the paper using PyTorch (Paszke et al., 2017). Our repository includes a few instructions on how to install and set up all the required libraries needed for running our implementation. We organize the code on scripts for each model architecture, as well as Jupyter Notebooks for preprocessing, running experiments and visualization.

In order to establish a valid benchmark, we implemented the VAE architecture from Kingma & Welling (2013), which, in the same way as the Variational Sparse Coding (VSC) model, was implemented by an encoder and a decoder function, both parametrized as fully connected neural networks. The architecture, implemented from scratch, allows to explicitly define the hyperparameters of the model (e.g., hidden layer dimension, latent space dimension, learning rate, epochs, batch size, etc.) and is highly modular, to encourage future modifications.

For the loss function, the authors used a continuous relaxation for the discrete binary component in the reconstruction term of the ELBO and applied the reparametrization trick (Kingma et al., 2015) for the Spike and Slab distribution to obtain a differentiable expression which can be optimized. Implementing this code was straightforward, our implementation forces the parameter c to increase by 0.001 per iteration to benefit convergence stability.

We stored all the checkpoints for the trained models, for reproducibility purposes, together with the training logs which can be visualized using TensorBoard.

OBSERVATIONS

- One of the missing details in the paper was the batch size. We assumed it to be 32 samples per batch, due to our memory restrictions.
- The original paper suggests using 20,000 iterations for model training using the ADAM optimizer (Kingma & Ba, 2014) with learning rate ranging between 0.001 and 0.01. In particular, we implemented the VSC model in a way that the number of epochs is one hyperparameter. Thus, we fixed the number of epochs to be equivalent the number of iterations given by the paper; i.e., for MNIST and Fashion-MNIST we trained the VSC model for 11 epochs with a batch size of 32. We fixed a learning rate of 0.001.
- A minor downside of the paper is that the weights initialization method was not specified.
- For the recognition function, in order to avoid numerical instability, we suggest to either use *clamp* or Sigmoid activation function to avoid spike values of zero (thus ensuring $\gamma_i < 1$).

2.3 REPRODUCIBILITY COST

Given that there was no mention in the paper about computational resources required, we describe the computational cost required for running the experiments. Since MNIST and Fashion-MNIST are relatively small datasets, the training procedure run on CPU took around 1 minute per epoch and 8 seconds per epoch on a Titan Xp GPU, using a latent size of 200 units and a single hidden layer of 400 units for both the encoder and decoder, as described in the paper. On the other hand, the VSC model training time for the CelebA dataset was around 30 seconds per epoch on a Titan Xp GPU, using a latent size of 800 units and two hidden layers of 2000 units also for the encoder and decoder. Regarding memory requirements on GPU, the network trained on MNIST and Fashion-MNIST consumed around 529MB, which scaled up to 850MB during training, while the network trained on

CelebA consumed around 637MB, which scaled up to 1322MB during training. In conclusion, the computational cost for running the experiments is not high, which facilitated the reproduction of the results.

2.4 RESULTS

2.4.1 ELBO EVALUATION

We evaluated how the Variational Lower Bound (VLB) varies while changing the latent space dimension of the models: VSC - $\alpha = 0.2$, VSC - $\alpha = 0.5$ and VAE (Figure 1). We observed that in general the ELBO decreases until reaching an optimal latent dimension and then they slowly increase as we add more latent dimensions.

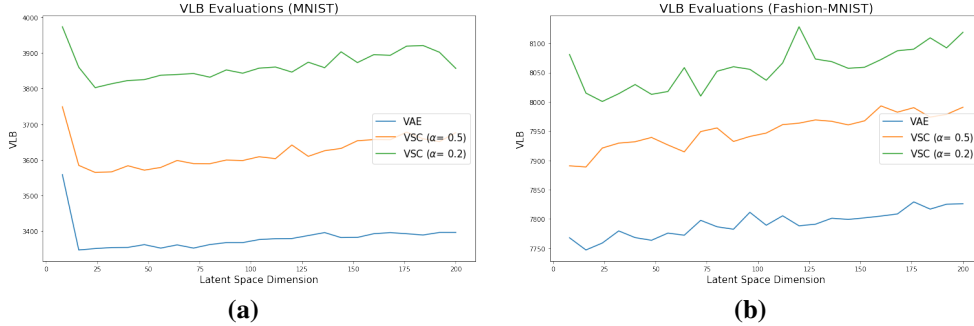


Figure 1: Evaluation of ELBO on the test set for the VSC trained model at varying number of latent dimensions.

2.4.2 CLASSIFICATION PERFORMANCE

We studied the classification accuracy obtained by using the latent representations as input (Figure 2), in order to validate the authors' conclusion that the Variational Sparse Coding learned representations improve the classification results and significantly increase robustness with respect to the latent number of dimensions. We can observe that for both MNIST and Fashion-MNIST, after surpassing the optimal latent dimension of the VAE, the representations learned by the VSC are significantly more informative as the latent dimension increases.

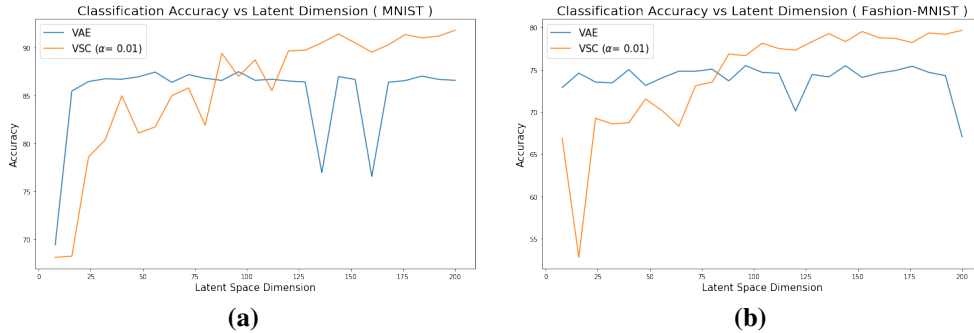


Figure 2: Classifications results on the test set (for MNIST (a) and Fashion-MNIST (b)) by varying the latent space dimensions using the model VSC - $\alpha = 0.01$ and a 2-layer fully connected neural network as base classifier.

2.4.3 INTERPRETATION OF SPARSE CODES

Authors claim the interpretability of the sparse learned representations. We qualitatively examined the interpretation of the non-zero elements in the sparse codes recovered with the VSC model by

running interpolations in the sparse space varying the dimension with the highest absolute value (Figure 3). In addition, we observe the effect of the latent dimensionality at capturing the image content (Figure 4).

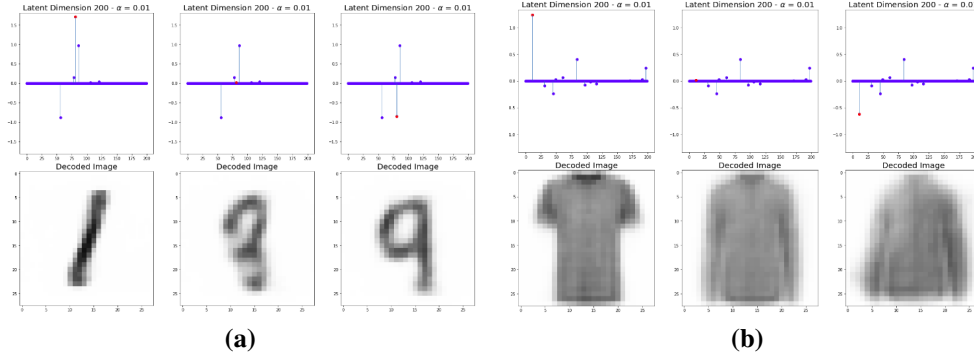


Figure 3: Reconstruction results by modifying encoding in 200-dimensional latent space for MNIST (a) and Fashion-MNIST (b) for the model VSC - $\alpha = 0.01$.

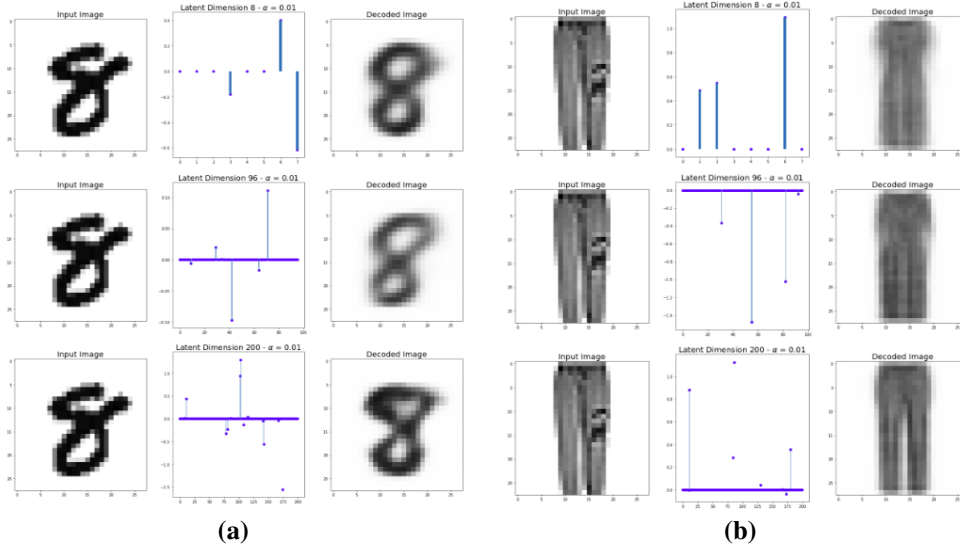


Figure 4: Encoding and reconstruction of samples using the VSC - $\alpha = 0.01$ model from MNIST (a) and Fashion-MNIST (b) datasets with varying latent dimensions of 8, 96 and 200.

2.4.4 VISUALIZATION / TRAVERSING LATENT SPACE

We explored how sampling from the latent space distribution can allow us to obtain interpretable variations in the generated images (Figure 5), and also how conditional sampling produces arguably realistic new samples from the same conceptual entity. (Figure 6). The traversal of the latent space is performed varying the latent codes with a high absolute value for a given image, one at a time. We can observe that these latent codes indeed represent interpretable features of the datasets, such as the digits shape in MNIST, the color and shape of the clothes in Fashion-MNIST and the orientation, background color, skin color and hair color in the CelebA results.

3 ANALYSIS AND DISCUSSION OF FINDINGS

We should note that the paper we reproduced was clearly written and overall, despite a few implementation details unspecified (batch size, number of epochs for the CelebA training and weights

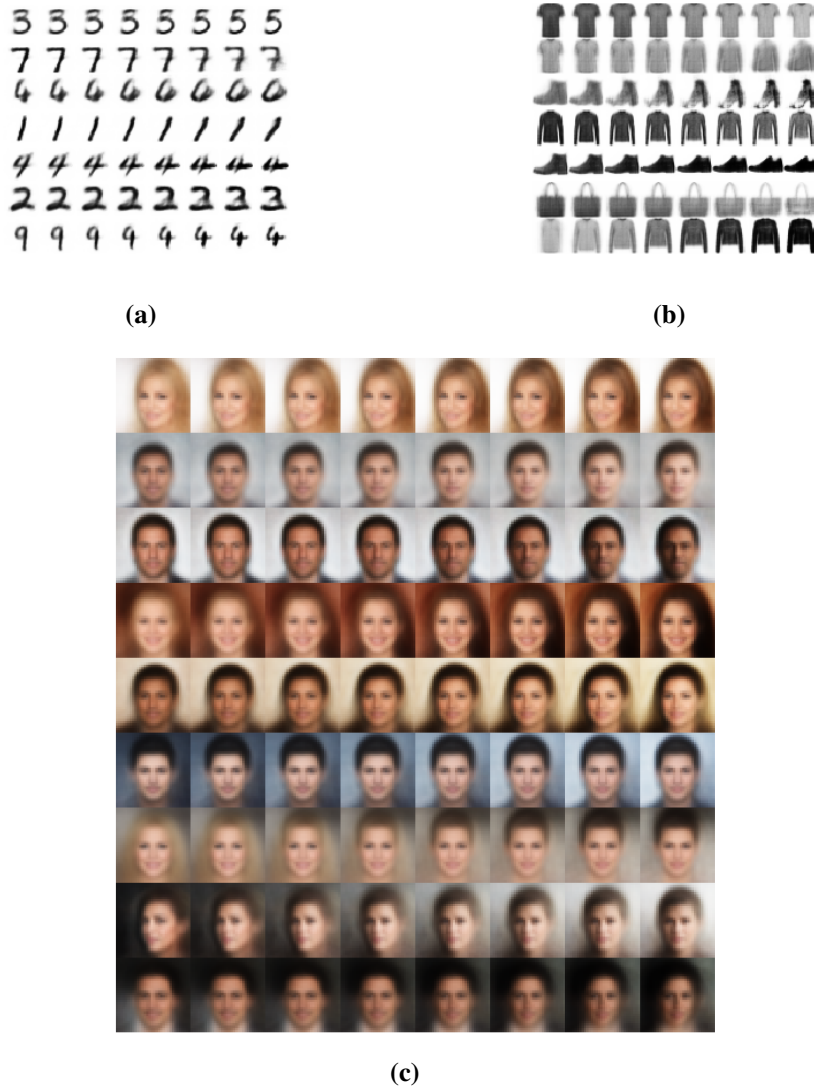


Figure 5: Traversing the latent space of the VSC - $\alpha = 0.01$ model trained for 11 epochs on MNIST (a) and Fashion-MNIST (b) datasets, and 90 epochs on CelebA (c) dataset.

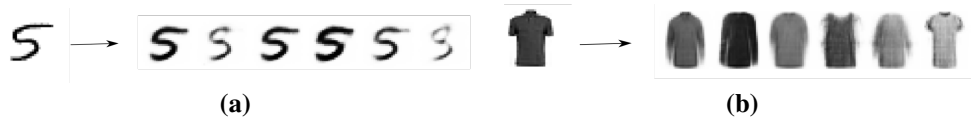


Figure 6: Conditional sampling from VSC - $\alpha = 0.01$ model trained for 11 epochs on MNIST (a) and Fashion-MNIST (b) datasets.

initialization procedure), it is possible to replicate the results reported. Moreover, we are able to validate the authors hypothesis that the VSC models generate sparse, informative and interpretable representations.

Minor discrepancies in the results can be explained by the restricted number of training epochs. In particular, in Figure 1, we observe a similar trend that the one shown in the paper; however, the gap between the curves can be decreased by training the model for larger number of iterations.

Furthermore, in the log optimization files, we can easily notice that by using only 20K iterations the model has not reached a local optimum yet. This situation is also present during training on the CelebA dataset: we noticed that at least 50 epochs are needed to be close to a local optimum, as it is shown in Figure 7. We believe that authors must provide more specifications on the optimization hyperparameters to facilitate the reproducibility task.

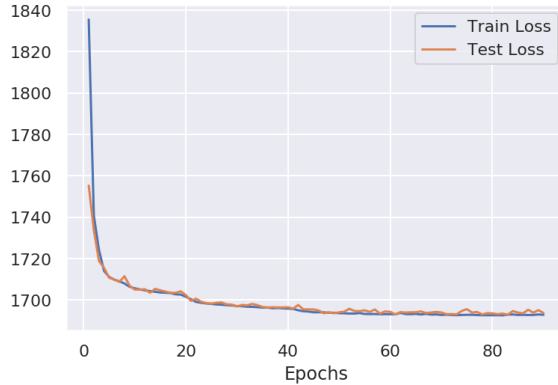


Figure 7: Evaluation of training and test loss of the VSC model on the CelebA dataset

Similarly, for the classification task (Figure 2) although we were able to obtain more informative representations using the VSC model with respect to VAE, the classification accuracy can still be improved by using a higher capacity classification model, instead of the simple classifier used in the original paper.

Many of the reviewers addressed the issue of how we can interpret the learned latent features by the VSC-model. Although authors affirm that the model does not explicitly induce interpretation, it certainly results into a higher expectation of interpretability in large latent spaces, provided that the sources of variations in the observed data can be considered sparse.

To account for the blurriness of generated images (Figures 3 and 5), we must understand that the model capacity is limited and perhaps convolutional architecture could drastically improve the quality of results.

4 CONCLUSIONS

Overall the paper describes in enough detail the VSC model implementation and we must applaud the authors in their ability to convey such a complex topic in an approachable and replicable manner. The hypothesis of using Variational Sparse Coding to obtain sparse, informative and interpretable representations was confirmed by reproducing the experiments.

Further development on testing the model and comparing it against other sparse models on well known benchmarks is critical. In order to assess how *interpretable* the learned latent features are, we could draw ideas from disentangled representations, to measure the effect of sparsity in the disentanglement metric, against benchmark models such as β -VAE or Factor-VAE.

To conclude our reproducibility report we propose the following directions for future research and improvement of current results:

- Increase the number of iterations suggested for training the models and re-run the experiments using the notebooks in our repository.
- In order to improve the quality of generated images, we suggest to expand the model capacity by either stacking more layers or trying convolutional architectures for the encoder and decoder.
- Apply the model to the Disentanglement testing Sprites dataset, to measure the effect of sparsity on a disentanglement metric.

ACKNOWLEDGMENTS

We would like to thank to Emilien Dupont for clarifying distinct aspects on variational autoencoders' implementation, and Lukas Mosser for his suggestions on training generative models.

REFERENCES

- Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β -VAE. apr 2018. URL <http://arxiv.org/abs/1804.03599>.
- Francesco Paolo Casale, Adrian Dalca, Luca Saglietti, Jennifer Listgarten, and Nicolo Fusi. Gaussian process prior variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 10390–10401, 2018.
- Matthew Chalk, Olivier Marre, and Gasper Tkacik. Relevant sparse codes with variational information bottleneck. may 2016. URL <https://arxiv.org/abs/1605.07332>.
- Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.
- Carl Doersch. Tutorial on Variational Autoencoders. jun 2016. URL <http://arxiv.org/abs/1606.05908>.
- I. J. Goodfellow, Aaron Courville, and Yoshua Bengio. Large-Scale Feature Learning With Spike-and-Slab Sparse Coding. *Arxiv*, (1):1–21, jun 2012. ISSN 0511-9618. URL <https://arxiv.org/abs/1206.6407>.
- 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.
- Hemant Ishwaran, J Sunil Rao, et al. Spike and slab variable selection: frequentist and bayesian strategies. *The Annals of Statistics*, 33(2):730–773, 2005.
- Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. *arXiv preprint arXiv:1802.04364*, 2018.
- Hyunjik Kim and Andriy Mnih. Disentangling by Factorising. feb 2018. URL <http://arxiv.org/abs/1802.05983>.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. dec 2013. URL <https://arxiv.org/abs/1312.6114>.
- Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems* 28, pp. 2575–2583. Curran Associates, Inc., 2015. URL <http://papers.nips.cc/paper/5666-variational-dropout-and-the-local-reparameterization-trick.pdf>.
- Matt J Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. *arXiv preprint arXiv:1703.01925*, 2017.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

- Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3730–3738, 2015.
- Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.
- Eric Nalisnick and Padhraic Smyth. Stick-Breaking Variational Autoencoders. may 2016. URL <http://arxiv.org/abs/1605.06197>.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.
- Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. jan 2014. URL <http://arxiv.org/abs/1401.4082>.
- Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.
- Tim Salimans. A structured variational auto-encoder for learning deep hierarchies of sparse features. *arXiv preprint arXiv:1602.08734*, 2016.
- Michalis K Titsias and Miguel Lázaro-Gredilla. Spike and slab variational inference for multi-task and multiple kernel learning. In *Advances in neural information processing systems*, pp. 2339–2347, 2011.
- Francesco Tonolini, Bjorn Sand Jensen, and Roderick Murray-Smith. Variational sparse coding, 2019. URL <https://openreview.net/forum?id=SkeJ6iR9Km>.
- Aaron van den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pp. 6306–6315, 2017.
- Jacob Walker, Carl Doersch, Abhinav Gupta, and Martial Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pp. 835–851. Springer, 2016.
- Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- Serena Yeung, Anitha Kannan, Yann Dauphin, and Li Fei-Fei. Epitomic variational autoencoders. 2016.