# Conditional Subspace Variational Autoencoders for Counterfactual Recourse on Images

Patrick Köhler
*University of Tübingen*
*Student of Machine Learning*
patrick.koehler@student.uni-tuebingen.de

Tobias Leemann
*University of Tübingen*
*PhD Student at Chair for Data Science and Analytics*
tobias.leemann@uni-tuebingen.de

***Abstract*—State of the art generative models exhibit low predictive capabilities for complex, high dimensional data. Trustworthiness in algorithms deployed for certain human in the loop applications can be improved by generating input specific samples, which reveal class related structure to the user.**

**Conditional Subspace Variational Autoencoders (CS-VAE) learn separate latent representations for label and non-label related information in a novel fashion, which could extend a powerful generative model by high predictive capacity.**

**We put the CS-VAE into the context of Counterfactual Recourse, elaborate thoroughly on the setup around Autoencoding Neural Networks and derive the mathematical framework of the CS-VAEs in a lightly accessible fashion.**

**Furthermore we provide practical insight into the behaviour of (vanilla) Variational Autoencoders on MNIST.**

***Index Terms*—Counterfactual Images, Generative Modelling**

## I. Introduction

The application of powerful Machine Learning Algorithms to real world scenarios, e.g. medical diagnosis, is often limited by their trustworthiness. Even though classification algorithms can increase the human performance in diagnosing breast cancer [1] their deployment is rare. In this setting trustworthiness can be increased substantially by providing counterfactual examples. If a physician thinks a tumor is benign (harmless) while an algorithm classifies it to be malign with high confidence, we would like to provide similar looking images which belong to the benign class with high probability.

Additionally we generate samples which look similar and belong to the same class as the image at hand. By comparing the two sets of generated images, the user can derive an explanation for the algorithmic decision. Physicians can be made aware of small effects which are hard to detect for the human eye or rare enough that they have not been investigated and reported comprehensively in the medical literature.

The problem corresponds to sampling points close to the input – from a high dimensional pixel space – but on the other side of a decision boundary. Two major challenges need to be addressed by the model. First, sampling directly from high dimensional distributions without prior knowledge about their structure is extremely inefficient in terms of the amount of required data and computationally not feasible for models without closed form joint distributions [2]. Second, generative models perform far below state of the art classifiers when the fitted class conditional distributions are used to construct a discriminator [4]. Nonetheless we require one model to solve both tasks to produce useful counterfactual examples as the image generation must be conditioned on the label. The first problem can be efficiently approached with lower dimensional representations of the distribution. Assuming that the data generation process is driven by latent variables allows us to fit a distribution without imposing any further restrictions such as explicit parameterizations. It results in a procedure that optimizes distributions directly in function space, which is generally known as Variational Inference (VI). Powerful Deep Learning architectures, e.g. Variational Autoencoders (VAEs) [3], have been developed to leverage the full potential of VI, thus generating images from latent distributions without any prior knowledge about their structure.

Conditional Subspace VAEs (CS-VAEs) [5], a recent extension of VAEs, tackle the second problem by implicitly separating label related structure in its lower dimensional representation. While CS-VAEs have been demonstrated to have semantic generative abilities, such as artificially adding, removing and changing beards and glasses on faces, their discriminatory power is not benchmarked yet. If the explicit constraint to separate class related information improves the predictive performance in comparison to other state of the art generative models, the CS-VAE is a promising architecture to increase explainability and trustworthiness in diagnostic algorithms.

The following section introduces generative neural networks and derives the CS-VAE model. Afterwards, we describe the behaviour of VAEs on MNIST under different architectures and hyper parameters. Finally, distribution approximations in the CS-VAE model are reflected critically and benchmarking experiments are motivated in the context of discriminatory limitations to generative models.

## II. Generative Neural Networks

CS-VAEs extend the popular Variational Autoencoder architecture by explicitly separating label related information in the latent representation. Before explaining the loss function through which this additional structure is imposed on the latent space, Autoencoders in general, and their Bayesian extensions, Variational Autoencoders, are motivated extensively and put into a formal framework.

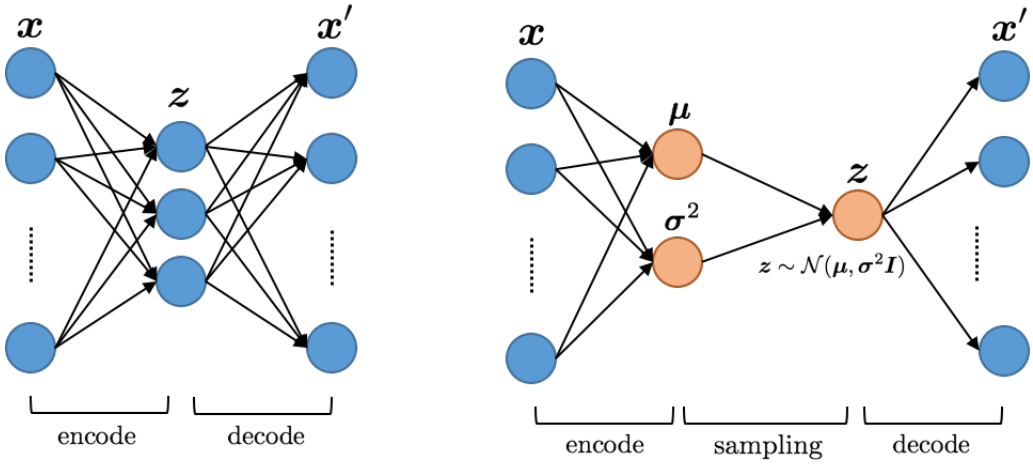


Fig. 1. Autoencoder (left) and Variational Autoencoder (right) architectures. [1]

### A. Autoencoders

*1) Motivation and Architecure:* For high dimensional data, generative algorithms leverage complex abstractions of features, i.e. non-linear functions of the input variables, to compose new samples. Images are usually abstracted in terms of convolutions to generate new images out of corners and edges.

Autoencoders fit this low-dimensional (latent) representation of the input, also referred to as an encoding, by means of a neural network. It is optimized such that another neural network, the decoder, can reconstruct the input from the latent variables. The network architecture is illustrated in Figure 1 on the left side. Note the symmetry of the layer sizes around the central layer, which contains the latent variables as individual nodes. While the first and last layer must contain one neuron per input dimension, the size and type of hidden layers are, as well as the specific loss function, to be set by the user. As a default, one could choose the Mean Square Error between the output of the last layer – the reconstruction – and the input as the loss.

If an Autoencoder reconstructs unseen data points well, we found a function of input variables which represents the entity at hand in terms of meaningful latent variables. Another point of view is that we've found a good compression algorithm.

Since useful latent representations for images arise from convolutions, convolutional hidden layers are used for the encoding and their de-convolutional counterparts for the decoding.

*2) Sampling from Latent Representations:* Each data point corresponds to a point in the vector space spanned by the latent variables, i.e. the activations of the central layer. Thus, sampling a random point in the latent space and decoding it may result in a point from within the original data distribution (in input space).

However, this extension of a data compression to a data generation algorithm comes with challenges. Sampling from the distribution in latent space is particularly hard if irregular regions of the distribution (i.e. regions with high variability) are not covered densely by encoded data points [2]. Furthermore the decoder is not guaranteed to produce reasonable reconstructions for these sparsely covered regions, since those points could have only contributed a minor amount to the loss.

Intuitively speaking, complex entities can be reconstructed best in terms of complex, irregular latent representations. In turn, irregular latent representations require many data points to fit a proper decoder.

Consider two different encodings of the same data set. One maps the data to a highly structured point cloud in latent space, whereas the other produces an irregular distribution of points over its latent space. Two respective empirical probability distributions are visualized in Figure 2. Autoencoders tend to fit latent representations with irregular distributions [7], exacerbating the aforementioned sampling challenge.

Hence, we need to enforce regularity onto the probability distribution over the latent space to enable high generative capabilities.

### B. Variational Autoencoder

We introduce a non-deterministic component into the encoder. Instead of encoding each input by a $d$ dimensional vector (where $d$ is the number of latent variables), we encode a data point through a probability distribution over the latent space. We attach a regularity constraint that the distribution should be as close as possible to a $d$-dimensional standardized Gaussian, where all covariances on the off-diagonal are explicitly set to 0.

*1) Architecture:* An illustration of the architecture is depicted on the right side of Figure 1. The encoding resembles now a mapping from input to parameter space, i.e. to a $2d$-dimensional space with $d$ mean and $d$ variance parameters. A latent, non deterministic representation is sampled from a Gaussian with these parameters and used by the decoder to reconstruct the input.

Since sampling is not differentiable, gradient descent can not be applied to this architecture right away. The training

[1] Taken from https://becominghuman.ai

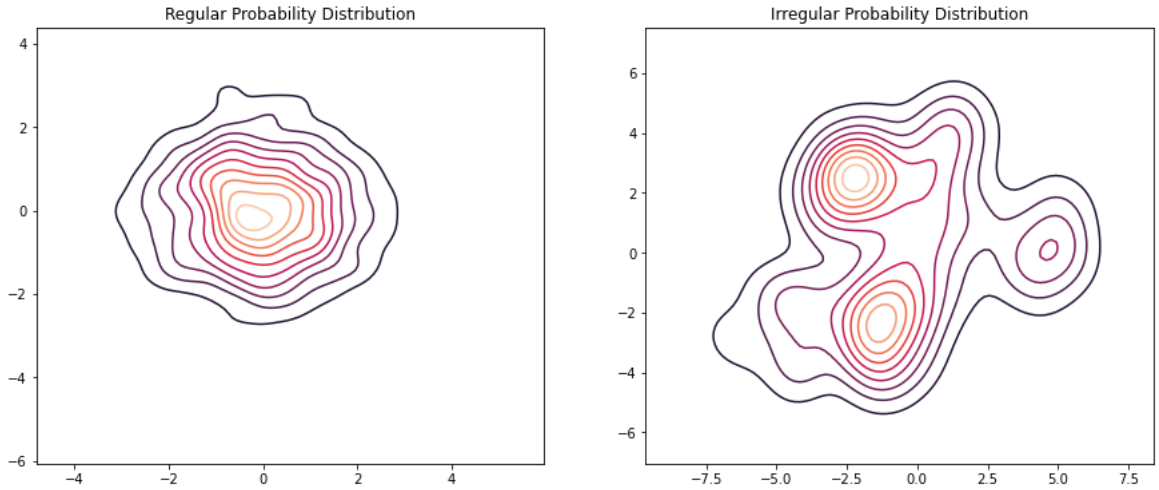


Fig. 2. Regular (left) and Irregular (right) Probability Distributions in 2-Dimensional Latent Space. [2]

is realized through the so called *reparameterization trick*. Introduce an additional input variable

$$\epsilon \sim \mathcal{N}(0, I_d)$$

and parameterize the sample from latent space

$$\mu + \epsilon \odot \sigma^2 = z \sim \mathcal{N}(\mu, \sigma^2 I_d),$$

where $\sigma^2 \in \mathbb{R}^d_+$ denotes the *vector* of variances and $\mu \in \mathbb{R}^d$ the mean vector fit by the encoder. This translation of the random component into an input variable allows the sampling step to compute gradients by Backpropagation.

*2) Loss Function:* Probabilistic Generative Models generally aim to maximize the data log-likelihood $\log p_\theta(\boldsymbol{x})$. For high-dimensional, complex data, this distribution is usually intractable or too expensive to compute thus it can not be optimized directly. However, a lower bound can be derived under a specific setup.

Assume the data generating process can be described sufficiently precise in terms of a vector of latent (unobserved) variables $\boldsymbol{z}$. Then there exists a prior distribution $p_\theta(\boldsymbol{z})$ representing the lower-dimensional counterpart of the data likelihood. Our setup implies that given a point $\boldsymbol{z}$ in latent space there exists a posterior distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ over the input space.

The conditional complement $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ defines our probabilistic encoder. Since this true posterior is not tractable in general, let it be approximated by a parametric distribution

$$p_\theta(\boldsymbol{z}|\boldsymbol{x}) \approx q_\phi(\boldsymbol{z}|\boldsymbol{x}),$$

which is modelled by a neural network with parameters $\phi$. Note that there are no further restrictions to $q$, such as the factorization constraint in classical Variational Inference, which parameterizes the posterior without neural networks [11], allowing the VAE to model a much broader class of posteriors.

[2]Taken from: https://towardsdatascience.com

Consequently, a lower bound on the marginal log likelihood can be approximated in terms of learnable parameters:

$$\log p_\theta(\boldsymbol{x}) \geq \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z})), \tag{1}$$

where $D_{KL}$ denotes the Kullback-Leibler Divergence. On an intuitive level the first term can be interpreted as an average likelihood of all possible reconstructions weighed by the probability density over the latent space the encoder yields. In practice, the expectation will be replaced by a Monte Carlo estimator of choice. For instance the Mean Square Error (MSE) or Binary Cross Entropy are suited for images.

The second term enforces our prior model belief about the form of the data distribution in latent space. As we aim for regular structure, we set $p(\boldsymbol{z}) = \mathcal{N}(\mu, \Lambda)$, where $\Lambda$ is diagonal.

Optimizing the lower bound corresponds to fitting $\theta$, the parameterization of the generative model and $\phi$ the encoding of $\boldsymbol{x}$ in terms of $\boldsymbol{z}$.

Referring to the image generation example, $\boldsymbol{z}$ can be a particular set of convolutions encoding an original image and the posterior $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ assigns a probability density to each image being the reconstruction of this particular encoding.

Using MSE as the reconstruction loss and introducing a weight hyper parameter $\beta > 0$ for the importance of the regularity in encoding space, we obtain the loss function $\mathcal{L}$

$$\mathcal{L}(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x} - \hat{\boldsymbol{x}}_{\boldsymbol{W}}\| + \frac{\beta}{2}\sum_{i=1}^{d}(\sigma_\phi^2 - \log \sigma_\phi^2 - 1 + \mu_\phi^2)_i, \tag{2}$$

where the subscript $_{\boldsymbol{W}}$ denotes that the reconstruction is parameterized by the whole network, whereas the subscript $_\phi$ indicates dependence only on the encoder. The index $i$ accesses the $i$-th entry of the corresponding vectorized operation.

### C. Conditional Subspace Variational Autoencoder (CS-VAE)

Providing counterfactual examples requires us to sample from class-conditional distributions. Colloquially speaking, we

desire the function, which maps an input to the parameters, that define the distribution over the latent space, to take the label into account.

*1) CS-VAE Predecessors: Cond VAE and Cond VAE-info:* Simply adding the label as another input variable to the encoder and decoder results in the Conditional VAE model [8]. To attach more weight to class specific distribution properties an additional network $R$ augments the Conditional VAE, which predicts the label $y$ from $\boldsymbol{z}$, while the encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is trained to minimize the performance of $R$ [9]. This approach, referred to as Conditional VAE-info, aims to remove information from $\boldsymbol{z}$ which is correlated to $y$.

The decoder reconstructs the input from the class-unconditional latent representation $\boldsymbol{z}$ and the *one-dimensional* label $y$. Experiments have demonstrated that the one-dimensional label information is outweighed by the class-unrelated information, leading to suboptimal reconstructions [5].

*2) Implicitly Disentangling the Label in the Latent Space:* To balance the weights of label and non-label related information in latent space, CS-VAEs learn a multidimensional representation of label related information. The architecture is depicted in Figure 3, showing the encoder on the left and the decoder on the right side. The input $\boldsymbol{x}$ encodes two sets of latent variables separating $\boldsymbol{z}$, the label-unrelated from $\boldsymbol{w}$, the label related information which additionally receives $y$ as an input. During training, separation is realized by optimizing $\boldsymbol{z}$ such that $y$ can not be predicted from it (referred to as adversarial training), as indicated by the dotted arrow.

The decoder reconstructs the input from both multidimensional latent variables, without explicitly accessing the value of $y$.

*3) Loss Function:* From the formal perspective, the latent probability space $\mathcal{H}$ is decomposed into two subspaces $\mathcal{Z}$ and $\mathcal{W}$, i.e.:

$$\mathcal{H} = \mathcal{Z} \times \mathcal{W}.$$

Modelling the label information leads requires a joint data likelihood between $\boldsymbol{x}$ and $y$, which is to be optimized with the additional constraint of minimizing the mutual information between $\boldsymbol{z}$ and $y$.

Assuming the model specifications to be true, i.e.:

$$\boldsymbol{z} \perp \boldsymbol{w}, \quad \boldsymbol{z} \perp y, \quad \boldsymbol{x}|\boldsymbol{w} \perp y, \tag{3}$$

the joint log-likelihood of $(\boldsymbol{x}, y, \boldsymbol{w}, \boldsymbol{z})$ can be decomposed with the basic property of factorizing joints from independence statements

$$\begin{aligned}\log p_{\theta,\gamma}(\boldsymbol{x}, y, \boldsymbol{w}, \boldsymbol{z}) = &\log p_\theta(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{z}) + \log p(\boldsymbol{z}) \\ &+ \log p_\gamma(\boldsymbol{w}|y) + \log p(y).\end{aligned}$$

The parameterizations $\gamma$ and $\theta$ will be the subjects of optimization through the neural networks.

Leveraging Jensen's inequality, a lower bound on $\log p_{\theta,\gamma}(\boldsymbol{x}, y)$ can be expressed similar to (1) with an additional term enforcing regularity in the latent space $\mathcal{W}$. Thus an upper bound on $-\log p_{\theta,\gamma}(\boldsymbol{x}, y)$ is given by

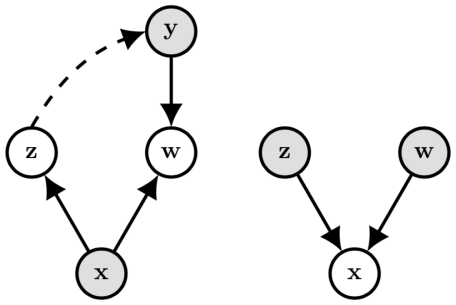


Fig. 3. Conditional Subspace VAE Architecture [5]. Encoder on the left, Decoder on the right side. Shaded nodes show conditioning, dotted arrows represent adverserial training.

$$\begin{aligned}m_1(\boldsymbol{x}, y) = &- \mathbb{E}_{q_\phi(\boldsymbol{z},\boldsymbol{w}|\boldsymbol{x},y}[\log p_\theta(\boldsymbol{x}|\boldsymbol{w}, \boldsymbol{z})] \\ &+ D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}, y)\|p(\boldsymbol{z})) - \log p(y) \\ &+ D_{KL}(q_\phi(\boldsymbol{w}|\boldsymbol{x}, y)\|p_\gamma(\boldsymbol{w}|y)),\end{aligned}$$

where the only difference to the VAE is the last summand and that we account for the label when conditioning on the observation.

The first part of the CS-VAE objective is the probability-weighted average of this data point likelihood over the whole joint data distribution

$$\mathcal{M}_1 = \mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)} m_1(\boldsymbol{x}, y).$$

Recall that the derivation builds on the independence assumptions in (3). Since the independence of $\boldsymbol{z}$ and $y$ is not necessarily satisfied after minimizing $\mathcal{M}_1$, their mutual information ($MI$) is explicitly minimized in the objective function. An approximately equivalent optimization criterion can be expressed as

$$\begin{aligned}&\text{argmin}_{\boldsymbol{z}} MI(\boldsymbol{z}, y) \\ &\approx \text{argmin}_{\phi,\delta} \mathbb{E}_{q_\phi(\boldsymbol{z}\ \boldsymbol{x}),\mathcal{D}(\boldsymbol{x})} \left( \int_y q_\delta(y|\boldsymbol{z}) \log q_\delta(y|\boldsymbol{z}) dy \right) \\ &=: \mathcal{M}_2.\end{aligned}$$

However, computing this expression requires the unknown posterior $q_\delta(y|\boldsymbol{z})$. While the $MI$ objective involves intractable integrals, the posterior is unknown but can be learned. The distribution is, again, fit by the Maximum Likelihood principle, leading to the final term of the CS-VAE loss:

$$\mathcal{N} = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x}),\mathcal{D}(\boldsymbol{x},y)} q_\delta(y, \boldsymbol{z}).$$

Finally, introduce hyperparameters $\beta_i$ as weights for all three training objectives and obtain the complete optimization problem as

$$\min_{\theta,\phi,\gamma} \beta_1 \mathcal{M}_1 + \beta_2 \mathcal{M}_2 \quad \wedge \quad \max_{\delta} \beta_3 \mathcal{N}.$$

In summary, $M_1$ optimizes the unconditional joint likelihood, while $M_2$ and $\mathcal{N}$ separate label related from non

label related information in the latent space. The involved distributions $q_\delta(y|\boldsymbol{z})$ and $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ are parameterized through the adversarial architecture in the encoder. While $q_\delta(y|\boldsymbol{z})$ is optimized through $\mathcal{N}$ to predict $y$ from $\boldsymbol{z}$, optimizing $\mathcal{M}_2$ at the same time attempts to produce $\boldsymbol{z}$ which precludes a good prediction.

## III. VAEs on MNIST

We implemented and trained the plain Variational Autoencoder on MNIST [10] to inspect its behaviour under different hyper parameter settings.[3]

*1) General Observations:* While the reconstructions already converge towards very good results after few ($\sim$5) epochs for most hyper parameter settings, generating good samples requires more fine tuning. In contrast to supervised problems, we observe that the generative performance generally still increases after the test set loss reached its minimum. All reconstructions and samples show rather blurry contours independent of the quality of their shape.

*2) Architecture Considerations:* Model 1 uses two hidden linear layers of size $d^2$ for $d = 20$ latent dimensions between the input and the sigmoid-activated output layer. Recall that the latent representation in $d$ dimensions requires $2d$ nodes in the central layer, since each dimension is defined by one mean and one variance parameter.

Its extension, Model 2, consists of two convolutional layers, employing 16 and 32 kernels respectively, each of size $5 \times 5$. The stride is set to 1. A linear layer maps the output of the second convolution to the latent representation of size $d = 20$.

Convolutions contribute to major improvements in the sample generation even though we still observe some discontinuous strokes and unnatural rotations of the digits. Increasing the latent dimension in Model 2 to $d = 128$ seems to cause minor improvements, however both models produce images which are not identifiable as digits.

Attaching more weight to the regularity of the latent distributions by setting $\beta = 3$ instead of the default $\beta = 1$, produces the best samples where all generated images resemble handwritten digits and only exhibit minor artifacts.

Figure 4 contrasts two sample generations from Model 1 and the tuned Model 2.

This demonstrates that $\beta$ needs to be optimized depending on the architecture and the number of latent dimensions, even though the original derivation [3] does not include this parameter.

On an intuitive level this observation can be explained by the difference in magnitude of the two terms contributing to the loss. If the reconstruction loss outweighs the KL-divergence because they are on different scales, the gradients will always point into a direction which only decreases the reconstruction loss, even when the relative decrease is negligibly small. The different magnitudes of the regularization and reconstruction terms depend on the choice of the reconstruction loss itself (e.g. squared vs. absolute differences) as well as on the number of latent dimensions, since each dimension contributes a non negative value to the loss (cf. eq. 2).

While model evaluation can be performed on this rather heuristic level for handwritten digits, it may require large efforts for data which are not as easy to interpret. Comparing performance metrics is not straight forward as their magnitudes depend on the model architecture.

*3) Implementation Details:* All neurons use ReLu activations; *Adam* with a learning rate of $10^{-4}$ is chosen as the optimization algorithm. Binary Cross Entropy is set as the reconstrucion loss. The data was processed in batches of size 256. Training could be completed in about 45 minutes on a Mid 2013 MacBook Air with 1,3 GHz Intel Core i5 CPU and 4GB of Memory.

## IV. Conclusion and Outlook

While the CS-VAE has been demonstrated to perform well for Attribute and Style Transfer in images [5], its discriminatory power has not been investigated yet.

Generative models usually arise from directly minimizing class conditional log-likelihoods. These models exhibit poor predictive capacities because the low dimensional label carries a negligible amount of bits to highlight structural differences in the class conditional distributions [4].

Tuning the dimensionality of the label related latent representation $\boldsymbol{w}$ could allow CS-VAEs to circumvent this problem.

However, extensive benchmarking (i.e. thorough hyper parameter optimization) for predictive performance is necessary to further assess their potential for Counterfactual Recourse.

If the predictive performance is comparable to state of the art classifiers, calibration of the uncertainty along the decision boundary is to be investigated either on a theoretical or on a use case specific basis.

Recall that the posteriors over latent representations can only be approximated. Without statements about the approximation quality, out-of-distribution generative artifacts and locally poor predictions can not be prevented, which might limit the trustworthiness in medical scenarios.

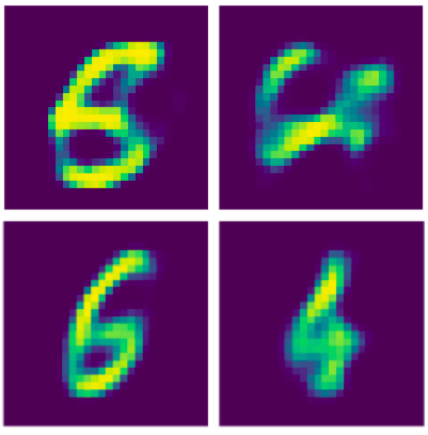

Fig. 4. Samples from Model 1 (two linear layers of size $d^2$ with $d = 20, \beta = 1$) in the upper row and tuned Model 2 (two convolutional + one linear layer with $d = 128$ and $\beta = 3$) in the lower row.

[3] https://github.com/pat-rig/csvae4classification

## REFERENCES


[1] Wang, D., Khosla, A., Gargeya, R., Irshad, H., and Beck, A. H. (2016). "Deep learning for identifying metastatic breast cancer." arXiv preprint arXiv:1606.05718.

[2] Bishop, C. M. (2006). "Pattern recognition and machine learning." Springer.

[3] Kingma, D. P., and Welling, M. (2014). "Auto-encoding variational bayes." ICLR 2014.

[4] Fetaya, E., Jacobsen, J. H., Grathwohl, W., and Zemel, R. (2020). "Understanding the limitations of conditional generative models." ICLR 2020

[5] Klys, J., Snell, J., and Zemel, R. (2018). "Learning latent subspaces in variational autoencoders." NeurIPS 2018.

[6] Watt, N., Du Plessis, M. (2020). Towards robot vision using deep neural networks in evolutionary robotics. Evolutionary Intelligence. 10.1007/s12065-020-00490-w.

[7] Sankaran, A., Vatsa, M., Singh, R., Majumdar, A. (2017). Group sparse autoencoder. Image and Vision Computing, 60, 64-74.

[8] Kingma, D.P., Mohamed, S., Rezende, D.J. and Welling, M. Semi-supervised learning with deep generative models. In Advances in Neural Information Processing Systems, pages 3581–3589, 2014.

[9] Creswell, A., Bharath, A. and Sengupta, B. (2017). Conditional autoencoders with adversarial information factorization. arXiv preprint arXiv:1711.05175

[10] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6), 141–142.

[11] Hoffman, M. D., Blei, D. M., Wang, C., Paisley, J. (2013). Stochastic variational inference. Journal of Machine Learning Research, 14(5).