SELF-SUPERVISED VARIATIONAL AUTO-ENCODERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Density estimation, compression, and data generation are crucial tasks in artificial intelligence. Variational Auto-Encoders (VAEs) constitute a single framework to achieve these goals. Here, we present a novel class of generative models, called *self-supervised Variational Auto-Encoder* (selfVAE), that utilizes deterministic and discrete transformations of data. This class of models allows performing both conditional and unconditional sampling while simplifying the objective function. First, we use a single self-supervised transformation as a latent variable, where a transformation is either downscaling or edge detection. Next, we consider a hierarchical architecture, i.e., multiple transformations, and we show its benefits compared to the VAE. The flexibility of selfVAE in data reconstruction finds a particularly interesting use case in data compression tasks, where we can trade-off memory for better data quality, and vice-versa. We present the performance of our approach on three benchmark image data (Cifar10, Imagenette64, and CelebA).

1 INTRODUCTION

The framework of variational autoencoders (VAEs) provides a principled approach for learning latent-variable models. As it utilizes a meaningful low-dimensional latent space with density estimation capabilities, it forms an attractive solution for generative modelling tasks. However, its performance in terms of the test log-likelihood and quality of generated samples is often disappointing, thus, many modifications were proposed. In general, one can obtain a tighter lower bound, and, thus, a more powerful and flexible model, by advancing over the following three components: the *encoder* (Rezende et al., 2014; van den Berg et al., 2018; Hoogeboom et al., 2020; Maaløe et al., 2016), the *prior* (or *marginal* over latents) (Chen et al., 2016; Habibian et al., 2019; Lavda et al., 2020; Lin & Clark, 2020; Tomczak & Welling, 2017) and the *decoder* (Gulrajani et al., 2016). Recent studies have shown that by employing deep hierarchical architectures and by carefully designing building blocks of the neural networks, VAEs can successfully model high-dimensional data and reach state-of-the-art test likelihoods (Zhao et al., 2017; Maaløe et al., 2019; Vahdat & Kautz, 2020).

In this work, we present a novel class of VAEs, called *self-supervised Variational Auto-Encoders*, where we introduce additional variables to VAEs that result from discrete and deterministic transformations of observed images. Since the transformations are deterministic, and they provide a specific aspect of images (e.g., contextual information through detecting edges or downscaling), we refer to them as *self-supervised representations*. The introduction of the discrete and deterministic variables allows to train deep hierarchical models efficiently by decomposing the task of learning a highly complex distribution into training smaller and conditional distributions. In this way, the model allows to integrate the prior knowledge about the data, but still enables to synthesize unconditional samples. Furthermore, the discrete and deterministic variables could be used to conditionally reconstruct data, which could be of great use in data compression and super-resolution tasks.

We make the following contributions: i) We propose an extension of the VAE framework by incorporating self-supervised representations of the data. ii) We analyze the impact of modelling natural images with different data transformations as self-supervised representations. iii) This new type of generative model (*self-supervised Variational Auto-Encoders*), which is able to perform both conditional and unconditional sampling, demonstrate improved quantitative performance in terms of density estimation and generative capabilities on image benchmarks.

2 BACKGROUND

2.1 VARIATIONAL AUTO-ENCODERS

Let $\mathbf{x} \in \mathcal{X}^{\mathrm{D}}$ be a vector of observable variables, where $\mathcal{X} \subseteq \mathbb{R}$ or $\mathcal{X} \subseteq \mathbb{Z}$, and $\mathbf{z} \in \mathbb{R}^M$ denote a vector of latent variables. Since calculating $p_\vartheta(\mathbf{x}) = \int p_\vartheta(\mathbf{x}, \mathbf{z})\mathrm{d}\mathbf{z}$ is computationally intractable for non-linear stochastic dependencies, a variational family of distributions could be used for approximate inference. Then, the following objective function could be derived, namely, the *evidence lower bound* (ELBO) (Jordan et al., 1999):

$$\ln p_\vartheta(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\ln p_\theta(\mathbf{x}|\mathbf{z}) + \ln p_\lambda(\mathbf{z}) - \ln q_\phi(\mathbf{z}|\mathbf{x})\right], \tag{1}$$

where $q_\phi(\mathbf{z}|\mathbf{x})$ is the variational posterior (or the *encoder*), $p_\theta(\mathbf{x}|\mathbf{z})$ is the conditional likelihood function (or the *decoder*) and $p_\lambda(\mathbf{z})$ is the *prior* (or *marginal*), ϕ, θ and λ denote parameters.

The expectation is approximated by Monte Carlo sampling while exploiting the *reparameterization trick* in order to obtain unbiased gradient estimators. The models are parameterized by neural networks. This generative framework is known as *Variational Auto-Encoder* (VAE) (Kingma & Welling, 2013; Rezende et al., 2014).

2.2 VAEs WITH BIJECTIVE PRIORS

Even though the lower-bound suggests that the prior plays a crucial role in improving the variational bounds, usually a fixed distribution is used, e.g., a standard multivariate Gaussian. While being relatively simple and computationally cheap, the fixed prior is known to result in over-regularized models that tend to ignore most of the latent dimensions (Burda et al., 2015; Hoffman & Johnson, 2016; Tomczak & Welling, 2017). Moreover, even with powerful encoders, VAEs may still fail to match the variational posterior to a unit Gaussian prior (Rosca et al., 2018).

However, it is possible to obtain a rich, multi-modal prior distribution $p(\mathbf{z})$ by using a *bijective* (or *flow-based*) model (Dinh et al., 2016). Formally, given a latent code $\mathbf{z}$, a base distribution $p_V(\mathbf{v})$ over latent variables $\mathbf{v} \in \mathbb{R}^M$, and $f : \mathbb{R}^M \to \mathbb{R}^M$ consisting of a sequence of L diffeomorphic transformations[1], where $f_i(\mathbf{v}_{i-1}) = \mathbf{v}_i$, $\mathbf{v}_0 = \mathbf{v}$ and $\mathbf{v}_L = \mathbf{z}$, the *change of variable* can be used sequentially to express the distribution of $\mathbf{z}$ as a function of $\mathbf{v}$ as follows:

$$\log p(\mathbf{z}) = \log p_V(\mathbf{v}) - \sum_{i=1}^{L} \log \left| \frac{\partial f_i(\mathbf{v}_{i-1})}{\partial \mathbf{v}_{i-1}} \right|, \tag{2}$$

where $\left| \frac{\partial f_i(\mathbf{v}_{i-1})}{\partial \mathbf{v}_{i-1}} \right|$ is the Jacobian-determinant of the ith transformation.

Thus, using the bijective prior yields the following lower-bound:

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\Big[\log p_\theta(\mathbf{x}|\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x}) + \log p_V(\mathbf{v}_0) + \sum_{i=1}^{L} \log \left| \frac{\partial f_i^{-1}(\mathbf{v}_i)}{\partial \mathbf{v}_i} \right| \Big]. \tag{3}$$

In this work, we utilize RealNVP (Dinh et al., 2016) as the prior, however, any other flow-based model could be used (Kingma & Dhariwal, 2018; Hoogeboom et al., 2020). For the experiments and ablation study that shows the impact of the bijective prior on VAEs, we refer to the appendix A.1.

3 METHOD

3.1 MOTIVATION

The idea of self-supervised learning is about utilizing original unlabeled data to create additional context information. It could be achieved in multiple manners, e.g., by adding noise to data (Vincent et al., 2008) or masking data during training (Zhang et al., 2017). Self-supervised learning could also be seen as turning an unsupervised model into a supervised by, e.g., treating predicting next pixels as a classification task (Hénaff et al., 2019; Oord et al., 2018). These are only a few examples of a quickly growing research line (Liu et al., 2020).

[1] That is, invertible and differentiable transformations.

Here, we propose to use non-trainable transformations to obtain information about image data. Our main hypothesis is that since working with highly-quality images is challenging, we could alleviate this problem by additionally considering partial information about them. Fitting a model to images of lower quality, and then enhancing them to match the target distribution seems to be overall an easier task (Chang et al., 2004; Gatopoulos et al., 2020). By incorporating compressed transformations (i.e., the self-supervised representations) that still contain global information, with the premise that it would be easier to approximate, the process of modelling a high-dimensional complex density breaks down into simpler tasks. In this way, the expressivity of the model will grow and gradually result into richer, better generations. A positive effect of the proposed framework is that the model allows us to integrate prior knowledge through the image transformations, without losing its unconditional generative functionality. Overall, we end up with a two-level VAE with three latent variables, where one is a data transformation that can be obtained in a self-supervised fashion. In Figure 1 a schematic representation of the proposed approach with downscaling is presented.

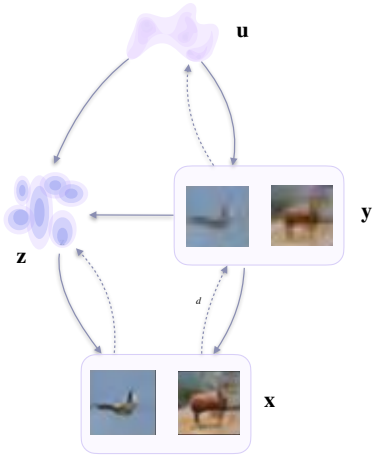


Figure 1: The proposed approach.

A number of exemplary image transformations are presented in Figure 2. We notice that with these transformations, even though they discard a lot of information, the global structure is preserved. As a result, in practice the model should have the ability to extract a general concept of the data, and add local information afterwards. In this work, we focus on downscaling (Figure 2.b, c & d) and edge detection or *sketching* (Fig. 2.i).

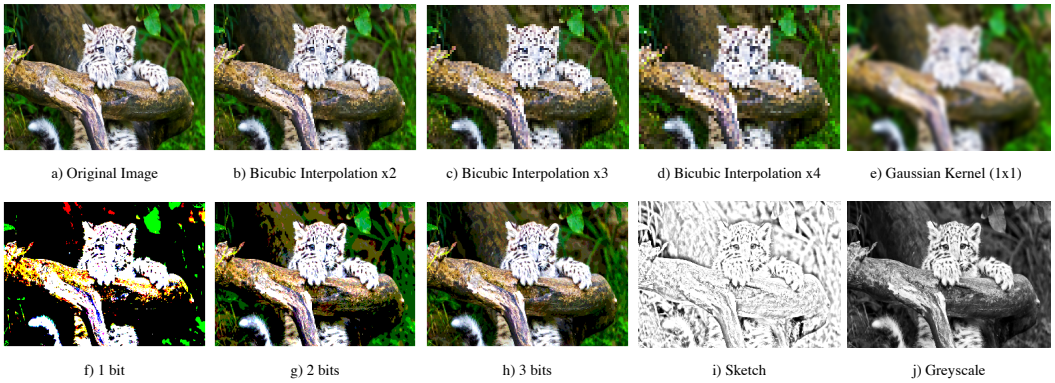


Figure 2: Image Transformations. All of these transformations still preserve the global structure of the samples but they disregard the high resolution details in different ways.

3.2 Model Formulation

In our model, we consider representations that result from *deterministic* and *discrete* transformations of an image. Formally, we introduce a transformation $d : \mathcal{X}^D \rightarrow \mathcal{X}^C$ that takes $\mathbf{x}$ and returns an image representation $\mathbf{y}$, e.g., a downscaled image. Since we lose information about the original image, $\mathbf{z}$ could be seen as a variable that compensates lost details in $\mathbf{x}$. Further we propose to introduce an additional latent variable, $\mathbf{u} \in \mathbb{R}^N$ to model $\mathbf{y}$ and $\mathbf{z}$. We can define the joint distribution of $\mathbf{x}$ and $\mathbf{y}$ as follows: $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$, where $p(\mathbf{y}|\mathbf{x}) = \delta(\mathbf{y} - d(\mathbf{x}))$ due to the deterministic transformation $d(\cdot)$, where $\delta(\cdot)$ is the Kronecker delta. Thus, the empirical distribution is $\delta(\mathbf{y} - d(\mathbf{x}))p_{data}(\mathbf{x})$. However, since we are interested in decomposing the problem of modeling a complex distribution $p(\mathbf{x})$, we propose to model $p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ instead, and utilize the variational inference of the form $Q(\mathbf{u}, \mathbf{z}|\mathbf{x}, \mathbf{y}) = q(\mathbf{u}|\mathbf{y})q(\mathbf{z}|\mathbf{x})$ that yields:

$$\ln p(\mathbf{x}, \mathbf{y}) \geq \mathbb{E}_Q \big[\ln p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z}) + \ln p(\mathbf{z}|\mathbf{u}, \mathbf{y}) + \ln p(\mathbf{y}|\mathbf{u}) + \ln p(\mathbf{u}) - \ln q(\mathbf{z}|\mathbf{x}) - \ln q(\mathbf{u}|\mathbf{y}) \big]. \quad (4)$$

Intuitively, the premise for selfVAE is that the latents $\mathbf{u}$ will capture the global structure of the input data and the latents $\mathbf{z}$ will encode the missing information between $\mathbf{y}$ and $\mathbf{x}$, guiding the model to discover the distribution of the target observations. In order to highlight the self-supervised part in our model, we refer to it as the *self-supervised Variational Auto-Encoder* (or selfVAE for short).

Further, we propose to choose the following distributions:

$$p(\mathbf{v}) = \mathcal{N}\left(\mathbf{v}|\mathbf{0}, \mathbf{1}\right) \qquad q_{\phi_1}\left(\mathbf{u}|\mathbf{y}\right) = \mathcal{N}\left(\mathbf{u}|\boldsymbol{\mu}_{\phi_1}(\mathbf{y}), \mathrm{diag}\left(\boldsymbol{\sigma}_{\phi_1}(\mathbf{y})\right)\right)$$

$$p_\lambda\left(\mathbf{u}\right) = p(\mathbf{v}) \prod_{i=1}^{F} \left| \det \frac{\partial f_i(\mathbf{v}_{i-1})}{\partial \mathbf{v}_{i-1}} \right|^{-1} \qquad q_{\phi_2}\left(\mathbf{z}|\mathbf{x}\right) = \mathcal{N}\left(\mathbf{z}|\boldsymbol{\mu}_{\phi_2}(\mathbf{x}), \mathrm{diag}\left(\boldsymbol{\sigma}_{\phi_2}(\mathbf{x})\right)\right).$$

$$p_{\theta_1}\left(\mathbf{y}|\mathbf{u}\right) = \sum_{i=1}^{I} \pi_i^{(\mathbf{u})} \mathrm{Dlogistic}\left(\mu_i^{(\mathbf{u})}, s_i^{(\mathbf{u})}\right)$$

$$p_{\theta_2}\left(\mathbf{z}|\mathbf{y}, \mathbf{u}\right) = \mathcal{N}\left(\mathbf{z}|\boldsymbol{\mu}_{\theta_2}(\mathbf{y}, \mathbf{u}), \mathrm{diag}\left(\boldsymbol{\sigma}_{\theta_2}(\mathbf{y}, \mathbf{u})\right)\right)$$

$$p_{\theta_3}\left(\mathbf{x}|\mathbf{z}, \mathbf{y}\right) = \sum_{i=1}^{I} \pi_i^{(\mathbf{z},\mathbf{y})} \mathrm{Dlogistic}\left(\mu_i^{(\mathbf{z},\mathbf{y})}, s_i^{(\mathbf{z},\mathbf{y})}\right)$$

where $\mathrm{Dlogistic}$ is defined as the discretized logistic distribution (Salimans et al., 2017), and we utilize a flow-based model for $p_\lambda\left(\mathbf{u}\right)$. Notice that we use the discretized logistic distribution, because images are represented by values between 0 and 255. For integer-valued random variables, other distributions like Gaussian are inappropriate.

3.3 Generation and Reconstruction in selfVAE

As generative models, VAEs can be used to synthesize novel content through the following process: $\mathbf{z} \sim p(\mathbf{z}) \rightarrow \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$, but also to reconstruct a data sample $\mathbf{x}^*$ by using the following scheme: $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}^*) \rightarrow \mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$.

Interestingly, our approach allows to utilize more operations regarding data generation and reconstruction. First, analogously to VAEs, the selfVAE allows to generate data by applying the following hierarchical sampling process (*generation*):

$$\mathbf{u} \sim p(\mathbf{u}) \rightarrow \mathbf{y} \sim p(\mathbf{y}|\mathbf{u}) \rightarrow \mathbf{z} \sim p(\mathbf{z}|\mathbf{u}, \mathbf{y}) \rightarrow \mathbf{x} \sim p(\mathbf{x}|\mathbf{y}, \mathbf{z}).$$

However, we can use the ground-truth $\mathbf{y}$ (i.e, $\mathbf{y}^* = d(\mathbf{x}^*)$), and sample or infer $\mathbf{z}$. Then, the generative process for the former (*conditional generation*) is:

$$\mathbf{z} \sim q(\mathbf{z}|\mathbf{x}^*) \rightarrow \mathbf{x} \sim p(\mathbf{x}|\mathbf{y}^*, \mathbf{z}),$$

and for the latter (*conditional reconstruction*):

$$\mathbf{u} \sim q(\mathbf{u}|\mathbf{y}^*) \rightarrow \mathbf{z} \sim p(\mathbf{z}|\mathbf{u}, \mathbf{y}^*), \rightarrow \mathbf{x} \sim p(\mathbf{x}|\mathbf{y}^*, \mathbf{z}).$$

If $\mathbf{y}$ is a downscaling transformation of the input image, selfVAE can be used in a manner similar to the super-resolution (Gatopoulos et al., 2020). Alternatively, we can sample (or generate) $\mathbf{y}$ instead, and choose to sample or infer $\mathbf{z}$. In this way, we can reconstruct an image in two ways, namely, *reconstruction 1*:

$$\mathbf{y}^* = d(\mathbf{x}^*) \rightarrow \mathbf{u} \sim q(\mathbf{u}|\mathbf{y}^*) \rightarrow \mathbf{y} \sim p(\mathbf{y}|\mathbf{u}) \rightarrow \mathbf{z} \sim p(\mathbf{z}|\mathbf{u}, \mathbf{y}) \rightarrow \mathbf{x} \sim p(\mathbf{x}|\mathbf{z}, \mathbf{y}),$$

and *reconstruction 2*:

$$\Big(\mathbf{y}^* = d(\mathbf{x}^*) \rightarrow \mathbf{u} \sim q(\mathbf{u}|\mathbf{y}^*) \rightarrow \mathbf{y} \sim p(\mathbf{y}|\mathbf{u})\Big), \text{ then } \mathbf{z} \sim q(\mathbf{z}|\mathbf{x}^*) \rightarrow \mathbf{x} \sim p(\mathbf{x}|\mathbf{y}, \mathbf{z}).$$

The presented versions of generating and reconstructing images could be useful in the compression task. As we will see in the experiments, each option creates a different ratio of the reconstruction quality against the memory that we need to allocate to send information. However, every inferred variable needs to be sent, thus, more sampling corresponds to lower memory requirements.

3.4 HIERARCHICAL SELF-SUPERVISED VAE

The proposed approach can be further extended and generalized by introducing multiple transformations, in the way that it is illustrated in Figure 3. By incorporating a single (or multiple) self-supervised representation(s) of the data, the process of modelling a high-dimensional complex density breaks down into K simpler modeling tasks. Thus, we obtain a K-level VAE architecture, where the overall expressivity of the model grows even further and gradually results into generations of higher quality. Some transformations cannot be applied multiple times (e.g., edge detection), however, others could be used sequentially, e.g., downscaling.

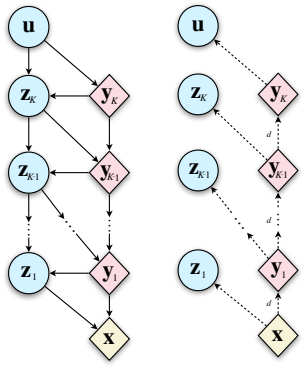


Figure 3: Hierarchical selfVAE.

We take K self-supervised data transformations $d_k(\cdot)$ that give K representations denoted by $\mathbf{y}_{1:K} = [\mathbf{y}_1, \ldots, \mathbf{y}_K]$, and the following variational distributions:

$$Q(\mathbf{u}, \mathbf{z}|\mathbf{x}, \mathbf{y}_{1:K}) = q(\mathbf{u}|\mathbf{y}_K) q(\mathbf{z}_1|\mathbf{x}) \prod_{k=1}^{K-1} q(\mathbf{z}_{k+1}|\mathbf{y}_k), \tag{5}$$

that yields the following objective:

$$\ln p(\mathbf{x}, \mathbf{y}_{1:K}) \geq \mathbb{E}_Q \big[\ln p_\theta(\mathbf{x}|\mathbf{y}_1, \mathbf{z}_1) + \sum_{k=1}^{K-1} \big(\ln p(\mathbf{z}_k|\mathbf{y}_k, \mathbf{z}_{k+1}) + \ln p(\mathbf{y}_k|\mathbf{y}_{k+1}, \mathbf{z}_{k+1}) \big) +$$
$$+ \ln p(\mathbf{z}_K|\mathbf{u}, \mathbf{y}_K) + \ln p(\mathbf{y}_K|\mathbf{u}) + \ln p(\mathbf{u}) - \ln q(\mathbf{u}|\mathbf{y}_K) - \ln q(\mathbf{z}_1|\mathbf{x}) - \sum_{k=1}^{K-1} \ln q(\mathbf{z}_{k+1}|\mathbf{y}_k) \big]. \tag{6}$$

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets We evaluate the proposed model on CIFAR-10, Imagenette64 and CelebA:

CIFAR-10 The CIFAR-10 dataset is a well-known image benchmark data containing 60.000 training examples and 10.000 validation examples. From the training data, we put aside 15% randomly selected images as the test set. We augment the training data by using random horizontal flips and random affine transformations and normalize the data uniformly in the range (0, 1).

Imagenette64 Imagenette64[2] is a subset of 10 classes from the downscaled Imagenet dataset. We downscaled the dataset to 64px $\times$ 64px images. Similarly to CIFAR-10, we put aside 15% randomly selected training images as the test set. We used the same data augmentation as in CIFAR-10

CelebA The Large-scale CelebFaces Attributes (CelebA) Dataset consists of 202.599 images of celebrities. We cropped original images on the 40 vertical and 15 horizontal component of the top left corner of the crop box, which height and width were cropped to 148. Besides the uniform normalization of the image, no other augmentation was applied.

Architectures Encoders and decoders consist of building blocks composed of DenseNets (Huang et al., 2016), channel-wise attention (Zhang et al., 2018), and ELUs (Clevert et al., 2015) as activation functions. The dimensionality of all the latent variables were kept at $8 \times 8 \times 16 = 1024$ and all models were trained using AdaMax (Kingma & Ba, 2014) with data-dependent initialization (Salimans & Kingma, 2016). Regarding the selfVAEs, in CIFAR-10 we used an architecture with a single downscaled transformation (selfVAE-downscale), while on the remaining two datasets (CelebA

[2]https://github.com/fastai/imagenette

and Imagenette64) we used a hierarchical 3-leveled selfVAE with downscaling, and a selfVAE with sketching. All models were employed with the bijective prior (RealNVP) comparable in terms of the number of parameters (the range of the weights of all models was from 32M to 42M). For more details, please refer to the appendix section A.2.

Evaluation We approximate the negative log-likelihood using 512 IW-samples (Burda et al., 2015) and express the scores in bits per dimension (*bpd*). Additionally, for CIFAR-10, we use the *Fréchet Inception Distance* (FID) (Heusel et al., 2017).

Table 1: Quantitative comparison on test sets from CIFAR-10, CelebA, and Imagenette64. *Measured on training set.

Dataset	**Model**	*bpd* ↓	FID ↓
CIFAR-10	PixelCNN (van den Oord et al., 2016)	3.14	65.93
	GLOW (Kingma & Dhariwal, 2018)	3.35	65.93
	ResidualFlow (Chen et al., 2019)	3.28	46.37
	BIVA (Maaløe et al., 2019)	3.08	-
	NVAE (Vahdat & Kautz, 2020)	**2.91**	-
	DDPM (Ho et al., 2020)	3.75	**5.24** (**3.17***)
	VAE (ours)	3.51	41.36 (37.25*)
	selfVAE-downscale	3.65	34.71 (29.95*)
CelebA	RealNVP (Dinh et al., 2016)	3.02	-
	VAE (ours)	3.12	-
	selfVAE-sketch	3.24	-
	selfVAE-downscale-3lvl	**2.88**	-
Imagenette64	VAE (ours)	3.85	-
	selfVAE-downscale-3lvl	**3.70**	-

4.2 Quantitative results

We present the results of the experiments on the benchmark datasets in Table 1. First, we notice that on CIFAR-10 our implementation of the VAE is still lacking behind other generative models in terms of *bpd*, however, it is better or comparable in terms of FID. The selfVAE-downscale achieves worse *bpd* than the VAE. A possible explanation may lie in the small image size (32×32), as the benefits of breaking down the learning process in two or more steps are not obvious given the small target dimensional space. Nevertheless, the selfVAE-downscale achieves significantly better FID scores than the other generative models. This result could follow from the fact that downscaling allows to maintain context information about the original image and, as a result, a general coherence is of higher quality.

Interestingly, on the two other datasets, a three-level selfVAE-downscale achieves significantly better *bpd* scores than the VAE with the bijective prior. This indicates the benefit of employing a multi-leveled self-supervised framework against the VAE in higher-dimensional data, where the plain model fails to scale efficiently. It seems that the hierarchical structure of self-supervised random variables allows to encode the missing information more efficiently in $\mathbf{z}_k$, in contrast to the vanilla VAE, where all information about images must be coded in $\mathbf{z}$. This result is promising and indicates that the proposed approach is of great potential for generative modelling.

4.3 Qualitative results

We present generations on CIFAR-10 and Imagenette64 in Figure 4 and on CelebA in Figure 5, and reconstructions on CIFAR-10 and CelebA in Figure 6.

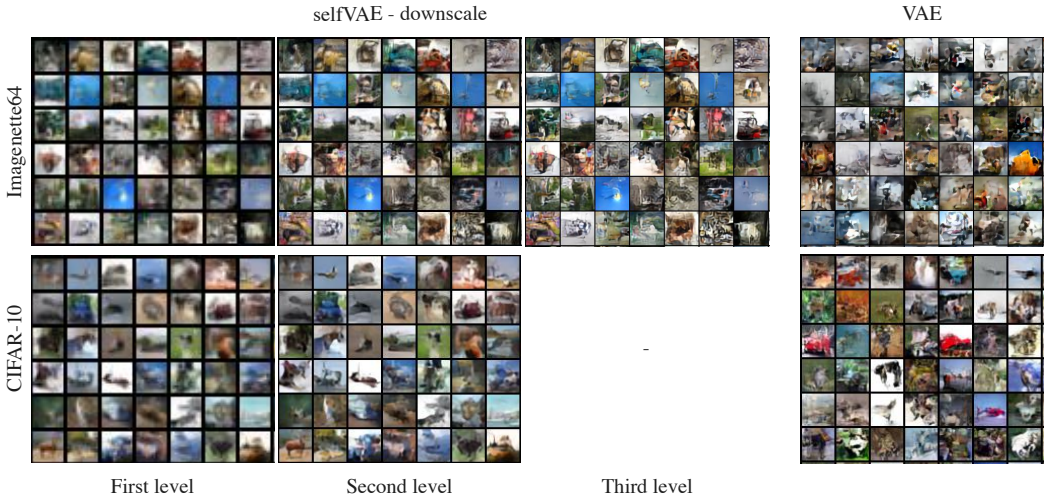


Figure 4: Uncoditional generations on Imagenette64 and CIFAR-10.

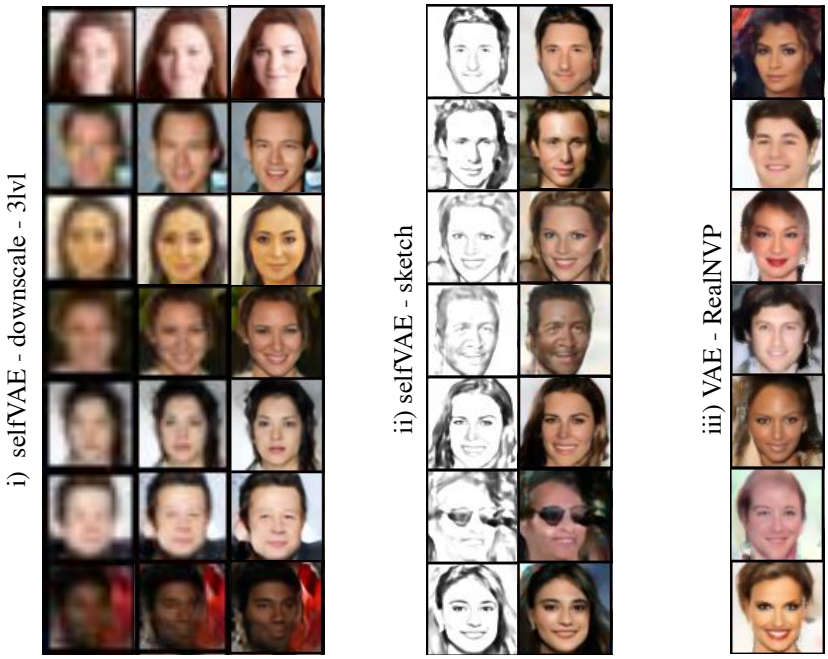


Figure 5: Unconditional CelebA generations from (i) the three-level self-supervised VAE with downscaling, (ii) the self-supervised VAE employed with edge detection (sketches), (iii) the VAE with RealNVP prior.

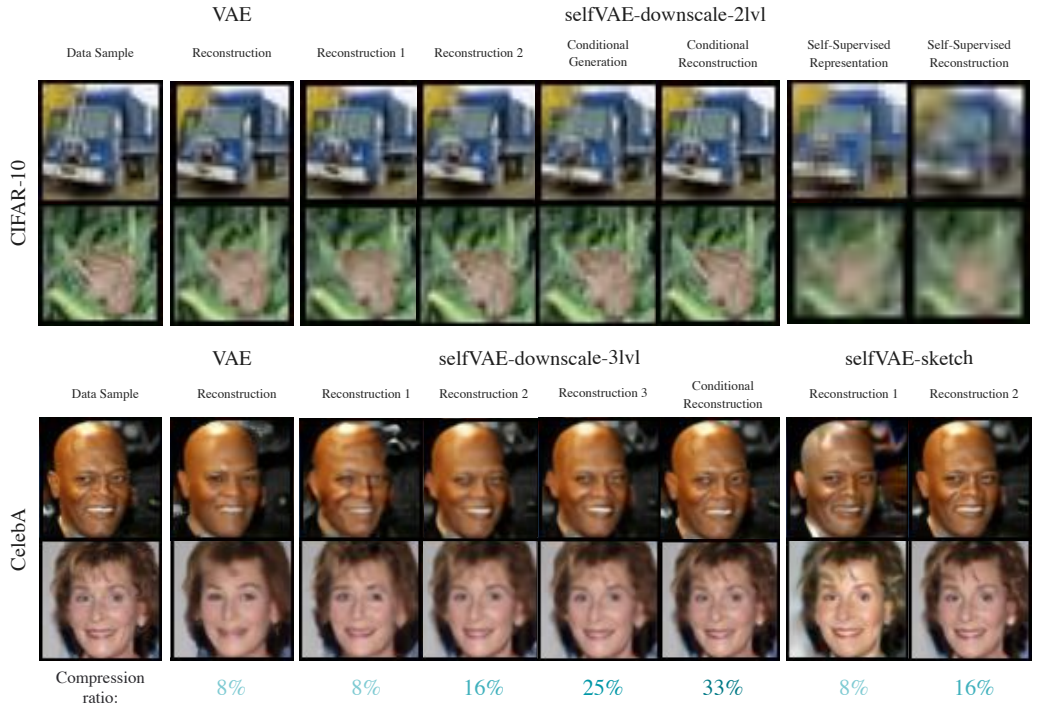


Figure 6: Comparison on image reconstructions with different amount of sent information.

We first notice that the generations from selfVAE seem to be more coherent, in contrast with these from VAE that produces overall more contextless and distorted generations. This result seems to be in line with the FID scores. Especially for CelebA, we observe impressive synthesis quality, great sampling diversity and coherent generations (Figure 5). On the Imagenette64 dataset, we can also observe crisper generations for our method compared to the VAE (Figure 4).

Furthermore, the hierarchical selfVAE seems to be of a great potential for compression purposes. In contrast to the VAE, which is restricted to using a single way of reconstructing an image, the selfVAE allows four various options with different quality/memory ratios (Figure 6). In the selfVAE-sketch, we can retrieve the image with high accuracy by using only 16% of the original data, as it manages to encode all the texture of the image to $\mathbf{z}$ (Figure 11). This shows the advantage of choosing prior knowledge into the learning process. Lastly, the latents learn to add extra information, which defines the end result, and we can alter details of an image like facial expressions (Figure 12.ii).

5 Conclusion

In this paper, we showed that taking deterministic and discrete transformations results in coherent generations of high visual quality, and allows to integrate prior knowledge without loosing its unconditional generative functionality. The experimental results seem to confirm that hierarchical architectures perform better and allow to obtain both better *bpd* scores and better generations and reconstructions. In the experiments, we considered two classes of image transformations, namely, *downscaling* and edge detection (*sketching*). However, there is a vast of possible other transformations (see Figure 2), and we leave investigating them for future work. Moreover, we find the proposed approach interesting for the compression task. A similar approach with a multi-scale auto-encoder for image compression was proposed, e.g, by Mentzer et al. (2019) or Razavi et al. (2019). However, we still use a probabilistic framework and indicate that various non-trainable image transformations (not only multiple scales) could be of great potential.

Acknowledgments

Anonymized for the double-blind review.

REFERENCES

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv*, 2015.

Hong Chang, Dit-Yan Yeung, and Yimin Xiong. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pp. I–I. IEEE, 2004.

Ricky T. Q. Chen, Jens Behrmann, David Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *arXiv*, 2019.

Xi Chen, Diederik P. Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv*, 2016.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv*, 2015.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv*, 2016.

Eduardo S. L. Gastal and Manuel M. Oliveira. Domain transform for edge-aware image and video processing. *ACM TOG*, 30(4):69:1–69:12, 2011. Proceedings of SIGGRAPH 2011.

Ioannis Gatopoulos, Maarten Stol, and Jakub M. Tomczak. Super-resolution variational auto-encoders. *arXiv.2006.05218*, 2020.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

Ishaan Gulrajani, Kundan Kumar, Faruk Ahmed, Adrien Ali Taiga, Francesco Visin, David Vazquez, and Aaron Courville. Pixelvae: A latent variable model for natural images. *arXiv*, 2016.

Amirhossein Habibian, Ties van Rozendaal, Jakub M Tomczak, and Taco S Cohen. Video compression with rate-distortion autoencoders. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7033–7042, 2019.

Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33, 2020.

Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, pp. 2, 2016.

Emiel Hoogeboom, Victor Garcia Satorras, Jakub M Tomczak, and Max Welling. The convolution exponential and generalized sylvester flows. *arXiv preprint arXiv:2006.01910*, 2020.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. *arXiv*, 2016.

Michael I. Jordan, Zoubin Ghahramani, and et al. An introduction to variational methods for graphical models. In *Machine Learning*, pp. 183–233. MIT Press, 1999.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv*, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv*, 2013.

Frantzeska Lavda, Magda Gregorová, and Alexandros Kalousis. Data-dependent conditional priors for unsupervised learning of multimodal data. *Entropy*, 22(8):888, 2020.

Shuyu Lin and Ronald Clark. Ladder: Latent data distribution modelling with a generative prior. *arXiv preprint arXiv:2009.00088*, 2020.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *arXiv preprint arXiv:2006.08218*, 2020.

Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv*, 2016.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. *arXiv*, 2019.

Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Practical full resolution learned lossless image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10629–10638, 2019.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pp. 14866–14876, 2019.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv*, 2014.

Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. Distribution matching in variational inference. *arXiv*, 2018.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *arXiv*, 2016.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv*, 2017.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *arXiv*, 2016.

Jakub M Tomczak and Max Welling. Vae with a vampprior. *arXiv preprint arXiv:1705.07120*, 2017.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv*, 2020.

Rianne van den Berg, Leonard Hasenclever, Jakub M. Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. *arXiv*, 2018.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv*, 2016.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1058–1067, 2017.

Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. *arXiv*, 2018.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from deep generative models. In *International Conference on Machine Learning*, pp. 4091–4099, 2017.

A APPENDIX

A.1 THE IMPACT OF THE BIJECTIVE PRIOR ON VAES

A.1.1 MOTIVATION

Bijective neural networks were first introduced to increase the expressibility of the encoder, due to its forced limitations to follow a mean-field variational family of Gaussian distributions. As they demand the base function to be known and simple (i.e. a distribution that is easy to estimate and sample from), the mapping of the input space to the latent follows a two step process; first the input sample is mapped to the base distribution though the encoder, and then transformed to match the unimodal Gaussian prior. However, despite the richer variational posterior, Rosca et al. (2018) proved that the optimal prior may still not much a fixed unit Gaussian distribution, which in addition, is known to result in over-regularized models that tend to ignore more of the latent codes (Burda et al., 2015; Tomczak & Welling, 2017).

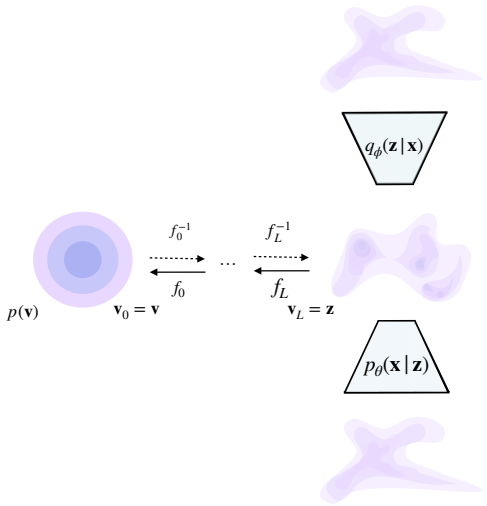
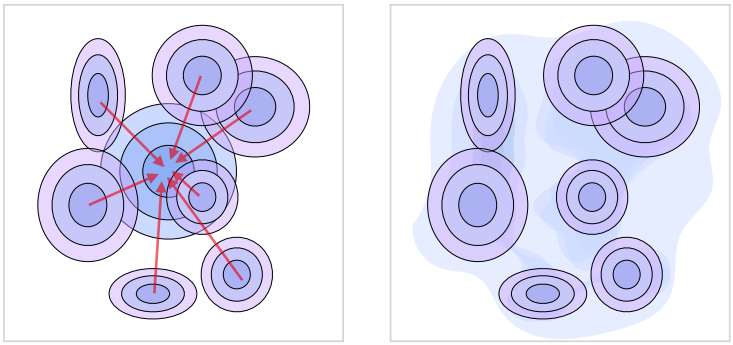


Figure 7: VAE employed with bijective prior.

However, little attention has been paid to adapt bijective neural networks as a learnable, data-driven prior of Variational Auto-Encoders. Since the prior of the latent dimensions plays a crucial part in the objective function, it will contribute into a better likelihood estimation of the data, while providing more informative latent codes. There are a couple of advantages using a bijective network to estimate the prior distribution:

- **Arbitrary complex, data-driven prior** Instead of forcing the encoder to match a fixed distribution, we follow the intuitive process, that now, the prior distribution adapts to the posterior during the training progress. Thus, the posterior is not being dragged to the centre of a standard Gaussian distribution and it is allowed to move freely in the latent space. As the normalizing flows framework allows the transformation of a simple function to an arbitrary complex one, the result would be a richer, multi-modal distribution that incorporates causal knowledge of the data at hand.

- **Simple implementation and integration** Instead of breaking the encoder into a two step process, we fit a flow-based generative model to the latent codes produced by the encoder. Additionally, getting advantage of the invertible nature of the network, we can generate a datapoint by simply sampling a point from its base distribution, and transforming it (through the inverse process) into a latent code of the variational posterior.



a) Standard Gaussian Prior

b) Learnable Multivariate Prior

Figure 8: Left: The standard prior is too strong and over-regularizes the encoder and create "holes" in the latent space. Right: Learnable prior adjusts to the posterior distribution, surrounding the variable posterior with smooth transitions.

- **Fast sampling and inference**. An alternative way of retrieving a powerful latent prior is through autoregressive modelling of the latent codes. Formally, $p(\mathbf{z})$ is factorized as

$$p(\mathbf{z}) = \prod_{i=1}^{d} p\left(z_i | \mathbf{z}_{<i}\right)$$

  so that estimating $p(\mathbf{z})$ reduces to the estimation of each single conditional probability density (CPD) expressed as $p(z|\mathbf{z}_{<i})$, where the symbol $<$ implies an order over random variables. However, because high-dimensional data often requires a relatively high-dimensional latent space, in order to extract the highly informative causal components, the sampling process can be an important barrier for fast generation of novel content. On the contrary, the bijective neural networks require only a forward pass of a sampled point, reducing significantly the inference time.

It is straightforward to obtain a rich, multi-modal prior, and the process its described on section 2.2.

A.1.2 Experiments

On this section, we will evaluate the performance of Variational Auto-Encoder with using different latent variable prior distributions, both on a quantitatively and qualitatively view. The models named as "VAE" represents the plain architecture with standard normal prior, while "VAE + MoG" and "VAE + RealNVP" represent data-driven priors, where the former employs 10 mixture of Gaussians and the latter the bijective network RealNVP. For fair comparison, all the models share the same neural networks architectures both on the encoder and decoder. For the specific architecture of the neural networks, please refer to the section .

Quantitative Analysis Table 2 represents the quantitative results of density estimation on the natural image dataset CIFAR-10. We measure performance by estimating the log-likelihood with 512 importance weighted samples on the test set. To begin with, the use of a mixture of Gaussians, a learned distribution, performs better that the VAE with fix prior, with a negligible cost on additional trainable parameters, which do not allocate extra time in sample generation. However, the adaption of RealNVP as a bijective network, improves significantly the likelihood score. Interestingly, both models with learned priors perform better than the fixed one, by reaching better reconstruction loss with the cost of allocating more regularization cost. This behavior exactly reflects the relationship between variational posterior and the prior; when a fix unimodal Gaussian is used, it pulls the variational ones to its pick. As a result, all the variational posteriors are close to the mean of the prior, resulting into a smaller KL diverge but blurry reconstructions. On the contrary, the learned multi-modal priors allow the posteriors to move more freely and are able to adjust to the variational family (see figure 8). This distribution mismatch is the greatest when the RealNVP prior is used, indicating that the prior allows the encoder to form more and more complex distributions, by trading of regularization loss for better reconstructions.

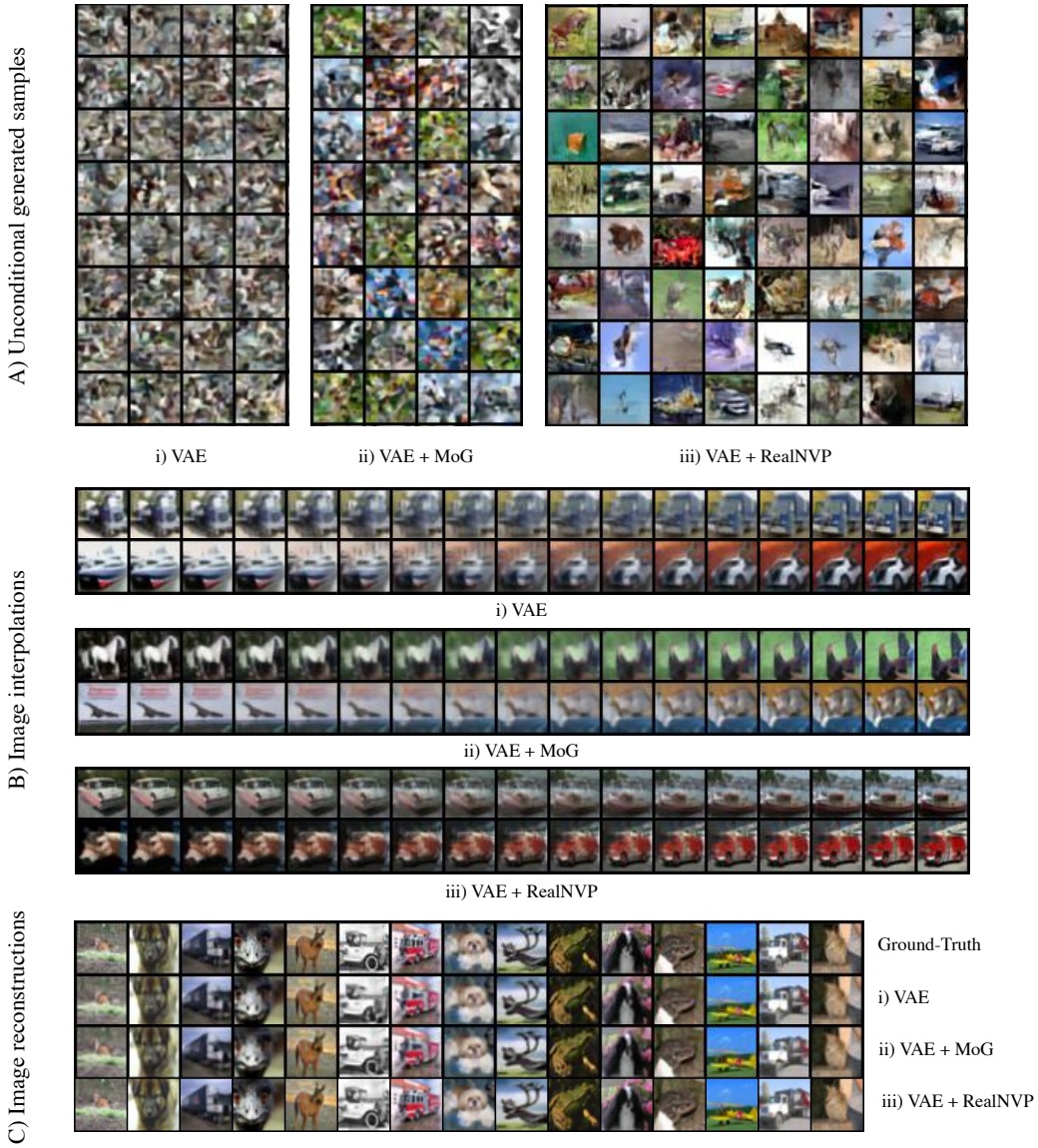


Figure 9: Qualitatively results on CIFAR-10 for VAEs with various latent prior distributions. The depicted results show A) unconditional generative samples B) image interpolations and C) image reconstructions.

Qualitative Analysis The difference in performance becomes more obvious by looking at the qualitatively results, presented in Figure 9. Looking at the generative samples, we see a clear improvement when moving to richer prior. The plain VAE with standard normal distribution generates random patters while the model with a mixture of Gaussian prior enhance them with colors. However, only VAE with RealNVP showcases generations that manage to display a global context. To examine the richness of the learnt space, we also perform interpolation on the latent codes between two ground-truth images. Ideally, the internal generations should result into meaningful modifications of the provided samples. In the case of VAE, we see that the inner generations produce blurry samples, indicating the unexpressivety of the latent codes. The generations are getting better with the MoG prior, but the most impressive results are coming from the framework with RealNVP. We can see that the samples are more influenced by their closest ground-truth image, while adapting more and more core characteristics when moving closer to the other. For example, on the second row of B:iii, the original brown horse first turns into red, adapting the color of the firetruck, before it smoothly result into it. Lastly, on image reconstructions we again witness the same patent, when the richer

Table 2: Generative modelling performance on CIFAR-10 obtained from different priors in bits per dimension. The generation time is based on 100 runs on a singe GeForce GTX 1080 Ti GPU for 1 image (and 100 images).

Model	*nll* *(bits/dim)*	*reconstruction loss* RE_x	*regularization loss* KL_z	#params	*generation time* (sec)
VAE	3.88	*6675*	*1596*	20M	0.01 (0.07)
VAE + MoG	3.83	*6512*	*1653*	22M	0.01 (0.07)
VAE + RealNVP	**3.51**	*5540*	*1966*	32M	0.07 (0.14)

prior does result into better reconstructions, confirming the quantitatively results presented on table 2. Specifically, the VAE with the bijective prior showcases an excellent performance on the natural image reconstruction task, which is contrary to the performance of the other two approaches. This is associated with the effectiveness of the powerful, invertible, data-driven prior (in our case, the RealNVP) and its ability to boost the performance significantly with negligible sacrifice on generation speed, and none on inference.

A.2 Neural Network Architecture

The choice of the NN architecture is crucial for the performance and the scalability of the overall framework, and usually architectures that showcased great performance in discriminate tasks (i.e. classification) are used in generative modelling tasks as well. However, the internal representations that the networks have to discover are fundamentally different, and little attention has been given into designing a NN specifically for an auto-encoder setting. For example, in classification tasks the network extracts specific representation of a particular object, in contrast with the generative models, where we aim for discovering the semantic structure of the data. Thus, as we argue that we can benefit from a carefully designed architecture, in this section we present our approach.

For building blocks of the network, we employed densely connected convolutional networks instead of residual ones. The motivation for this choice is that since DenseNets encourage feature reuse, it will help preserve visual information from the very first layer effectively, while requiring less trainable parameters. Thus, the network could discover easier generic graphical features and local pixel correlations. The concatenation of the filters will also alleviate the vanishing-gradient problem and allow us to build deep architectures. Additionally, exponential linear units (ELUs) are used everywhere as activation functions. In contrast to ReLUs, ELUs have negative values which allows them to push mean unit activations closer to zero, which speeds up the learning process. This is due to a reduced *bias shift effect*; bias that is introduced to the units from those of the previous layer which have a non-zero mean activation.

Typically, every convolution operation precedes a batch normalization layer, as they empirically exhibit a boost in performance in discriminate tasks. However, their performance is known to degrade for small batch sizes, as the variance of the activation noise that they contribute is inversely proportional to the number of data that is processed. This noise injection, in combination with their intensive memory demands, can be critical drawbacks when we process image data, especially high-dimensional ones. We instead use weight normalization, where even though it separates the weight vector from its direction just like batch normalization, they do not make use of the variance. This allows them to get the desired output even in small mini-batches, while allocating a small proportion of memory. We empirically find out that indeed using weight normalization reduces the overfitting of the model. In addition, we used a data-dependant initialization of the model parameters, by sampling the first batch of the training set. This will allow the parameters to be adjusted by the output of the previous layers, taking into account and thus resulting into a faster learning process.

An important element of the auto-encoding scheme is the process of feature downscaling and upscaling. Despite its success in classification tasks, pooling is a fixed operation that replacing it with a stride-convolution layer can also be seen as generalization, as the scaling process is now learned. This will increase the models' expressibility with the cost of adding a negligible amount of learning parameters. For the upscaling operation, even though various methods have been proposed (Shi et al., 2016), we found out that the plain transposed convolution generalised better than the others, while requiring far less trainable parameters. Finally, inspired from the recent advantages on

super-resolution neural network architectures, we used *channel-wise attention* blocks (CA) at the end of every DenseNet block (Zhang et al., 2018). The CA blocks will help the network to focus on more informative features, by exploiting the inter-dependencies among feature channels. Thus, it performs feature recalibration in a global way, where the per-channel summary statistics are calculated and then used to selectively emphasise informative feature-maps as well as suppress useless ones (e.g. redundant feature-maps). This is done through a global average pooling, that squeezes global spatial information into a channel statistical descriptor, followed by a gating mechanism, where it learns nonlinear interactions between the input channels.

The core building blocks and the network of an auto-encoding network are illustrated in Figure 10.

A.3 Self-Supervised VAE - Sketch Reconstructions

Given the astonishing performance on visual tasks, CNNs are commonly thought to recognise objects by learning increasingly complex representations of object shapes. However, Geirhos et al. (2019) showed that where humans see shapes, CNNs are strongly biased towards recognising textures. Furthermore, architectures that learn shape-based representations come with several unexpected emergent benefits such as previously unseen robustness towards a wide range of image distortions. This acted as a motivation to employ the framework of self-supervised auto-encoder with a representation that captures the shape of the object. Thus, the first part will model the outlines of the given object while the second will be responsible for its texture.

We can retrieve a shape-based representation of an image by detecting its edges. Edges appear when there is a sharp change in brightness and it usually corresponds to the boundaries of an object. There are many different techniques for computing the edges, like using filters that extract the gradient of the image (Sobel kernels, Laplacian of Gaussian, etc.). In this experiment, we will use the method proposed in Gastal & Oliveira (2011), as it is a fast, high-quality edge-preserving filtering of images. Specifically, in order to obtain a pencil *sketch* (that is, a black-and-white drawing) of the input image, we will make use of two image-blending techniques, known as *dodging* and *burning*. The former lightens an image, whereas the latter darkens it. A sketch transformation can be obtained by using dodge to blend the grayscale image with its blurred inverse. In this way, we produce high-quality edge-preserving filtering of the image efficiently performed in linear time.

Qualitative Analysis Similar to the case when we used a downscaled transformation of the input image as conditional representation, the selfVAE framework here also allows for different ways to reconstruct an image. The qualitative results when we employ a sketch representation are visible in Figure 11. To start with, we see that even though because of the sketch representation we lose the texture of the image, we preserve not only the global information but also high-level details that characterize each person. In this way, we let the generative model emphasise on these specifics at its first step, which through the latent variable $\mathbf{u}$, manages to reconstruct with tremendous accuracy. This can be confirmed by visually comparing the images of the original compressed images (CM) with those of the reconstructed ones. These great results hold also for the conditional reconstruction, where the original sample (OG) is synthesised from conditioning on the original sketch representation (CM). Furthermore, we gain further interpretation of the model through the reconstructions that use only the latent variable $\mathbf{u}$ (RS1) and both latent variables (RS2). Given that both methods infer the sketch image, the end results still preserve all detail concepts of the human portrait. However, they are different in terms of the texture (colour) of the image, which is modelled through $\mathbf{z}$. One the one hand, in the first case its value is inferred through the sketch, and thus it produces the most probable values. That is why we see on ground-truth images that incorporate unusual lighting, it outputs a more natural outcome. On the other hand, in the second case, the latent codes of $\mathbf{z}$ are computed given the original sample. And from the results, we can see that, indeed, all the information about the texture of the image can be compressed into the latent $\mathbf{z}$, as these reconstructions are identical with the input ones. Additionally, the compressed generation (CG) process is alike to this of RS1 as they both infer the texture but different as it uses the ground-truth sketch representation. It is worth to note that the end result in both cases share the same quality, which is another indication that the reconstructions of the sketch images are excellent.

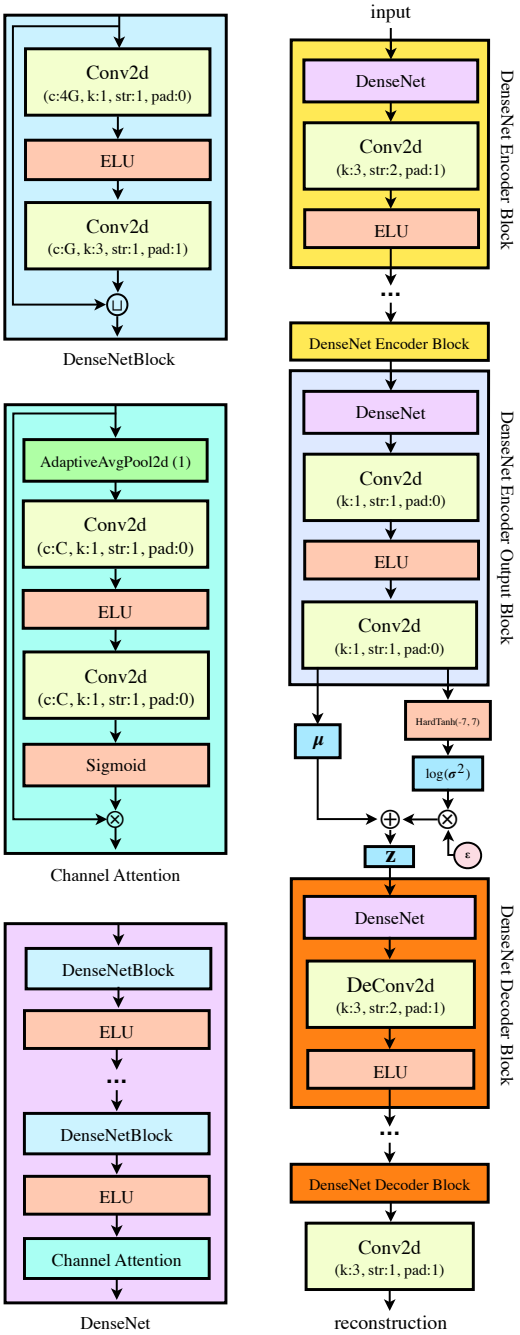


Figure 10: Architecture of our autoencoder. On the right, there are some basic buildings block of the network. The notation as 'G' on the Conv2D channels indicate the growth rate of the densely connected network. The ϵ indicates a random variable drawn from a standard Gaussian, which helps us to make use of the *reparametrization* trick. Until $\mathbf{z}$, we refer to this architecture as *Encoder NN* and thereafter as *Decoder NN*. The former and the later form the *building blocks* to every model that we train and evaluate.

A.4 INTERPOLATION THROUGH LATENT SPACES AND CONDITIONAL GENERATIONS

In Figure 12 we visualise i) interpolations through two ground-truth images through the latent code $\mathbf{u}$ and ii) image reconstructions, where we keep the latent code $\mathbf{u}$ but varying all the others ($\mathbf{z}_1$ and $\mathbf{z}_2$)

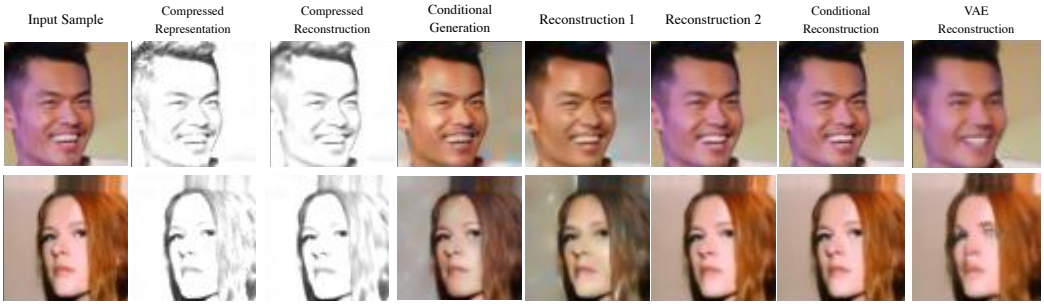


Figure 11: Qualitative results illustrating all the reconstruction techniques on CelebA for selfVAE-sketch.

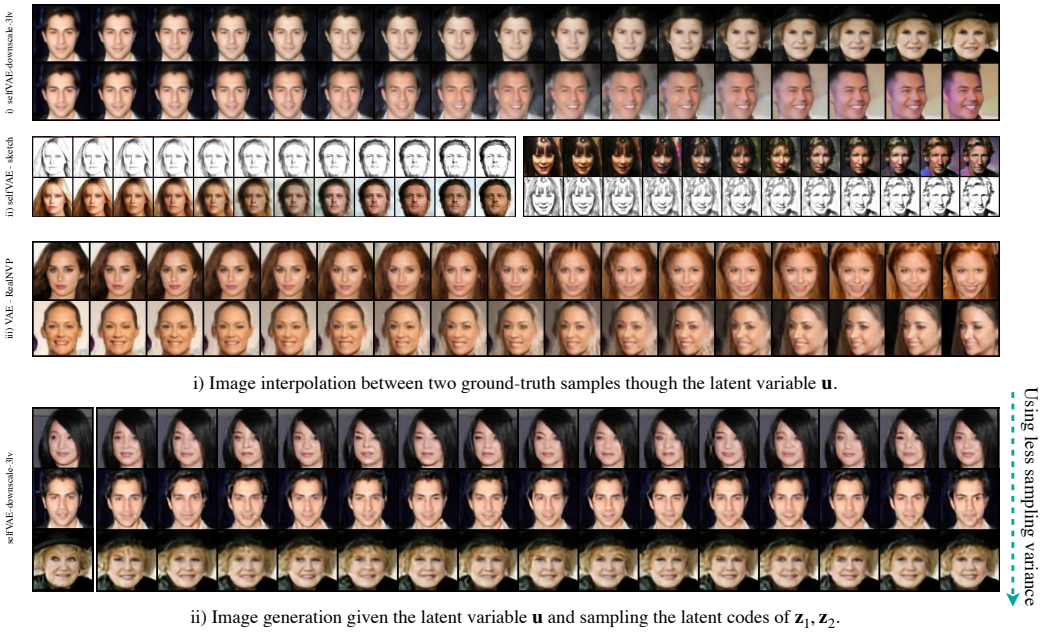


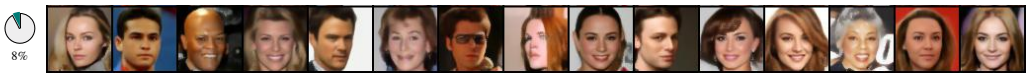
Figure 12: Latent space interpolations and conditional generations of the selfVAEs.

of the 3-leveled selfVAE architecture. Just like the previous cases, in the first case, we see that the model incorporated a rich latent space $\mathbf{U}$, which is responsible for the generation and construction of the global structure of the image. Moving from one latent code $\mathbf{u}$ of a given image to another, we obtain meaningful modifications of the image that result in images that share characteristics from both of them. However, in the latter case, we see that we can alter only high-level features of the image when we keep the values of $\mathbf{u}$ but vary the others; $\mathbf{z}_1$ and $\mathbf{z}_2$. Interestingly, we see that given that ground-truth image that is illustrated on the very left, we can sample different expressions and characteristics of the same person, as the latent variable $\mathbf{u}$ is kept constant.

A.5 Additional Results

Additional results of reconstructions for CelebA are shown in Figure 13.

Ground-truth data samples



i) VAE with RealNVP prior reconstructions

ii) selfVAE 3-level downsample model reconstructions using; (a) **First row:** 1 latent code ($\mathbf{u}$);
(b) **Second row:** 2 latent codes ($\mathbf{u}, \mathbf{z}_1$); (c) **Third row:** all three latent codes ($\mathbf{u}, \mathbf{z}_1, \mathbf{z}_2$); **Fourth row:** latent code $\mathbf{z}_1$ and $\mathbf{y}_2$

iii) selfVAE 2-level sketch model reconstructions using; (a) **First row:** 1 latent code ($\mathbf{u}$);
(b) **Second row:** 2 latent codes ($\mathbf{u}, \mathbf{z}_2$).

Figure 13: Comparison on image reconstructions with different amount of sent information.