Automatic selection of latent variables in variational auto-encoders

Emma Jouffroy *†, Audrey Giremus †, Yannick Berthoumieu †, Olivier Bach *, Alain Hugget *
*CEA, Le Barp, FRANCE
†University of Bordeaux - INP Bordeaux ENSEIRB-MATMECA - IMS - UMR CNRS 5218, Talence, FRANCE

***Abstract*—Variational auto-encoders (VAEs) are powerful generative neural networks based on latent variables. They aim to capture the distribution of a dataset, by building an informative space composed of a reduced number of variables. However, the size of this latent space is both sensitive and difficult to adjust. Thus, most state-of-the-art architectures experience either disentanglement issues, or, at the opposite, posterior collapse. Both phenomena impair the interpretability of the latent variables. In this paper, we propose a variant of the VAE which is able to automatically determine the informative components of the latent space. It consists in augmenting the vanilla VAE with auxiliary variables and defining a hierarchical model which favors that only a subset of the latent variables are used for the encoding. We refer to it as NGVAE. We compare its performance with other auto-encoder based architectures.**

***Index Terms*—deep neural networks, variational inference, generative models, unsupervised models**

I. Introduction

Variational autoencoders (VAEs) are generative neural networks based on latent variables that are widely used for many unsupervised complex tasks in the field of machine learning, such as image or text generation. They consist of a first network, called the encoder, that performs dimensionality reduction and yields a compressed representation of the input data in a latent space. The extracted features should capture the generative factors of the data. Then, they serve as inputs of a decoder used for reconstruction. The training is performed by minimizing a loss function usually defined as the evidence lower bound (ELBO). One of the major challenge of current research is to build an interpretable representation of the data within the latent space [1]. To this end, it is crucial to properly set its dimension. A too small one results in correlated variables whereas disentanglement is desired. The latter refers to the case where the encoding variables are independent from one another, each standing for a different generative factor of the input data. Recent studies prove that it significantly improves many common tasks in deep learning or representation learning [2] and allows to perform uncertainty studies on the learned data. On the contrary, too many latent variables lead to the posterior collapse phenomenon. It occurs when the learned posterior distribution of the latent variables conditional upon the input data matches the prior one. In this case, the latent space do not contain any information and the inference model generate images that are mostly an averaging of the dataset [3]. To overcome this difficulties, different strategies can be considered. In [4] and [5], it is proposed to add inductive biases intrinsic to some generative factors, whereas in [6], the objective function is modified. Some authors also enrich the prior distribution [8] [9] [10]. However, most papers are interested in the quality of the image generation without taking into account the level of interpretability of the latent variables. To better balance both objectives, [7] proposes to regularize the loss function by enforcing more constraints on the latent space. For that purpose, the similarity term between the learned posterior law and the prior distribution, usually defined by a multivariate centered Gaussian distribution with identity covariance matrix, is weighted. The choice of the weighting factor is intricate and dataset-dependent. As an alternative, considering more complex prior distributions, as it is the case in hierarchical latent models [9], results in getting more information in the latent variables but induces in return correlations between them.

The objective of our work is to propose a learning model based on the vanilla VAE that automatically divides the latent space in informative and uninformative components. We refer to this new model as NGVAE for "Normal-Gamma Variational Auto-Encoder". It takes advantage of the fact that informative latent variables are associated to small inferred variances contrary to the ones that experience posterior collapse. The principle is then to add a level of hierarchical knowledge about the variances which become stochastic. They are assumed to be distributed according to mixtures of two Inverse-Gamma laws that enforce opposite behaviors, i.e. low or high values. The probabilities of the mixtures are provided by the encoding network, jointly with the mean of the encoding variables. This enriched model favors the information to be carried by only a subset of the latent space that is well-identified.

The remainder of the paper is organized as follows. In part II, we review the standard VAE and its theoretical foundations. Section III is dedicated to the proposed methodology and section IV presents some experimental results. The developed model is tested on a dataset of ours and compared with state-of-the-art approaches based on variational inference.

The following notations are used throughout the paper: $\mathcal{N}(x;\mu,\sigma^2)$ denotes the Gaussian probability density function (pdf) with mean μ and variance σ^2, $\mathcal{G}(x;\alpha,\beta)$ the Gamma pdf with hyperparameters (α,β) and $\mathcal{NG}(x,\lambda;\mu,\alpha,\beta) = \mathcal{N}(x;\mu,\lambda^{-1})\mathcal{G}(\lambda;\alpha,\beta)$ the Normal-Gamma pdf.

II. Variational autoencoders

Variational auto-encoders find their origin in the principle of variational inference. They consist in learning a structured representation of input data $\boldsymbol{x} \in \mathbb{R}^D$ by leveraging the high predicting capacity of neural networks. The rationale behind this approach is to decompose the data distribution using a set of hidden variables forming the latent space and stored in a vector $\boldsymbol{z} \in \mathbb{R}^K$:

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z} = \int p(\boldsymbol{x}|\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}. \quad (1)$$

The true conditional distribution $p(\boldsymbol{z}|\boldsymbol{x})$ of the latent variables, called the posterior law, is usually intractable. It is therefore estimated by a distribution $q(\boldsymbol{z})$ [1] that minimizes a similarity measure often taken as the Kullback-Leibler (KL) divergence. By taking advantage of the following equality

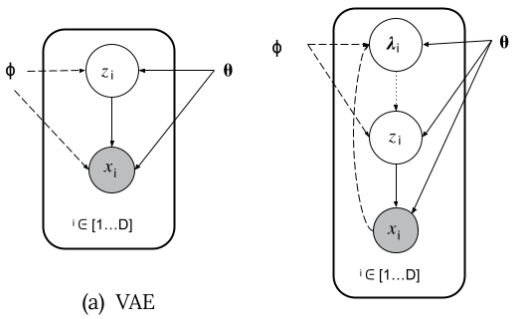
$$\log p(\boldsymbol{x}) = KL\left[q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})\right] + \int \log p(\boldsymbol{x}, \boldsymbol{z}) q(\boldsymbol{z}) d\boldsymbol{z}, \quad (2)$$

it can be noted that minimizing the KL-divergence between the approximate and the true posterior pdfs is equivalent to maximizing the second term of (2) which is the ELBO. By using the mean-field approximation, it can be decomposed as follows:

$$ELBO = \mathbb{E}_{\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_{i=1}^{D} \log p(x_i|\boldsymbol{z})\right] - \sum_{k=1}^{K} KL[q(z_k|\boldsymbol{x})||p(z_k)] \quad (3)$$

where x_i stands for the i^{th} component of the input data and z_k for the k^{th} latent variable.

The emergence of deep learning made it possible to express the unknown distributions in (3) as complex parametric functions of the input data. In this way, the inference problem amounts to recover network parameters that are shared across a whole dataset, making it amortized.



(a) VAE (b) NG-VAE (ours)

Fig. 1: (a) Graphic representation of amortized inference in the context of variational-autoencoder. (b) Graphic representation of the NGVAE model.

Building upon this principle, a vanilla-VAE is composed of two neural networks. The first one, the encoder, yields the mean and the variance of the approximated posterior distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ which depends on a vector $\boldsymbol{\phi}$ that gathers all the variables to be learned. This pdf is used to sample the latent variables that are then processed by a decoder network. The latter finally provides the mean of the likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ as a function of a vector of parameters $\boldsymbol{\theta}$. This distribution can be used to simulate reconstructed data. The relationships between the different quantities are illustrated in Fig. 1.

Most of the time, both the encoder and the decoder consist of a series of convolutional layers. Their parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are adjusted by minimizing the opposite of the ELBO using stochastic gradient descent algorithms. In most common cases, both the posterior and the likelihood distributions are defined as a product of Normal laws. As for the prior $p(\boldsymbol{z})$, it is chosen as a Gaussian distribution with identity covariance, resulting in a closed-form calculus of the ELBO.

III. The Normal-Gamma variational auto-encoder

The above-mentioned choice of prior distribution may be restrictive. Indeed, when the dimension of the latent space exceeds the actual number of generative factors of the input data, two types of behaviors can be observed. On the one hand, the variables carrying information are associated with very low learned variance values. On the other hand, the others experience posterior collapse and exhibit higher variances. Thus, adding prior knowledge about these variances makes sense to automatically bring out relevant components within the latent space. For that purpose, the proposed architecture extends the latter with the variances that are no longer assumed deterministic. They are assigned distributions, the parameters of which depend on the probabilities for the corresponding variables to be informative. These probabilities are obtained as additional outputs of the encoder and directly yield a partition of the latent space in two classes.
The developed models are detailed hereafter. For the sake of simplicity in the derivations, we infer the inverse-variances of the latent variables instead of the variances. They are gathered in a vector $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_K)$ so that the encoding variables become the pair of vectors $(\boldsymbol{z}, \boldsymbol{\lambda})$.

The encoder: In the proposed architecture, the hidden space comprises both the latent variables and their inverse-variances. The specificity of our modeling lies in the chosen posterior distribution for the couples (z_k, λ_k), with $k \in [\![1, K]\!]$. It is defined as a mixture of Normal-Gamma laws:

$$\begin{aligned} p_{\boldsymbol{\phi}}(z_k, \lambda_k|\boldsymbol{x}) =& p_k \mathcal{NG}(z_k, \lambda_k; \mu_k, \alpha_1, \beta_1) \\ &+ (1 - p_k) \mathcal{NG}(z_k, \lambda_k; \mu_k, \alpha_2, \beta_2) \end{aligned} \quad (4)$$

where the means $\{\mu_k\}_{k=1,...,K}$ and the probabilities $\{p_k\}_{k=1,...,K}$ are both provided by the encoder. As for the sets of hyperparameters (α_1, β_1), respectively (α_2, β_2), they are defined to favor values close to 1, respectively 0, for λ_k^{-1}.

However, such a theoretical model is not well-suited for training by gradient descent. To avoid differentiability issues, the straight-cut dependency between the probabilities $\{p_k\}_{k=1,...,K}$ and the Normal-Gamma parameters (α_1, β_1) and (α_2, β_2) must be relaxed. In practice, we thus propose to

[1] In the sequel, the notation p is used for the actual distributions of the variables and conversely q for the estimated ones

associate each couple of latent variables with hyperparameters $(\alpha(p_k), \beta(p_k))$ that continuously depend on p_k as follows:

$$\begin{aligned}\alpha(p_k) =& f_r(p_k) \\ =& \frac{\alpha_2 - \alpha_1}{1 + e^{-r(p_k - 0.5)}} + \alpha_1.\end{aligned} \tag{5}$$

A similar function as (5) is considered for $\beta(p_k)$. The parameter r allows to adjust the slope of the sigmoid $f_r(p_k)$ so that it acts as a threshold.

The decoder: The generative part of the NGVAE aims to model the likelihood distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$. It is not impacted by the modified architecture since it does not leverage the information from the variances. In the case when the inputs are gray-scaled or colored pictures, the likelihood is usually defined as a product of independent Normal distributions $p_\theta(x_i|\boldsymbol{z}) = \mathcal{N}(x_i|m_i(\boldsymbol{\theta}, \boldsymbol{z}), \sigma^2)$ where the mean parameter m_i is the output of the decoder network and the standard deviation $\boldsymbol{\sigma}$ is an hyperparameter.

The objective loss: Training the NGVAE model consists in optimizing the variational parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ by maximizing a ELBO loss using a stochastic gradient descent algorithm. The enriched latent space results in a modified ELBO expressed as:

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = -\mathbb{E}_{\sim q_\phi(\boldsymbol{z}, \boldsymbol{\lambda}|\boldsymbol{x})} \left[\sum_{i=1}^{D} \log p_\theta(x_i|\boldsymbol{z}) \right] \\ + \sum_{k=1}^{K} KL\left[q_\phi(z_k, \lambda_k|\boldsymbol{x})||p(z_k, \lambda_k)\right]\end{aligned} \tag{6}$$

where it is worth noticing that the prior distributions are now defined for the pairs (z_k, λ_k). They are also chosen to be Inverse-Gamma pdfs with hyperparameters (μ, α, β) tuned to obtain non-informative laws. Finally, using Bayes'rule, the KL-divergence can be split in two terms:

$$\begin{aligned}KL\left[q_\phi(z_k, \lambda_k|\boldsymbol{x})||p(z_k, \lambda_k)\right] = \\ \mathbb{E}_{\sim q_\phi(\boldsymbol{\lambda}|\boldsymbol{x})}\left[KL[q_\phi(z_k|\lambda_k)||p(z_k|\lambda_k)]\right] \\ + KL[q_\phi(\lambda_k|\boldsymbol{x})||p(\lambda_k)].\end{aligned} \tag{7}$$

The loss (6) can be computed analytically. The first component assesses the quality of the reconstruction. The term inside the expectation can be expressed as the sum squared error between the input and the output data, weighted by the variance of the assumed Gaussian likelihood. Up to an additive constant, it writes:

$$NLL = \frac{1}{2\sigma^2} \sum_{i=1}^{D} \left(x_i - m_i(\boldsymbol{\theta}, \boldsymbol{z})\right)^2. \tag{8}$$

The KL-divergence terms in (6) admit closed-form expressions. They can be derived by taking advantage of the formulas of the KL-divergence between two gaussian pdfs and two gamma pdfs, respectively. The ones related to the Gaussian distributions become:

$$\begin{aligned}\mathbb{E}_{\sim q_\phi(\boldsymbol{\lambda}|\boldsymbol{x})} \left[\sum_{k=1}^{K} KL[q_\phi(z_k|\lambda_k)||p(z_k|\lambda_k)] \right] \\ = \sum_{k=1}^{K} \frac{1}{2} \frac{\alpha(p_k)}{\beta(p_k)} (\mu - \mu_k)^2.\end{aligned} \tag{9}$$

As for the terms corresponding to the Gamma distributions, they can be developed as follows:

$$\begin{aligned}KL[q_{\boldsymbol{\phi}}(\lambda|\boldsymbol{x})||p(\lambda)] = \sum_{k=1}^{K} \Bigg[\alpha \log \frac{\beta(p_k)}{\beta} - \log \frac{\Gamma(\alpha(p_k))}{\Gamma(\alpha)} \\ + (\alpha(p_k) - \alpha)\Psi(\alpha(p_k)) - (\beta(p_k) - \beta)\frac{\alpha(p_k)}{\beta(p_k)} \Bigg]\end{aligned} \tag{10}$$

where $\Gamma(x)$ denotes the gamma function, and $\Psi(x)$ the digamma function.

The training: Training our model implies performing a gradient descent on functions involving expectations with respect to distributions depending on the unknown parameters. This dependency makes it impossible to resort to sample approximations as required to apply a stochastic optimization procedure. As we depart from the classical Gaussian assumption, the standard reparameterization trick cannot be applied. We thus developed a new one. First, let us consider the gradient to be backpropagated:

$$\begin{aligned}\boldsymbol{\nabla}_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\theta}) = -\nabla_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathbb{E}_{\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{\lambda}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] \\ + \nabla_{\boldsymbol{\phi}} KL[q_{\boldsymbol{\phi}(\boldsymbol{z}, \boldsymbol{\lambda}|\boldsymbol{x})||p(\boldsymbol{z}, \boldsymbol{\lambda})]}.\end{aligned} \tag{11}$$

Since the KL-divergence term between two Normal-Gamma distributions can be computed analytically, the second term does not raise difficulties for the calculation of the gradient. However, the expectation over the likelihood distribution is more problematic and requires a specific reparametrization over the $\boldsymbol{z}$ and $\boldsymbol{\lambda}$ stochastic vectors. By denoting $\boldsymbol{v} = (\boldsymbol{\phi}, \boldsymbol{\theta})$ the variational parameters and $f(\boldsymbol{z}, \boldsymbol{\lambda}) = -\log p_\theta(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\lambda})$, we propose to consider the following change of variables:

$$\begin{aligned}\boldsymbol{z} &= T_2(\boldsymbol{\epsilon_2}; \boldsymbol{v}) = \boldsymbol{m}(\boldsymbol{v}) + (T_1(\boldsymbol{\epsilon_1}; \boldsymbol{v}))^{-\frac{1}{2}} \boldsymbol{\epsilon_2} \\ \boldsymbol{\lambda} &= T_1(\boldsymbol{\epsilon_1}; \boldsymbol{v})\end{aligned} \tag{12}$$

where the vector $\boldsymbol{m}(\boldsymbol{v})$ stores the means computed by the decoder and $\epsilon_2 \sim \mathcal{N}(0, \mathrm{I})$. As for the definition of T_1, we follow [11] so that the k^{th} component of the vector $\boldsymbol{\epsilon}_1$ satisfies:

$$\boldsymbol{\epsilon}_{1,k} = \frac{\log(\lambda_k) - \Psi(\alpha(p_k)) + \log(\beta(p_k))}{\sqrt{\Psi_1(\alpha(p_k))}} \tag{13}$$

where λ_k, $\alpha(p_k)$ and $\beta(p_k)$ implicitly depend on $\boldsymbol{v}$. It should be noted that this reparameterization has the advantage of yielding a vector $\boldsymbol{\epsilon}_1$ only weakly dependent on the parameters $\boldsymbol{v}$. The expectation in (11) can now be rewritten as a function

of the new variables $\boldsymbol{\epsilon}_1$ and $\boldsymbol{\epsilon}_2$. Its gradient, denoted ∇NLL, becomes:

$$\begin{aligned}\nabla\text{NLL} &= \nabla_{\theta,\phi}\int_{\boldsymbol{\epsilon_1}}\int_{\boldsymbol{\epsilon_2}} g(\boldsymbol{\epsilon_1},\boldsymbol{\epsilon_2};\boldsymbol{v})q(\boldsymbol{\epsilon_2},\boldsymbol{\epsilon_1};\boldsymbol{v})d\boldsymbol{\epsilon_2}d\boldsymbol{\epsilon_1} \\ &= \int_{\boldsymbol{\epsilon_1}}\int_{\boldsymbol{\epsilon_2}} \nabla_{\theta,\phi} g(\boldsymbol{\epsilon_1},\boldsymbol{\epsilon_2};\boldsymbol{v})q(\boldsymbol{\epsilon_2},\boldsymbol{\epsilon_1};\boldsymbol{v})d\boldsymbol{\epsilon_2}d\boldsymbol{\epsilon_1} \\ &+ \int_{\boldsymbol{\epsilon_1}}\int_{\boldsymbol{\epsilon_2}} g(\boldsymbol{\epsilon_1},\boldsymbol{\epsilon_2};\boldsymbol{v})\nabla_{\phi,\theta} q(\boldsymbol{\epsilon_2},\boldsymbol{\epsilon_1};\boldsymbol{v})d\boldsymbol{\epsilon_2}d\boldsymbol{\epsilon_1},\end{aligned} \tag{14}$$

where $g(\boldsymbol{\epsilon_1},\boldsymbol{\epsilon_2};\boldsymbol{v}) = f(T_2(\boldsymbol{\epsilon_2};\boldsymbol{v}), T_1(\boldsymbol{\epsilon_1};\boldsymbol{v}))$. Then, the technique of the score function [11] can be applied to the second integral. It ensues:

$$\begin{aligned}\nabla\text{NLL} =& \mathbb{E}_{q(\boldsymbol{\epsilon_1},\boldsymbol{\epsilon_2};\boldsymbol{v})}[\nabla_{\phi,\theta} g(\boldsymbol{\epsilon_1},\boldsymbol{\epsilon_2};\boldsymbol{v})] \\ &+ \mathbb{E}_{q(\boldsymbol{\epsilon_1},\boldsymbol{\epsilon_2};\boldsymbol{v})}\left[g(\boldsymbol{\epsilon_1},\boldsymbol{\epsilon_2};\boldsymbol{v})\nabla_{\phi,\theta}\log q(\boldsymbol{\epsilon_1};\boldsymbol{v})\right].\end{aligned} \tag{15}$$

This final expression makes it possible, as expected, to replace the expectations by empirical means in the stochastic gradient descent.

IV. Experiments

The ability of the proposed model to partition the latent space is tested on a synthetic image dataset. Each image consists in a 2D sprite simulated from five ground truth generative factors which are the color, the shape, the position in two dimensions and the depth, as it can be seen in Fig. 2. In this section, we first present the experimental setup. Then, we provide a set of quantitative and qualitative results so as to get insights about the contribution of the NGVAE compared to state-of-the-art variational inference techniques: a vanilla VAE [12] as well as a β-VAE with $\beta = 27$ and $\beta = 150$ [7]. Regarding these values, $\beta = 27$ has been determined based on the optimal choice suggested in [7].

Fig. 2: Example of reconstructed images after training the NGVAE.

Experimental setup: For fair comparisons, all the encoders and decoders share the same architectures. Therefore, every mapping is composed of 4 convolutional layers with a stride of 2 and a kernel size of $(3, 3)$. The first 2 layers of the encoders contain 32 filters, and the last ones 64 filters. The decoders are built as the reverse of the encoders. For both networks, the Leaky Rectified Linear Unit is applied as activation function. Finally, the standard deviation of the likelihood distribution is set to 0,02 and the latent space dimension to 15. We use the Adam algorithm as an optimizer with a learning rate of 10^{-3} and train each model until the training process reaches the same stopping criterion. When the resulting losses of the five last epochs do not vary over 1%, the learning rate is divided by ten until the next five epochs are stable. The implementation is carried out using Tensorflow framework.
Concerning the NGVAE hyperparameters, we set those of the prior Normal-Gamma distribution to $(0, 8.25, 3.62)$ and those of the mixture of Gamma laws $(\alpha_1,\ \beta_1)$ and $(\alpha_2,\ \beta_2)$ to $(3.65,\ 0.24)$ and $(902,\ 810.9)$, respectively. These have been defined in a *ad-hoc* way until the latter distributions match our assumptions.

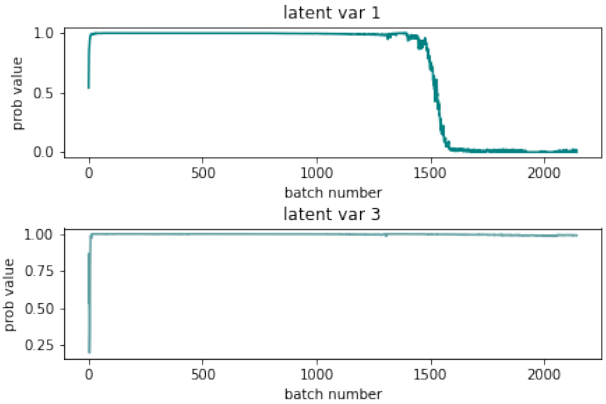


Fig. 3: Mixture probability inferred by the encoder during training for the first and the third latent variables.

Qualitative analysis: In the first place, we explore the ability of the NGVAE to properly separate the latent space in informative and uninformative components by visualizing the mixture probabilities inferred by the encoder for each latent variable all along the epochs as in Fig. 3. We recall that when p_k is close to 0, the k^{th} variable is associated with a low variance, and is therefore informative. It can be observed that it takes several epochs for the probabilities to converge. During the first ones, the reconstruction term is prevalent and the selected posterior distribution is the one that *a priori* minimizes the KL-divergence. It corresponds to a mixture probability close to 1. Then, when the reconstruction term has difficulties in improving, the model starts giving to some components more information than to the others. After convergence, the mixture probabilities make the identification of the relevant latent variables possible. For the conducted experiments, the NGVAE identifies 7 informative components within the latent space.

For the purpose of refining the analysis, we also display the empirical normalized covariance matrix of the inferred latent variables over the whole dataset, as presented in Fig. 4. In this way, we are able to visualize both which components are impacted by the diversity of the dataset and to which extent they are correlated with one another. The results confirm our initial guess: the vanilla-VAE tends to spread information over the whole latent space and there is no disentanglement. Regarding the β-VAE architecture, the outcomes are intrinsically linked with the value chosen for β. Indeed, with $\beta = 27$, the results are similar to those of the vanilla-VAE. However, with $\beta = 150$, the model is able to emphasize a set of 7 informative variables. However, the covariance matrix shows that all the latent variables are impacted by the dataset diversity. Furthermore, the choice of β is directly related to the dimensions of the latent space and the data, which can be problematic for real-world applications.

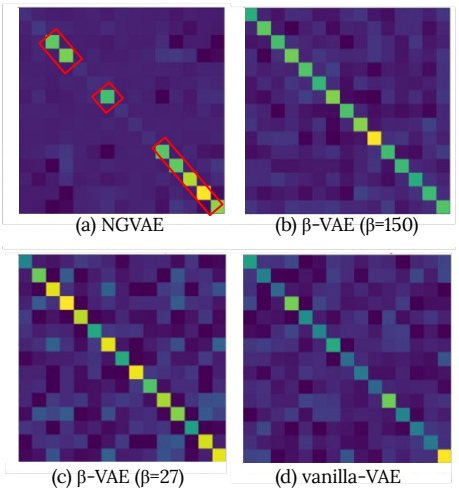


Fig. 4: Scaled empirical covariance matrices of the latent variables over the whole dataset. The colors getting closer to yellow stand for values close to 1. Conversely, the others tend to 0.

Finally, the NGVAE appears to focus the information on a reduced subset of latent variables that are well decorrelated with the others.

Quantitative comparison: To quantitatively compare the methods, we introduce two metrics based on the empirical covariance matrix described above. The first one is a measure of the entropy of the diagonal terms, which is minimal when few components exhibit high values and encode information. The second one evaluates the decorrelation between the latent variables by calculating the ratio between the trace of the matrix and the sum of its non-diagonal values. In addition to these metrics, we also report in Table I a disentanglement metric described in [7] and the number of estimated informative components. The vanilla and the β-VAE are not designed to select latent variables. However, for comparison purposes, we apply a rule-of-the-thumb: a variable with an inferred variance close to one is thus considered as uninformative. The considered performance indicators are respectively denoted in Table I : "entropy", "decorrelation", "disentanglement" and "information".

The obtained results confirm the previous observations. Indeed, between the four models, the NGVAE proves to get the more uncorrelated latent space and the more parsimonious one. It is also the only model that explicitly identifies the informative variables. The vanilla-VAE and β-VAE with $\beta = 27$, for their parts, infer correlated latent space. Concerning the β-VAE with $\beta = 150$, an interesting separation is made between informative and non informative components but the encoded variables remain very sensitive to the dataset diversity. This results in a higher entropy of the matrix covariance diagonal vector. However, none of these architectures is able to perform a good disentanglement even if the NGVAE is slightly better. This can be explained by the fact that they do not include inductive biases related to the generative factors of the dataset.

TABLE I: Quantitative analysis.

Metrics	***vanilla***	***β-VAE(27)***	***β-VAE(150)***	***NGVAE***
entropy	2.6	2.6	2.6	**2.3**
decorrelation	12.7	7.8	11.9	**14.3**
disentanglement	0.67	0.64	0.62	**0.68**
information	15	15	**7**	**7**

V. Conclusion and Discussion

A novel VAE architecture is presented that automatically defines the number of informative variables in the latent space. It consists in randomizing the variances of the VAE posterior distributions and assigning them a mixture model that favors either low or high values. In our experiments, the number of relevant components closely matches the number of ground truth generative factors without the need of adjusting hyperparameters that can be dataset dependent. This makes our model better suited to real-world applications. The limitations of the NGVAE reside in the fact that the required reparameterization trick for the training involves the computation of a Jacobian matrix over the batches of data, which is very computationally expensive. Also, the latent variables are not well disentangled. Future research directions include the extension to flat-hierarchical representations in the manner of [14] and the application to state-of-the art datasets.

References

[1] R. T. Chen, X. Li, R. B. Grosse and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders" NeurIPS 2018, Vol. 31.
[2] S. van Steenkiste, F. Locatello, J Schmidhuber, and O. Bachem, "Are disentangled representations helpful for abstract visual reasoning?", NeurIPS 2019, Vol. 32
[3] J. Lucas, G. Tucker, R. B. Grosse and M Norouzi, "Understanding posterior collapse in generative latent variable models" DGS@ICLR 2019
[4] N. Watters, L. Matthey, C. P. Burgess, and A. Lerchner, "Spatial broadcast decoder: a simple architecture for learning disentangled representations in VAEs". arXiv preprint 2019, arXiv:1901.07017
[5] R. Liu, J. Lehman, P. Molino, F. Petroski Such, E. Frank, A. Sergeev and J Yosinski, "An intriguing failing of convolutional neural networks and the CoordConv solution" NeurIPS 2018, Vol. 31
[6] H.Kim and A. Mnih, "Disentangling by factorising", ICML 2018, pp. 2649-2658
[7] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed and A. Lerchner, "Beta-VAE: learning basic visual concepts with a constrained variational framework" ICLR 2017, Poster
[8] C. K. Sønderby, T. Raiko, L. Maaløe and O. Winther, "Ladder variational autoencoders", NeuIPS 2016, Vol. 29.
[9] A. Vahdat, and J. Kautz, "Nvae: A deep hierarchical variational autoencoder", NeurIPS 2020, Vol. 33.
[10] L. Maaløe, M. Fraccaro, V. Liévin and O. Winther, "Biva: A very deep hierarchy of latent variables for generative modeling" NeuRIPS 2019, Vol. 32.
[11] F. JR. Ruiz, M.K. Titsias and D. M. Blei, "The generalized reparameterization gradient" NeurIPS 2016, Vol. 29.
[12] D. Kingma and P. Welling,"Auto-encoding variational bayes", ICLR 2014
[13] J. Soch and C. Allefeld, "Kullback-Leibler divergence for the Normal-Gamma distribution" arXiv preprint 2916, arXiv:1611.01437.
[14] S. Zhao, J. Song and S. Ermon, "Learning hierarchical features from generative models" arXiv preprint 2017, arXiv:1702.08396.