IS THE SPARSITY OF HIGH DIMENSIONAL SPACES THE REASON WHY VAES ARE POOR GENERATIVE MODELS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Variational autoencoders (VAE) encode data into lower-dimensional latent vectors before decoding those vectors back to data. Once trained, decoding a random latent vector from the prior usually does not produce meaningful data, at least when the latent space has more than a dozen dimensions. In this paper, we investigate this issue drawing insight from high dimensional physical systems such as spin-glasses, which exhibit a phase transition from a high entropy random configuration to a lower energy and more organised state when cooled quickly in the presence of a magnetic field. The latent vectors of a standard VAE are by construction distributed uniformly on a hypersphere, and thus similar to the high entropy spin-glass state. We propose to formulate the latent variables of a VAE using hyperspherical coordinates, which allows compressing the latent vectors towards an island on the hypersphere, thereby reducing the latent sparsity, analogous to a quenched spin-glass. We propose a new parametrization of the latent space with limited computational footprint that improves the generation ability of the VAE.

1 INTRODUCTION

In today's machine learning landscape, and deep learning in particular, one of the main mathematical tools to represent data (e.g. images) are high dimensional (HD) Euclidean spaces.

However, our intuition about Euclidean geometry stems from the physical world and everyday life, which are low dimensional spaces (mostly two and three dimensions). This presents a challenge because HD spaces behave, mathematically speaking, in very different ways than their low dimensional counterparts, often in ways that seem counterintuitive or even paradoxical if interpreted through low dimensional intuition.

HD spaces have been used in physics, for example to model the state space of systems such as magnetic materials. We will argue in this paper that the state space of some physical systems studied in statistical physics has a remarkable similarity with the HD spaces created by generative models, in particular Variational Autoencoders (VAE) (Diederik P Kingma, 2013). We will discuss how our handling of HD spaces in machine learning can benefit from the intuition about those physical systems.

We will highlight how issues associated with VAE are related to volume and entropy, that create voids and sparsity in latent spaces, hampering their performance to generate meaningful new samples, even when reconstruction metrics can be optimized very well.

Finally, we will propose a method that parametrize latent data on the hypersphere with hyperspherical coordinates. This allows manipulating the data distribution on the latent manifold more effectively. In particular, we use it to compress the latent manifold volume and reduce the sparsity. This is made possible thanks to an efficient transformation between Cartesian and hyperspherical coordinates, which can be implemented with minimal computational overhead using a fully vectorized algorithm, for high enough dimensions (it becomes costly in the very large case; but, as our results show, those cases are of no practical interest from the point of view of the metrics we are checking, see Fig.2).

1.1 VARIATIONAL AUTOENCODER

An autoencoder (AE) is a common self-supervised method to encode the data into a latent space of lower dimension n. Variational autoencoder (VAE) (Kingma & Welling, 2019) introduces a prior to control the distribution of the latent, allowing to sample the latent space and untangle the dimensions (Rolínek et al., 2019; Bhowal et al., 2024).

Considering some dataset $X = \{x_i\}_{i=1}^{N}$ consisting of N independent samples $x \in \mathcal{R}$. It is further assumed that x is related to some underlying unknown unobserved continuous latent variable z of dimension n. One would like to estimate a probabilistic encoder $p_\theta(z|x)$, defined by unknown parameters θ. Using Bayes, $p_\theta(z|x) = p_\theta(x|z)p_\theta(z)/p_\theta(x)$ but computing $p_\theta(x)$ is intractable since it requires to sum over all possible z: $p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz$

One can model $p_\theta(z|x)$ with $q_\phi(z|x)$ whose parameters ϕ can be estimated using the VAE framework by maximising the so-called Evidence Lower Bound (ELBO):

$$\mathcal{L}(\theta, \phi, x) = \mathbb{E}_{q_\phi(z|x)} \log\left(p_\theta(x|z)\right) - KLD(q_\phi(z)||p_\theta(z)) \tag{1}$$

where *KLD* refers to the Kullback-Leibler divergence.

$q_\phi(z|x)$ is a probabilistic encoder: 'for each data point x, it produces a distribution of the code z in the latent space Z, from which the data point could have been generated' (Diederik P Kingma, 2013). In the VAE framework, a decoder $p_\theta(x|z)$ is defined to approximate a marginal inference of x given the latent z. The parameters θ and ϕ are estimated through ELBO maximisation, using a reparameterization trick allowing to chain the gradients from the decoder to the encoder. The classic VAE assumes that z is distributed as an independent multivariate Gaussian: $\log q_\phi(z|x) = \log \mathcal{N}(z; \mu, \sigma)$ (Diederik P Kingma, 2013; Kingma & Welling, 2019).

The first term of the ELBO equation 1 is the data fidelity term, typically the mean squared error $MSE(x, x_z)$. It tries to minimise the error between the output of the decoder $p_\theta(x|z) = x_z$ with the original data x. The second term of the ELBO is a regularisation term and enforces z to match the prior distribution $p(z)$, a multivariate Gaussian, using the Kullback-Leibler divergence: KLD(z, ϵ), with $\epsilon = \mathcal{N}(0, 1)$. In order to backpropagate the gradient, a reparameterization trick is used: the probabilistic encoder estimates μ and σ, and z is computed as $z = \mu + \epsilon \odot \sigma$, where $\odot$ denotes element-wise multiplication.

Eventually, the two terms of the ELBO can be computed over a mini batch of size N_b:

$$\text{MSE}(x, x_z) = \frac{1}{N_b}\sum^{N_b}(x - x_z)^2,\ \text{KLD}(z, \epsilon) = \frac{1}{2}\sum^{N_b}\sum^{n}\left(1 + \log(\sigma^2) - \mu^2 - \sigma^2\right) \tag{2}$$

The final cost to be optimized weighs the two terms with the gain β (Higgins et al., 2017):

$$\mathcal{L} = \text{MSE}(x, x_z) + \beta\text{KLD}(z, \epsilon) \tag{3}$$

The prior, and thus the latent, is a high dimensional independent multivariate Gaussian, which has specific properties that we briefly recall in the next section, by way of background for the following sections.

1.2 HIGH DIMENSIONAL SPACES IN MATHEMATICS

A multivariate Gaussian sampling in a HD Euclidean space of dimension n is such teh probability to find samples close to the origin is close to zero (despite having maximum probability density) and most of the samples lie close to a $(n-1)-$hypersphere, $\mathbb{S}^{n-1}_{\sqrt{n}}$, of radius $\sqrt{n}$. The distribution of the norm of those samples follows a $\chi(n)$ distribution. Therefore, the samples are located within a region very close to the hypersphere, region which becomes very thin in high dimensions, relative to the radius $\sqrt{n}$. These effects are called 'concentration of measure', in the mathematical literature.

As n increases, a multivariate Gaussian tends towards the uniform distribution on that hypersphere. In addition, any two samples from $\mathcal{N}(0, I_n)$ are always almost orthogonal to each other (this is

called almost-orthogonality). A formal description of these phenomena can be found elsewhere (Vershynin, 2018). See also Appendix A.4.

These facts are closely related to how *(hyper-)volume* behaves in HD spaces: if we consider the standard uniform measure on the hypersphere, then most of its volume or mass is *concentrated* in very thin '*equatorial*' bands for *any* randomly chosen north pole (this is, of course, just an intuitive statement, for a formal description see Wainwright (2019)). The contrast with our intuition coming from two dimensional spheres is remarkable. In the next section, we will review that similar HD spaces exist for spin-glass systems.

1.3 High Dimensional Spaces in (Statistical) Physics

Consider a system consisting of n persons each simultaneously tossing a coin. After the tossing, we can record the result with a vector $x \in \{H, T\}^n$ (H for heads, T for tails). For example, it could be $x_0 = (H, H, T, H, ..., T)$. Each of these vectors is called a possible *microstate* for the system.

Given a microstate, we could define a function $F_H(x)$ on microstates which for example, counts how many heads are in that microstate: the value obtained is called a *macrostate*. Different microstates can give rise to the same macrostate. In statistical mechanics, the (Boltzmann) entropy of a system is proportional to the natural logarithm of the number of different microstates giving rise to a same macrostate. The configurations that maximize this number (and, thus, also entropy) are the thermal equilibrium ones.

For a general continuum microstate space, the thermal equilibrium configurations are characterized by the Gibbs probability measure: $\mu(\mathrm{d}x) \propto e^{-\beta \mathcal{H}(x)} \nu_0(\mathrm{d}x)$, where β is the inverse temperature, $\nu_0(\mathrm{d}x)$ is a fixed reference measure on the submanifold of $\mathbb{R}^n$ formed by all the possible microstates, and $\mathcal{H}(x)$ is the energy function of the system (Bricmont, 2022).

There's a class of systems that have been extensively studied called *spin glasses* (Parisi, 2002). In these systems, the submanifold of microstates is given precisely by the $(n-1)$-hypersphere, $\mathbb{S}^{n-1}_{\sqrt{n}}$, and $\nu_0(\mathrm{d}x)$ by the standard uniform measure on it. The dimension n is taken to be very large. The uniform measure is one of maximal entropy, even among all the measures of thermal equilibrium.

At very high temperatures, $\mu(\mathrm{d}x) \approx \nu_0(\mathrm{d}x)$, two 'replicas' of the system are two *i.i.d.* samples from the Gibbs measure. Their overlap, measured by their inner product, will be almost zero, because of the almost-orthogonality effect of HD spaces mentioned before. This means that these samples are in the 'equatorial' region, where most of the volume is concentrated. The system is said to be in a 'replica symmetric phase'.

Until a critical temperature value, all solutions have this qualitative behaviour. This is to be expected since the function whose stationary points define a general solution for these systems is indifferent to a permutation of replicas. What is surprising in these systems is the emergence at low enough temperatures of a different phase, which is *not* replica symmetric; that is, the inner product or correlation between them is appreciably different from zero. This is called the 'replica symmetry breaking phase' (Montanari & Sen, 2024). Two replicas, in this case, lie in a very thin band or ring centered at a deep minimum of the energy function (Subag, 2017), center which is not in the initial high volume equatorial region. This ring is a $(n-2)$-hypersphere, submanifold of the initial $(n-1)$-hypersphere.

We discussed so far the 'static' structure of the Gibbs measure across the temperature range. A separate and complicated question is how, starting from a high temperature state, one can reach *dynamically*, by some physical process[1], the replica symmetry breaking phase; that is, to dynamically achieve this *phase transition*.

This can be very tricky, since the local minima landscape of the energy in the high entropy region is very rugged (with exponentially many local minima), and one would need to overcome it first in order to reach the less entropic regions of the hypersphere (Arous & Jagannath, 2024). In practice, it's usually done by 'quenching' processes, where the system is suddenly cooled down in the presence of externally applied magnetic fields. The energy function for a $p-$spin glass in the presence of an

[1]Typically, a Langevin dynamics of the form $\mathrm{d}x_t = \mathrm{d}B_t - \beta \nabla \mathcal{H}(x_t)\mathrm{d}t$, for which the Gibbs measure is stationary.

external magnetic field $\mathbf{h}$ is given by $\mathcal{H}(x) = \mathcal{P}_p(x) - \sum_j h_j x_j$, where $\mathcal{P}_p(x)$ is a homogeneous random $p-$polynomial (we omit the normalization constants).

The high entropy 'replica symmetric phase' of the spin-glass is analogous to the training of a VAE where the KL term forces the latent samples to be uniformly distributed on the hypersphere (the dynamics of spin glasses and the training process of some deep learning models has already received some attention in the literature, for example in Baity-Jesi et al. (2018), and also Arous et al. (2022); in this work, though, we will focus our attention on some other issues, more geometrical in nature). In the next section, we argue that this brings an issue related to the sparsity of the resulting latent space.

1.4 High Dimensional Spaces in Generative Models (VAE)

One use case of the VAE is to generate data. Since the latent distribution is known (high dimensional multivariate Gaussian), one could sample from that distribution and decode the latent vector to generate a novel data sample. This works very well for simple data and low dimension latent spaces (e.g., MNIST with $n = 2$, as done in Diederik P Kingma (2013)), but not so well for more complicated data or high dimensional latent (e.g., CIFAR10, or MNIST with $n = 32$ as in Cinelli et al. (2021)). Indeed, variations of the VAE are commonly used in generative models of images and videos (Jonathan Ho Ajay Jain, 2020), but the latent space is not sampled directly: instead, a reverse diffusion process dynamically transforms a random sample into a valid latent location.

One of the reasons why VAE perform poorly when sampling directly the high dimensional latent is because of conflicting constraints. On the one hand, high resolution images need many latent dimensions to capture all the information they convey. Then, the VAE KL divergence term in the loss encourages the system to distribute this HD latent uniformly on the hypersphere, that is, to maximize entropy. By doing so, the latent becomes extremely sparse given the number of training samples and the immensity of the latent hyperspherical manifold (the volume growing exponentially with the number of dimensions; in these regimes, trying to find a *specific* microstate is akin to a '*needle in the haystack*' kind of situation). The sparsity hampers any attempt to model the latent as a continuous manifold (Peng et al., 2023). But, on the other hand, this usually leads to meaningless decoding of random samples from the prior. These two opposing forces (sparsity due to HD spaces needed to model high resolution images vs need for a continuous manifold for meaningful generation from all of the latent space) conspire to severely limit the capacity of VAEs to function as generative models.

From the previous sections, the regions of high entropy are such that, for a given macrostate, there's an enormous number of different microstates that can realize it. For tasks such as clustering, generation, interpolation, etc., one is interested in specific macrostates of the system. The disordered high multiplicity of possible microstates that can give rise to these macrostates in the high entropy regime may hinder the ability of the model to perform these tasks, as we will explicitly show in the experimental section.

Since the samples in latent space live on the hypersphere, it comes naturally to consider using hyperspherical coordinates to describe latent variables.

2 Related works

'Hyperspherical Variational Auto-Encoder' Davidson et al. (2018) proposed replacing the standard Euclidean KL divergence with a KL divergence between a uniform distribution on the hypersphere as a prior, and a von Mises-Fisher distribution as an approximate posterior. Elaborating in that direction, Yang et al. (2023) used a von Mises-Fisher mixture model rather than a single distribution, which 'leads to spherical latent embeddings that are well-suited for clustering'.

A different way of building a Hyperspherical Variational Auto-Encoder is proposed in Bonet et al. (2022), based on a spherical Sliced-Wasserstein discrepancy, and as an extension of the well-known Euclidean models (Soheil Kolouri Phillip E. Pope, 2020). Hyperspherical aspects of data are studied in Löwe et al. (2023), where high dimensional Rotating Features are introduced. Of particular interest for our project, the authors remarked:

'*We represent Rotating Features in Cartesian coordinates rather than (hyper) spherical coordinates, as the latter may contain singularities that hinder the model's ability to learn good representations. [...]. This representation can lead to singularities due to dependencies between the coordinates. For example, when a vector's radial component (i.e. magnitude) is zero, the angular coordinates can take any value without changing the underlying vector. As our network applies ReLU activation on the magnitudes, this singularity may occur regularly, hindering the network from training effectively*'.

As we reviewed in Section 1.2, in high dimensions, the random samples of an independent multivariate Gaussian distribution fall in the equator of a hypersphere, and thus none of them is near the singularities of the hyperspherical coordinates (the poles and the center of the hypersphere).

While the problem of formulating latent spaces given by non-Euclidean Riemannian manifolds, and hyperspheres in particular has been studied, explicit use of hyperspherical coordinates is avoided. At first look, the conversion from Cartesian to hyperspherical coordinates seems to require computationally expensive recurrent trigonometric formulas (see Appendix A.1). Instead, formulations of Riemannian geometry that rely on Cartesian coordinates are used. Riemannian geometry is not just about a curved metric, but also being able to express it in a convenient coordinate system *adapted* to it (of central importance in physics).

The possible use of hyperspherical coordinates has thus been discarded by all the works that we reviewed. Notwithstanding the previous arguments, we believe that the use of hyperspherical coordinates is feasible and can be beneficial, as our results show. In Appendix A.2 we provide a vectorized implementation for transforming between hyperspherical and Cartesian coordinates, which adds only a small computational overhead for training a VAE.

Even if one could assemble a sufficiently large training dataset to densely populate the latent hypersphere (an impossible task in practice), the data in high dimensional latent spaces tend to form manifolds with complicated topologies that include holes and cracks. If a random sample falls into one of these holes, it will be decoded into something meaningless. This also affects interpolations, since locations sampled on a trajectory between two valid latent locations are likely to fall between clusters or classes (in the holes and cracks). This problem has been noted for VAE and called the 'prior hole problem' (Tomczak; Lin & Clark, 2020; Cinelli et al., 2021; Asperti et al., 2021; Singh & Ogunfunmi, 2022; Hao & Shafto, 2023; Aneja et al., 2021). It highlights that the approximate posterior of the data often fails to perfectly match the prior: in the case of a Gaussian distribution in high dimensions the uniform distribution on the hypersphere has no hole.

Common approaches for tackling this problem include a learnable prior (rather than static, as in the standard VAE), or using a mixture of Gaussians as a prior (rather than a single one, as in the standard VAE); see Tomczak for discussion of both cases. A different approach is to dispose of the continuum completely and work with discrete latent representations, as in Aaron van den Oord (2018) where the latent vectors are quantized (VQ-VAE). These discrete representations are useful for tasks dealing with a discrete sequence of symbols, otherwise a joint distribution on the dictionary needs to be estimated after the VAE is trained (Salimans et al., 2017).

There are applications that necessitate and use VAEs with a latent continuum. For example, generative models for drug discovery often deal with chemical properties that span a continuous spectrum (e.g. measure of synthesizability by living entities Ochiai et al. (2023)).

In the next section we formulate a VAE with the latent variables described using hyperspherical coordinates.

3 METHOD: VAE WITH HYPERSPHERICAL COORDINATES

Our approach is based on formulating the initial KL divergence term with a prior from the original VAE, which is in Cartesian coordinates, to one in hyperspherical coordinates. See Appendix A.1 for the standard conversion formulas between Cartesian and hyperspherical in high dimension.

In cartesian coordinates, the KL divergence between the estimated posterior defined by μ_k and σ_k and the prior defined by μ_k^p and σ_k^p is (Odaibo, 2019):

$$KLD^{w/Prior}_{\text{CartCoords}} = \frac{1}{2}\sum_{k=1}^{n}\left[\left(\frac{\sigma_k}{\sigma_k^p}\right)^2 - \ln\left(\frac{\sigma_k}{\sigma_k^p}\right)^2 - 1 + \frac{(\mu_k-\mu_k^p)^2}{(\sigma_k^p)^2}\right] \tag{4}$$

A Taylor approximation (up to second order) of the part for sigma around its prior yields for some constants γ_k and $\widetilde{\gamma}_k$:

$$KLD^{w/Prior}_{\text{CartCoords}} \approx \sum_{k=1}^{n}\left[\gamma_k\left(\sigma_k-\sigma_k^p\right)^2 + \widetilde{\gamma}_k\left(\mu_k-\mu_k^p\right)^2\right] \tag{5}$$

In practice, the optimization is performed over mini batches of data (of size N_b), using the objective below:

$$KLD^{w/Prior}_{\text{CartCoords}} \approx \frac{1}{N_b}\sum_{l=1}^{N_b}\sum_{k=1}^{n}\left(\gamma_k\left(\sigma_{k,l}-\sigma_k^p\right)^2 + \widetilde{\gamma}_k\left(\mu_{k,l}-\mu_k^p\right)^2\right) \tag{6}$$

If we denote the corresponding batch statistics as $\mathbb{E}_b$ and σ_b, then, by using the basic formula,

$$\mathbb{E}_b[X^2] = \mathbb{E}_b[X]^2 + \sigma_b[X]^2, \tag{7}$$

we can write this objective as (we omit the constants for ease of reading)

$$KLD^{w/Prior}_{\text{CartCoords}} \approx \sum_{k=1}^{n}\left(\left(\mathbb{E}_b[\sigma_k]-\sigma_k^p\right)^2 + \sigma_b[\sigma_k]^2 + \left(\mathbb{E}_b[\mu_k]-\mu_k^p\right)^2 + \sigma_b[\mu_k]^2\right) \tag{8}$$

So far, not much has been gained other than rewriting the KL function (in Cartesian coordinates) in terms of the batch statistics. This rewriting was partly inspired by the construction in Bardes et al. (2021), and will be useful for our next step.

The similarity with the spin glass we described in section 1.3 is emerging from this expression. The terms of the form $(x-x_0)^2 = x^2 - 2xx_0 + x_0^2$ can be interpreted in the following manner: x^2 contributes to the homogeneous polynomial part of the energy function, $-2xx_0$ corresponds to an external magnetic field $2x_0$, while x_0^2 is a just an inconsequential constant term. The VAE is not exactly a spin glass though, since the reconstruction part of the loss, given by the standard MSE, is not a homogeneous polynomial. However, the analogy provides insight into the replica symmetry breaking as a way to move away from the equator, and how the application of external magnetic fields in the adequate manner can help achieve this, as we will see below, by applying these 'magnetic fields' in all the *angular directions* provided by hyperspherical coordinates.

We now introduce hyperspherical coordinates in the KL formulation. We start with the Cartesian coordinates (μ_i, σ_i), given by the encoder, and transform these to their hyperspherical counterparts $(\overset{\mu}{r}, \overset{\mu}{\varphi}_k; \overset{\sigma}{r}, \overset{\sigma}{\varphi}_k)$ with r a scalar and k the index of the $n-1$ spherical angles.

The KL-like objective becomes for the angles φ_k:

$$\begin{aligned}\text{KLD}^{w/Prior}_{\text{HSphCoords}}(\varphi_k) = \sum_{k=1}^{n-1}\Bigg(&\alpha_{\sigma,k}\left(\mathbb{E}_b[\cos\overset{\sigma}{\varphi}_k]-a_{\sigma,k}\right)^2 + \beta_{\sigma,k}\left(\sigma_b[\cos\overset{\sigma}{\varphi}_k]-b_{\sigma,k}\right)^2\\ &+\alpha_{\mu,k}\left(\mathbb{E}_b[\cos\overset{\mu}{\varphi}_k]-a_{\mu,k}\right)^2 + \beta_{\mu,k}\left(\sigma_b[\cos\overset{\mu}{\varphi}_k]-b_{\mu,k}\right)^2\Bigg)\end{aligned} \tag{9}$$

and for the norm r:

$$\begin{aligned}\text{KLD}^{w/Prior}_{\text{HSphCoords}}(r) = &\alpha_{\sigma,r}\left(\mathbb{E}_b[\overset{\sigma}{r}]-a_{\sigma,r}\right)^2 + \beta_{\sigma,r}\left(\sigma_b[\overset{\sigma}{r}]-b_{\sigma,r}\right)^2\\ &+\alpha_{\mu,r}\left(\mathbb{E}_b[\overset{\mu}{r}]-a_{\mu,r}\right)^2 + \beta_{\mu,r}\left(\sigma_b[\overset{\mu}{r}]-b_{\mu,r}\right)^2\end{aligned} \tag{10}$$

with the priors for the mean over the batch $a_{i,j}$, the standard deviation over the batch $b_{i,j}$, and the gains for each term $\alpha_{i,j},\ \beta_{i,j}$, for $i \in \{\sigma, \mu\}$ and $j \in \{1, ..., n-1, r\}$

We use the cosines rather than the angles to avoid costly extra computations of the corresponding arccosines (Appendix A.1). The reparameterization trick is still done in the Cartesian coordinates representation. The coordinate transformation is done using a vectorized implementation (code provided in Appendix A.2). The coordinate transformation and the extra KLD terms add about 32% computation time (200 samples per batch, $n = 200$). For more dimensions the increase is higher. The final cost to be optimized weighs the reconstruction term and KL terms with an overall gain β for similarity with the standard βVAE (3):

$$\mathcal{L} = \text{MSE}(x, x_z) + \beta \left(\text{KLD}^{w/Prior}_{\text{HSphCoords}}(\varphi_k) + \text{KLD}^{w/Prior}_{\text{HSphCoords}}(r) \right) \tag{11}$$

3.1 Volume Compression of the latent manifold

We discussed previously that the standard VAE forces the latent samples to be uniformly distributed on the hypersphere, maximising the entropy, which results, in high dimensions, in the data being located within equators of the hypersphere where the volume is the greatest. A benefit of using hyperspherical coordinates is the possibility to set a prior for the φ_k that forces the latent samples away from the equator, thereby escaping these highly entropic regions. This can be done for each angular coordinate, which are all uncorrelated with each other, by simply setting

$$a_{\mu,k} \neq 0,\ \forall k. \tag{12}$$

By doing so, the samples can be moved to a zone with much less volume, thereby increasing the density of the latent, with the hope that random samples from that denser region will have better quality decoding because of the reduced sparsity. This can be seen more directly by analysing the hypervolume element of the hypersphere in hyperspherical coordinates. The volume can be reduced much faster and effectively by reducing the angular coordinates (away from the equators), than by either reducing just the radius of the hypersphere or, equivalently, all of the Cartesian coordinates.

The higher the dimension, the more pronounced this difference becomes because each added dimension k adds extra powers of $\sin\varphi_k$ in the hypervolume element (Appendix A.3). Then, the further the angles from $\pi/2$, the smaller the infinitesimal hypervolume element becomes as it is multiplied by an increasingly smaller quantity lower than 1. This is a purely geometric effect. It can already be easily seen in the two-dimensional sphere, where a spherical coordinate rectangle of unvarying angular coordinates size has smaller area when moved away from the equator towards any of the poles. Thus, high dimensions bring the problem of high entropy in the equators, but also a non-Euclideanity to the manifold; we explored to which extent one can take advantage of the latter to mitigate the former.

Finally, by setting

$$a_{\mu,r} = \sqrt{n} \tag{13}$$

(and normalizing z, after sampling via the reparameterization trick, to the same radius $\sqrt{n}$) we can force the latent samples to be on the hyperspherical surface of that radius.

4 Experimental Results

4.1 Model and implementation

For all our experiments, we use a ResNet-type architecture (He et al., 2015) for both encoder and decoder. When using the loss in hyperspherical coordinates (11), we use an annealing schedule (Fu et al., 2019) for the gain β of the KL-like loss, consisting of an initial stage which increases proportionally with $\sqrt{\text{epoch}}$ for the first 100 epochs, and is constant afterwards. This was necessary because we observed that too much compression of the volume was detrimental to the performance,

while a strong compression was still necessary at the initial stage. The total training was 300 epochs in all cases.

4.2 CHOOSING THE GAIN FOR EACH LOSS TERMS

The constants $\alpha_{i,j}$, $\beta_{i,j}$ multiplying the elements of the hyperspherical loss are proportional to $1/\sqrt{k}$, where k is the coordinate index. This was necessary because, unlike the Cartesian coordinates, the hyperspherical coordinates are asymmetric and vary with k. This can be seen in the transformation formulas (Appendix A.1), where a product of an increasing amount of sine functions is necessary as the coordinate index increases. We chose $1/\sqrt{k}$, guided by the fact that the vector whose Cartesian coordinates are $(1, 1, ..., 1)$ has a cosine of its spherical angles equal to $1/\sqrt{k}$ as the coordinate index, and because it gave the best results experimentally.

In this way, we were able to avoid lengthy calculations to obtain the mathematically exact formulas for both these constants and the KLD in hyperspherical coordinates, which we do not believe, anyway, to be of the most importance for the particular goals we had in this work.

4.3 VISUALISATION OF THE HIGH ENTROPY LATENT IN STANDARD VAE

A standard VAE with 128 latent dimensions was trained using the MNIST dataset (Y.LeCun et al, 1998). Generating and decoding random samples from the prior latent resulted in meaningless decoded/generated data (Fig.1a), left panel).

The latent hypersphere can be visualized in 3D as shown in Fig.1b), left. This was done by averaging the 128 latent dimensions into three (first 42, second 42, and the remaining 44), and normalizing each of the resulting 3D vectors to the sphere. Each latent vector could thus be plotted as a point in 3D, and shows a uniform-like distribution on the 2D sphere as expected.

This visualisation allows us to directly see the entropy. A k-NN classifier for the 10 classes of MNIST from the latent had an accuracy of 0.95, and 10 clusters can readily be seen when projecting the 128 latent dimensions into 2 (Fig.1c), left) using t-SNE. However, no particular clustering can be observed on the 3D visualisation (Fig.1b), left). Such a direct visualisation of the latent space *cannot* display clusters, because they are buried into the 'disorder' of the entropy of the hypersphere equators, where most of the samples are located. There are many different possible microstates (points in latent space) that can realize the same macrostate (the clustering).

4.4 IMPROVED GENERATION WHEN THE LATENT MANIFOLD IS COMPRESSED

Next, we train the same VAE using the same dataset but now with the KL-type loss in hyperspherical coordinates (11). The prior was set to *compress* the (hyper-)volume in latent space by using $a_{\mu,k} = 1$, $\forall k$, which pushes *all* the φ_k towards 0. They could not be exactly 0 because the reconstruction would all be the same. The reconstruction term balances the KL term to spread the latent samples away from the angles 0. Recall from section 1.3 that, in the replica symmetry breaking phase, two replicas in that case lie in a very thin band or ring centered at a deep minimum of the energy function, center which is not in the initial high volume equatorial region; this ring is a $(n-2)$-hypersphere, submanifold of the initial $(n-1)$-hypersphere. This situation corresponds to sending the angle φ_1 towards 0, the rest free. Thus, by sending all the angles towards 0, and gven the geometrical interpretation of the hyperspherical coordinates, we aim to induce a similar replica symmetry breaking transition also in the mentioned $(n-2)$-hypersphere, as well as in *all* the remaining sub $(n-k)$-hyperspheres, $\forall k$. We call this process 'Nested Replica Symmetry Breaking' (NRSB), and it's only in this regime where we get the results described below.

In this configuration, for generating new data the latent was not randomly sampled on the whole hypersphere, but from a von Mises–Fisher distribution with the same mean and covariance as the ones empirically calculated from the latent embedding of the full test dataset. These decoded random samples generated data with a quality close to the actual training dataset (Fig.1a), right panel), to be compared to the meaningless decoding of the previous experiment when random sampling from the prior was done (Fig.1a), left panel). By compressing the latent using hyperspherical coordinates, the VAE became a functional generative model, despite having 128 latent variables.

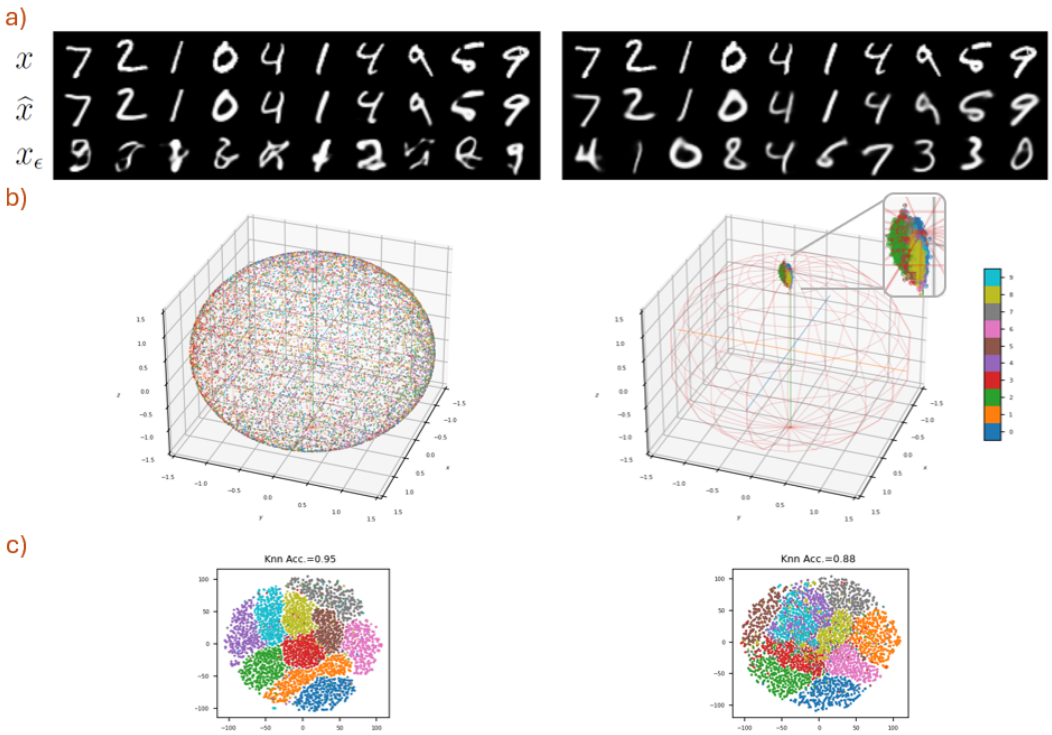


Figure 1: Comparison using MNIST between the standard βVAE (left) and the proposed compressed version (right). The top panel (a) shows the original data (x), the reconstruction ($\hat{x}$), and the generation sampling from the prior (x_ϵ). The middle panel (b) shows the 3D projection on the latent 2D-sphere of the test dataset: the βVAE posterior is a uniform distribution whereas the proposed method compresses the latent vector on a small volume within an island of the hypersphere. The bottom panel (c) shows that in both cases the classes are clustered in the latent (using t-SNE) and that a k-NN classifier achieves good performance, with the compression βVAE resulting in lower accuracy (0.88 Vs 0.95) because the lower volume of the latent manifold forces the classes to overlap more (as seen on the clustering of panel c).

Furthermore, the $3-$dimensional visualization shows something remarkable (Fig.1b), right): besides showing that the latent samples are compressed towards a small 'island' on the hypersphere and away from the equator, the classes are actually *visible*. We believe that it is because the samples are now located away from the equator, in a region with a much lower entropy, where there are many fewer possibilities to realize this clustering in terms of different possible microstates. In other words, the VAE's latent space has made a phase transition. The same k-NN classifier from the latent space shows a similar accuracy, of 0.88, and the class clusters can also be seen in a t-SNE 2D projection (Fig.1c), left).

4.5 Trade off between reconstruction and generation

The reconstruction quality of an AE and VAE improves as the number of latent dimensions increases, as measured by the MSE between x and $\hat{x}$. However, the quality of data generation (from decoding random sampling of the latent) decreases as the number of latent dimensions increases. We have argued in this paper that the later is due to the increased sparsity of the latent, as demonstrated qualitatively in the previous experiment when that sparsity is reduced by compressing the latent using our proposed method.

The quality of randomly generated data can be measured using the Frechet Inception Distance (FID) (Heusel et al., 2017). FID compares the distribution of features between the images of the training/testing dataset and an equivalent number of randomly generated images. We used in this experi-

ment CIFAR10 (Krizhevsky, 2009), a more challenging dataset, and an FID computed using 10,000 samples (we compare the random decoded samples with the *reconstructed* testing set).

In a VAE, the quality of the reconstruction, still measured by the MSE, also varies with the gain β of the loss: the more weight for the KL term, the more the latent matches the prior and the worse the reconstruction (Cf. βVAE Higgins et al. (2017)). We show in Appendix A.5 this behaviour by comparing the results for several values of β.

We can now explore quantitatively the quality of the reconstruction (using MSE) and the quality of the generation (using FID) when the number of latent dimensions increases and the β varies. We compared the standard VAE with our proposed compressed VAE.

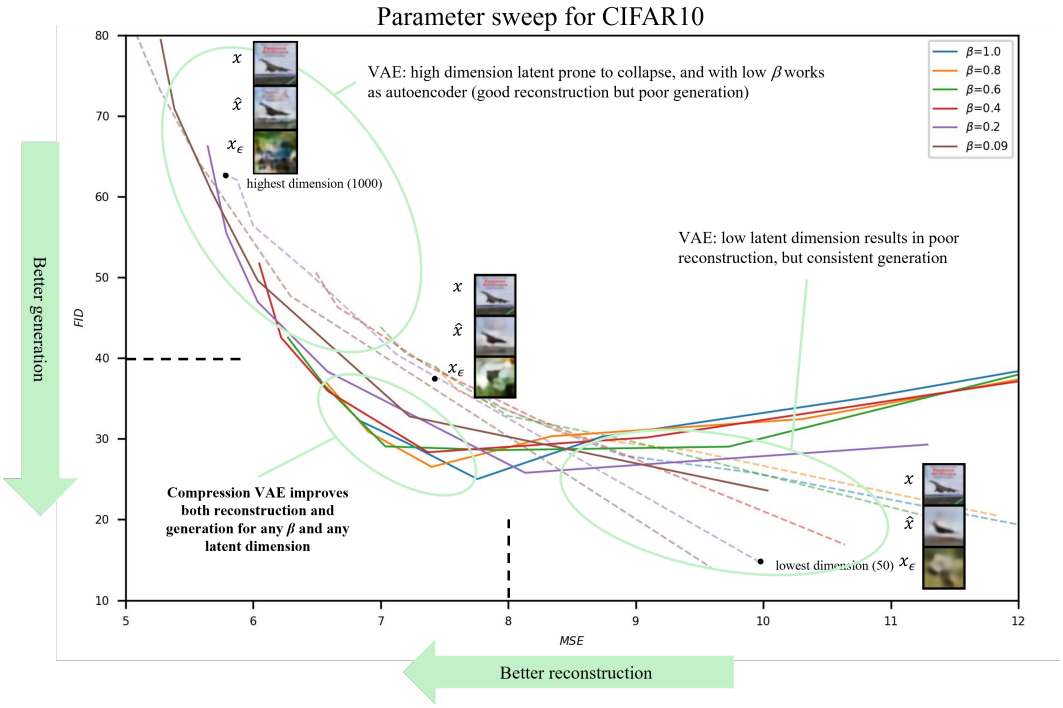


Figure 2: Effect of latent dimension and β on the trade off between reconstruction and generation on CIFAR10. Each curve represents a VAE for a given β while spanning the number of latent dimensions from low (50, bottom right endpoints) to high (1000 top left corner endpoints). The standard VAEs are shown using dashed lines, whereas the compressed versions are shown using solid lines. We excluded from our discussion the regimes where generation was of very poor quality (FID$>$ 40) or the reconstruction was too blurry (MSE$>$ 8), with the best trade off close to the bottom left corner. In that useful area, the compressed VAEs outperformed their standard equivalent for any combination of β and latent size (solid lines closer to the bottom left corner than the dashed lines).

These two metrics (MSE and FID), should give us a good idea regarding how good our models are for the general generative task: the fist measures how 'crisp/sharp' the reconstructed images are, while the second how close the random decoded images resemble images from training dataset. A good generative model should minimize both of these metrics *simultaneously*.

Fig.2 sumarizes the results with more details provided in Appendix A.5. As expected, the MSE decreases as one increases the latent space dimension, while the exact opposite is true for the FID. To obtain a good generative model in the way we defined it can be a very difficult task, involving a very delicate balance between these two opposing trends we just described, and often relying on off-equilibrium configurations.

The results show that the compression VAE version improves on absolute terms over the standard VAE over any combination of β and dimension of the latent.

REFERENCES

Koray Kavukcuoglu Aaron van den Oord, Oriol Vinyals. Neural Discrete Representation Learning. 2018.

Jyoti Aneja, Alexander G Schwing, Jan Kautz, and Arash Vahdat. A Contrastive Learning Approach for Training Variational Autoencoder Priors. Technical report, 2021.

Gérard Ben Arous and Aukosh Jagannath. Shattering versus metastability in spin glasses. *Communications on Pure and Applied Mathematics*, 77(1):139–176, 1 2024. ISSN 10970312. doi: 10.1002/cpa.22133.

Gérard Ben Arous, Reza Gheissari, and Aukosh Jagannath. High-dimensional limit theorems for SGD: Effective dynamics and critical scaling. Technical report, 2022.

Andrea Asperti, Davide Evangelista, and Elena Loli Piccolomini. A Survey on Variational Autoencoders from a Green AI Perspective. *SN Computer Science*, 2(4), 7 2021. ISSN 26618907. doi: 10.1007/s42979-021-00702-9.

Marco Baity-Jesi, Levent Sagun, Mario Geiger, Stefano Spigler, Gérard Ben Arous, Chiara Cammarota, Yann Lecun, Matthieu Wyart, and Giulio Biroli. Comparing Dynamics: Deep Neural Networks versus Glassy Systems. Technical report, 2018.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. 5 2021. URL http://arxiv.org/abs/2105.04906.

Pratik Bhowal, Achint Soni, and Sirisha Rambhatla. Why do Variational Autoencoders Really Promote Disentanglement? Technical report, 2024. URL https://github.com/criticalml-uw/.

Clément Bonet, Paul Berg, Nicolas Courty, François Septier, Lucas Drumetz, and Minh-Tan Pham. Spherical Sliced-Wasserstein. 6 2022. URL http://arxiv.org/abs/2206.08780.

Jean Bricmont. *Making Sense of Statistical Mechanics*. Springer International Publishing, Cham, 2022. ISBN 978-3-030-91793-7. doi: 10.1007/978-3-030-91794-4.

Lucas Pinheiro Cinelli, Matheus Araújo Marins, Eduardo Antônio Barros da Silva, and Sérgio Lima Netto. *Variational methods for machine learning with applications to deep networks*. Springer International Publishing, 5 2021. ISBN 9783030706791. doi: 10.1007/978-3-030-70679-1.

Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical Variational Auto-Encoders. Technical report, 2018. URL https://github.com/nicola-decao/s-vae.

Max Welling Diederik P Kingma. Auto-Encoding Variational Bayes. *arXiv preprint*, 2013. URL https://arxiv.org/abs/1312.6114.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing. 3 2019. URL http://arxiv.org/abs/1903.10145.

Xiaoran Hao and Patrick Shafto. Coupled Variational Autoencoder. Technical report, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. Technical report, 2015. URL http://image-net.org/challenges/LSVRC/2015/.

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. 6 2017. URL http://arxiv.org/abs/1706.08500.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, Alexander Lerchner, and Google Deepmind. β-VAE: LEARNING BASIC VISUAL CONCEPTS WITH A CONSTRAINED VARIATIONAL FRAMEWORK. Technical report, 2017.

Pieter Abbeel Jonathan Ho Ajay Jain. Denoising Diffusion Probabilistic Models. *arXiv preprint*, 2020. URL https://arxiv.org/abs/2006.11239.

Diederik P Kingma and Max Welling. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8237. doi: 10.1561/2200000056. URL https://www.nowpublishers.com/article/Details/MAL-056.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

Shuyu Lin and Ronald Clark. LaDDer: Latent Data Distribution Modelling with a Generative Prior. 8 2020. URL http://arxiv.org/abs/2009.00088.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. *Proceedings of International Conference on Computer Vision (ICCV)*, 12 2015.

Sindy Löwe, Phillip Lippe, Francesco Locatello, and Max Welling. Rotating Features for Object Discovery. 6 2023. URL http://arxiv.org/abs/2306.00600.

Andrea Montanari and Subhabrata Sen. A Friendly Tutorial on Mean-Field Spin Glass Techniques for Non-Physicists. 4 2024. URL http://arxiv.org/abs/2204.02909.

Jean-Christophe Mourrat. An informal introduction to the Parisi formula. 10 2024. URL http://arxiv.org/abs/2410.12364.

Toshiki Ochiai, Tensei Inukai, Manato Akiyama, Kairi Furui, Masahito Ohue, Nobuaki Matsumori, Shinsuke Inuki, Motonari Uesugi, Toshiaki Sunazuka, Kazuya Kikuchi, Hideaki Kakeya, and Yasubumi Sakakibara. Variational autoencoder-based chemical latent space for large molecular structures with 3D complexity. *Communications Chemistry*, 6(1), 12 2023. ISSN 23993669. doi: 10.1038/s42004-023-01054-6.

Stephen G Odaibo. Tutorial: Deriving the Standard Variational Autoencoder (VAE) Loss Function. Technical report, 2019.

Giorgio Parisi. The physical Meaning of Replica Symmetry Breaking. Technical report, 2002.

Dehua Peng, Zhipeng Gui, and Huayi Wu. Interpreting the Curse of Dimensionality from Distance Concentration and Manifold Effect. Technical report, 2023.

Michal Rolínek, Dominik Zietlow, and Georg Martius. Variational Autoencoders Pursue PCA Directions (by Accident). Technical report, 2019.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. 1 2017. URL http://arxiv.org/abs/1701.05517.

Aman Singh and Tokunbo Ogunfunmi. An Overview of Variational Autoencoders for Source Separation, Finance, and Bio-Signal Applications, 1 2022. ISSN 10994300.

Charles E Martin Gustavo K Rohde Soheil Kolouri Phillip E. Pope. Sliced-Wasserstein Autoencoder: An Embarrassingly Simple Generative Model. 2020. URL https://arxiv.org/abs/1804.01947.

Eliran Subag. The geometry of the Gibbs measure of pure spherical spin glasses. *Inventiones Mathematicae*, 210(1):135–209, 10 2017. ISSN 00209910. doi: 10.1007/s00222-017-0726-4.

J Tomczak. Priors (blogpost). URL https://jmtomczak.github.io/blog/7/7_priors.html.

Roman Vershynin. *High-Dimensional Probability*. Cambridge University Press, 4 2018. ISBN 9781108231596. doi: 10.1017/9781108231596. URL https://doi.org/10.1017/9781108231596.

Martin J. Wainwright. Concentration of measure. In *High-Dimensional Statistics*, chapter 3, pp. 58–97. Cambridge University Press, 2 2019. doi: 10.1017/9781108627771.004. URL https://www.cambridge.org/core/books/highdimensional-statistics/concentration-of-measure/A649A3B05DC79C2B10BF1C80CC6F5F10.

Lin Yang, Wentao Fan, and Nizar Bouguila. Deep Clustering Analysis via Dual Variational Autoencoder With Spherical Latent Embeddings. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):6303–6312, 9 2023. ISSN 21622388. doi: 10.1109/TNNLS.2021.3135460.

Y.LeCun et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 11 1998.

A Appendix

A.1 Conversion between cartesian and hyperspherical coordinates

For reference, we note here the standard formulas for converting between cartesian and spherical coordinates (as they appear in https://en.wikipedia.org/wiki/N-sphere).

In n dimensions, given a set of cartesian coordinates x_k with $k \in \{1, \ldots, n\}$, the hyperspherical coordinates are defined by a radius r and $n-1$ angles φ_k with $k \in \{1, \ldots, n-1\}$; $\varphi_k \in [0, \ldots, \pi]$ for $k \in \{1, \ldots, n-2\}$ and $\varphi_{n-1} \in [0, \ldots, 2\pi)$.

From hyperspherical to cartesian conversion:

$$\begin{aligned}
x_1 &= r\cos(\varphi_1) \\
x_2 &= r\sin(\varphi_1)\cos(\varphi_2) \\
x_2 &= r\sin(\varphi_1)\sin(\varphi_2)\cos(\varphi_3) \\
&\vdots \\
x_{n-1} &= r\sin(\varphi_1)\sin(\varphi_2)\ldots\sin(\varphi_{n-2})\cos(\varphi_{n-1}) \\
x_n &= r\sin(\varphi_1)\sin(\varphi_2)\ldots\sin(\varphi_{n-2})\sin(\varphi_{n-1})
\end{aligned} \tag{14}$$

From cartesian to hyperspherical conversion:

$$\begin{aligned}
r &= \sqrt{x_n^2 + x_{n-1}^2 + \ldots + x_2^2 + x_1^2} \\
\cos(\varphi_1) &= \frac{x_1}{\sqrt{x_n^2 + x_{n-1}^2 + \ldots + x_2^2 + x_1^2}} \\
\cos(\varphi_2) &= \frac{x_2}{\sqrt{x_n^2 + x_{n-1}^2 + \ldots + x_2^2}} \\
&\vdots \\
\cos(\varphi_{n-2}) &= \frac{x_{n-2}}{\sqrt{x_n^2 + x_{n-1}^2 + x_{n-2}^2}} \\
\cos(\varphi_{n-1}) &= \frac{x_{n-1}}{\sqrt{x_n^2 + x_{n-1}^2}}
\end{aligned} \tag{15}$$

A.2 Vectorized code for converting between cartesian and hyperspherical coordinates

This code is accessible here and provided below for reference.

```
import torch

def r (x):
    r = torch.linalg.norm (x, dim=1)
    return r

def cart_to_cos_sph (x, device):
    m = x.size(0)
    n = x.size(1)
    mask = torch.triu(torch.ones(n, n)).to(device)
    mask = torch.unsqueeze(mask, dim=0)
    mask = mask.expand(m, n, n)
    X = torch.unsqueeze(x, dim=1).expand(m, n, n)
    X_squared = torch.square(X)
    X_squared_masked = X_squared * mask
    denom = torch.sqrt(torch.sum(X_squared_masked, dim=2)+0.001)
    cos_phi = x / denom
    return cos_phi[:, 0:n-1]

def cart_to_sin_sph (x, device):
    return torch.sqrt (1 - cart_to_cos_sph (x, device).pow(2))

def cart_to_sph (x, device):
    m = x.size(0)
    n = x.size(1)
    mask = torch.triu(torch.ones(n, n)).to(device)
    mask = torch.unsqueeze(mask, dim=0)
    mask = mask.expand(m, n, n)
    X = torch.unsqueeze(x, dim=1).expand(m, n, n)
    X_squared = torch.square(X)
    X_squared_masked = X_squared * mask
    denom = torch.sqrt(torch.sum(X_squared_masked, dim=2)+0.001)
    phi_plus = torch.arccos (x / denom)
    phi_minus = 2*3.141592654 - phi_plus
    phi = phi_plus
```

```
    phi[:, n-2] = torch.where (x[:, n-1] >= 0, phi_plus[:, n-2],
        phi_minus[:, n-2])

    return phi[:, 0:n-1]

def sph_to_cart (R, phi, device):

    m = phi.size(0)

    n = phi.size(1)+1

    mask = torch.tril(torch.ones(n-1, n-1)).to(device)

    mask = torch.unsqueeze(mask, dim=0)

    mask = mask.expand(m, n-1, n-1)

    PHI = torch.unsqueeze(phi, dim=1).expand(m, n-1, n-1)

    sin_PHI = torch.sin(PHI)

    mask_ = torch.unsqueeze(torch.triu(torch.ones(n-1, n-1),
        diagonal=1).to(device), dim=0).expand(m, n-1, n-1)

    sin_PHI_masked = sin_PHI * mask + mask_

    sin_prod = torch.prod (sin_PHI_masked, dim=2)

    ones = torch.ones(m).to(device)

    sin_PROD = torch.column_stack((ones, sin_prod))

    cos_R = torch.mul (torch.column_stack((torch.cos (phi), ones)),
        torch.unsqueeze(R, dim=1))

    x = torch.mul (sin_PROD, cos_R)

    return x
```

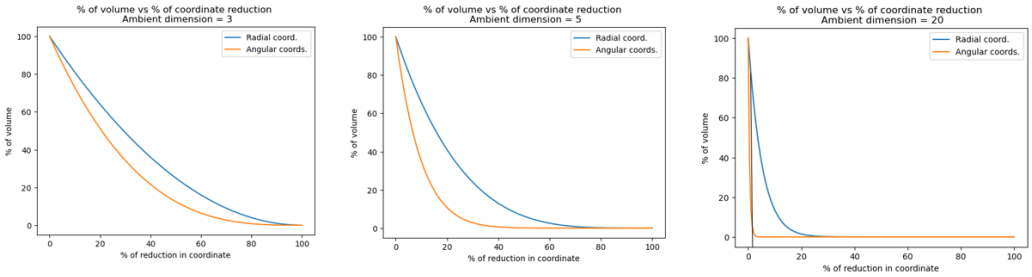
A.3 Hypervolume element in hyperspherical coordinates

The hypervolume element of the hypersphere $\mathbb{S}_R^{n-1}$ is given by the following expression when using hyperspherical coordinates (see https://en.wikipedia.org/wiki/N-sphere):

$$\mathrm{d}V_{\mathbb{S}_R^{n-1}} = R^{n-1}\sin^{n-2}(\varphi_1)\sin^{n-3}(\varphi_2)\cdots\sin(\varphi_{n-2})\mathrm{d}\varphi_1\mathrm{d}\varphi_2\cdots\mathrm{d}\varphi_{n-1} \tag{16}$$

In the small angle regime, where $\sin\varphi \approx \varphi$, we can approximately integrate this expression for an angular coordinate hypercube $[0,\varphi_0]^{n-1}$, and the result is proportional to $v_0 = R^{n-1}\varphi_0^{n(n-1)/2}$. If now we reduce the size of the angular coordinate hypercube by a schedule of the form $\varphi_t = \varphi_0(1-t)$, $t \in [0,1]$, then we can compare the percentage of hypervolume being reduced from the initial value, while keeping R fixed, to the percentage obtained by reducing the size of the hypersphere by an schedule of the form $R_t = R(1-t)$, $t \in [0,1]$, while keeping φ_0 fixed (this second case is equivalent to reducing all the Cartesian coordinates at once, because $r^2 = x_n^2 + x_{n-1}^2 + \ldots + x_2^2 + x_1^2$). Indeed, we get, respectively, $v_t = v_0(1-t)^{n(n-1)/2}$ and $v_t = v_0(1-t)^{n-1}$. In Fig. 3 we plot the behavior of v_t/v_0 in terms of the reduction of the coordinate, given by $(1-t)$, for three, increasing values of dimension n. As we can see, already in dimension 20 (right figure in the panel), there's a sharp decrease in volume in the angular case as soon as one decreases the angular coordinates

by a minimal amount; in comparison to the radial/Cartesian coordinate case, the abrupt decrease in volume looks almost discontinuous.



$$(1-t)^{n-1} \quad vs. \quad (1-t)^{n(n-1)/2}$$

Figure 3: Hypervolume element reduction comparison.

A.4 Concentration of measure effects

In this appendix, we collect the results of simple experiments that clearly show the concentration of measure effects that occur in high dimensions. In Fig. 4a), we show the distribution of a simple Normal distribution in 2 dimensions (left), and the histogram for the norm of the samples (right). In b), the same but for a Normal distribution in 100 dimensions. In Fig. 5a), we show the histogram for the angle between two random samples from a Normal distribution in 2 dimensions (left), and the same but for a Normal distribution in 100 dimensions (right). In b), we display a schematic diagram of the mass concentration of the uniform measure of the hypersphere in very high dimensions. The intuition in this diagram comes from the more precise result (Wainwright, 2019) which states that, for *any* given $y \in \mathbb{R}^n$, if we define on the hypersphere an 'equatorial' slice of width $\epsilon > 0$ as $T_y(\epsilon) \doteq \left\{z \in \mathbb{S}^{n-1} / \mid (z, y) \mid \leq \epsilon/2\right\}$, then its volume according to the uniform measure satisfies the following concentration inequality:

$$\mathbb{P}\left[T_y(\epsilon)\right] \geq 1 - \sqrt{2\pi}\exp(-\frac{n\epsilon^2}{2}). \tag{17}$$

The previous inequality shows that, in very high dimensions, the equatorial slice $T_y(\epsilon)$ occupies a huge portion of the total volume, even for a very small width.

Finally, with this in place, we can understand the peculiar shape that a high dimensional Normal distribution takes when expressed in hyperspherical coordinates (Fig.6).

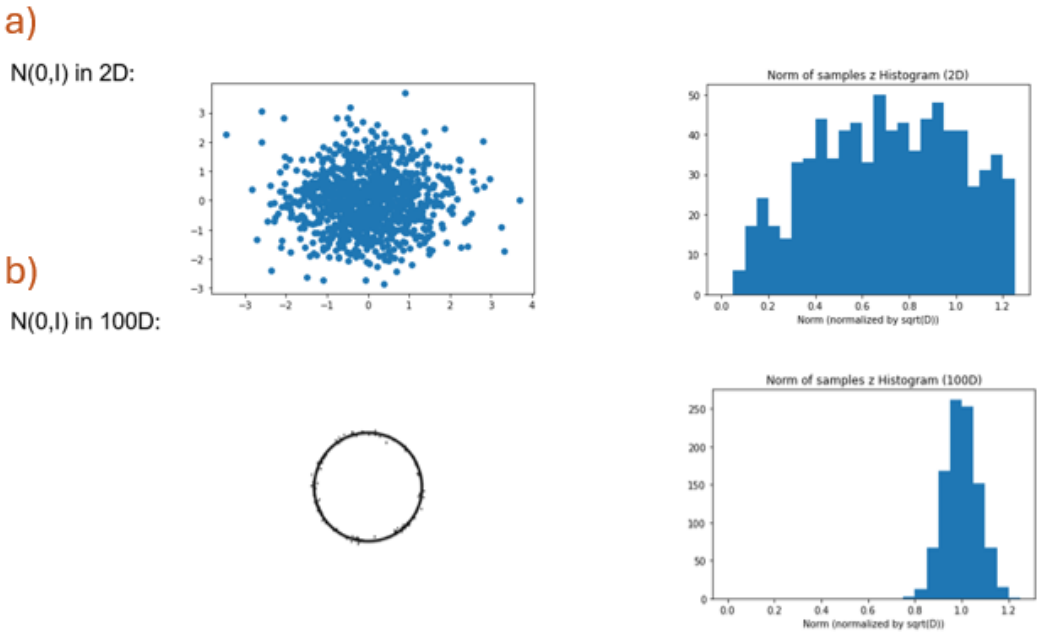


Figure 4: Measure concentration, norm (left image in b), adapted from Vershynin (2018))

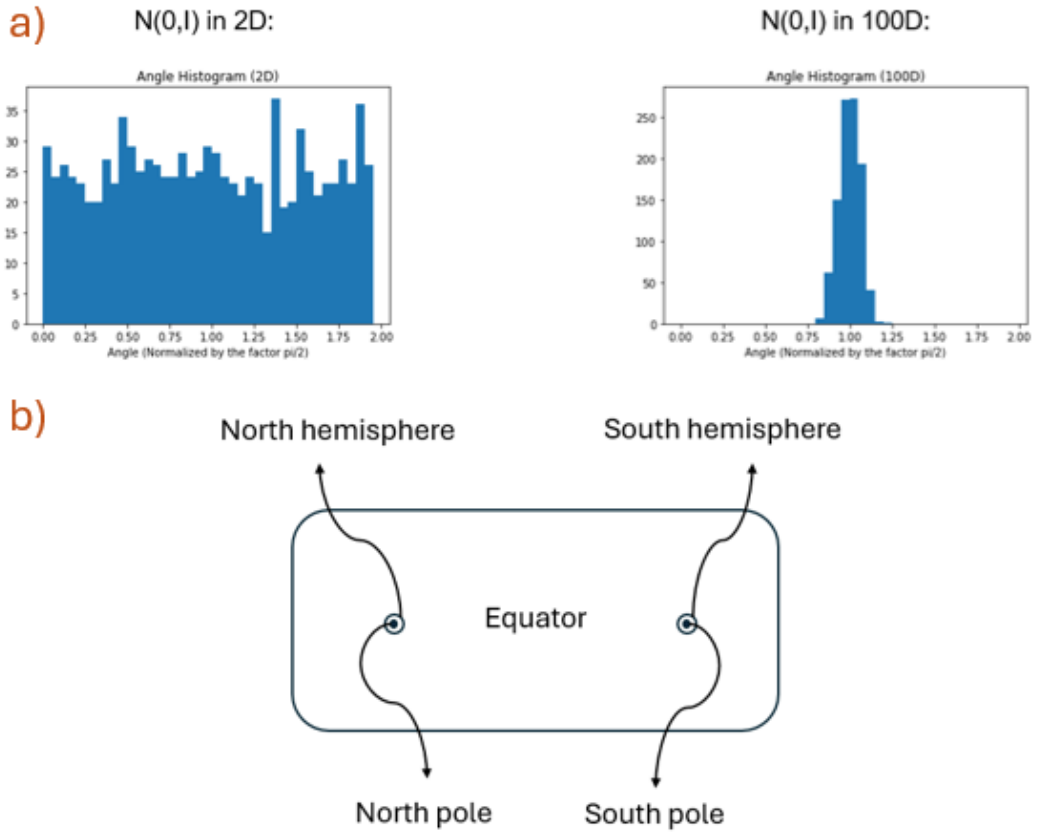


Figure 5: a) Measure concentration, angle; b) Schematic diagram of the mass concentration of the uniform measure of the hypersphere in very high dimensions: most of the volume is in the equator.

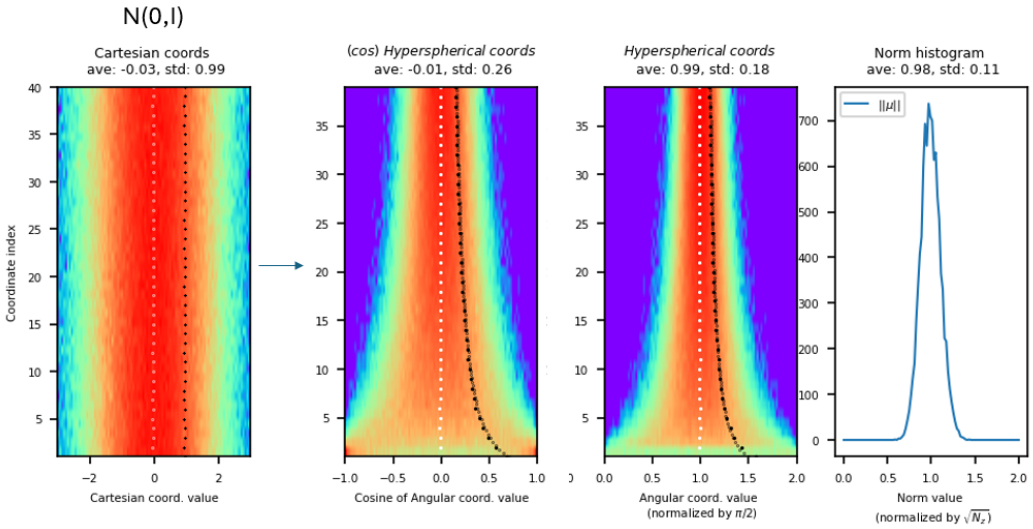


Figure 6: High dimensional Normal distribution in hyperspherical coordinates. For the first three images from the left, each horizontal slice at some vertical index value shows the color coded histogram (red, high density; blue, low density) for the range of the coordinate of that index; the vertical axis stacks all the histograms for all the dimensions (in this example, 40). The white dots represent the mean and the black dots represent the standard deviation of the corresponding histogram. The numbers on top are the total mean and standard deviation of all these previous values taken together.

A.5 ADDITIONAL ANALYSIS OF CIFAR10 RESULTS

Here, we continue the analysis of the experimental results that we obtained for CIFAR10. Fig.7 shows the same results as Fig.2, but we now make a more detailed breakdown of the dependencies of both the MSE and FID wrt both the number of latent space dimensions and the total gain β (as in Fig.2, solid lines correspond to the compression model, while dashed lines to the standard one).

Fig.8 shows the typical training of a standard VAE in one of our experimental rounds. In the upper panel we show, from left to right, the histograms of μ, σ, and z, respectively, using the same conventions as in Fig.6. The fourth histogram in this panel shows the norm histograms of μ and z, as well as the 'replica angle' (dashed red lines) between the testing samples and the mean for all the test set (this value should give an idea about the angular size of the island as well as to signal if there's an overall replica symmetry breaking in our model; in this particular example, there's no such phase transition, since the mean value of the replica angle is close to $\pi/2$). The second, middle panel shows the behaviour of the MSE and KL loses during training for the test set. The bottom panel corresponds to the histogram of the cosine of the hyperspherical coordinates of μ (cf. Fig.6).

Fig.9 shows the typical training of our compression VAE in one of our experimental rounds. The conventions are the same as in Fig.8. Of note is that the replica angle value in this case shows the desired phase transition. The middle panel shows the annealing schedule used for training our model. Finally, we can see how in the histogram of the cosine of the hyperspherical coordinates all of them are shifted towards a cosine value of 1, which corresponds to an angle equal to 0, as expected.

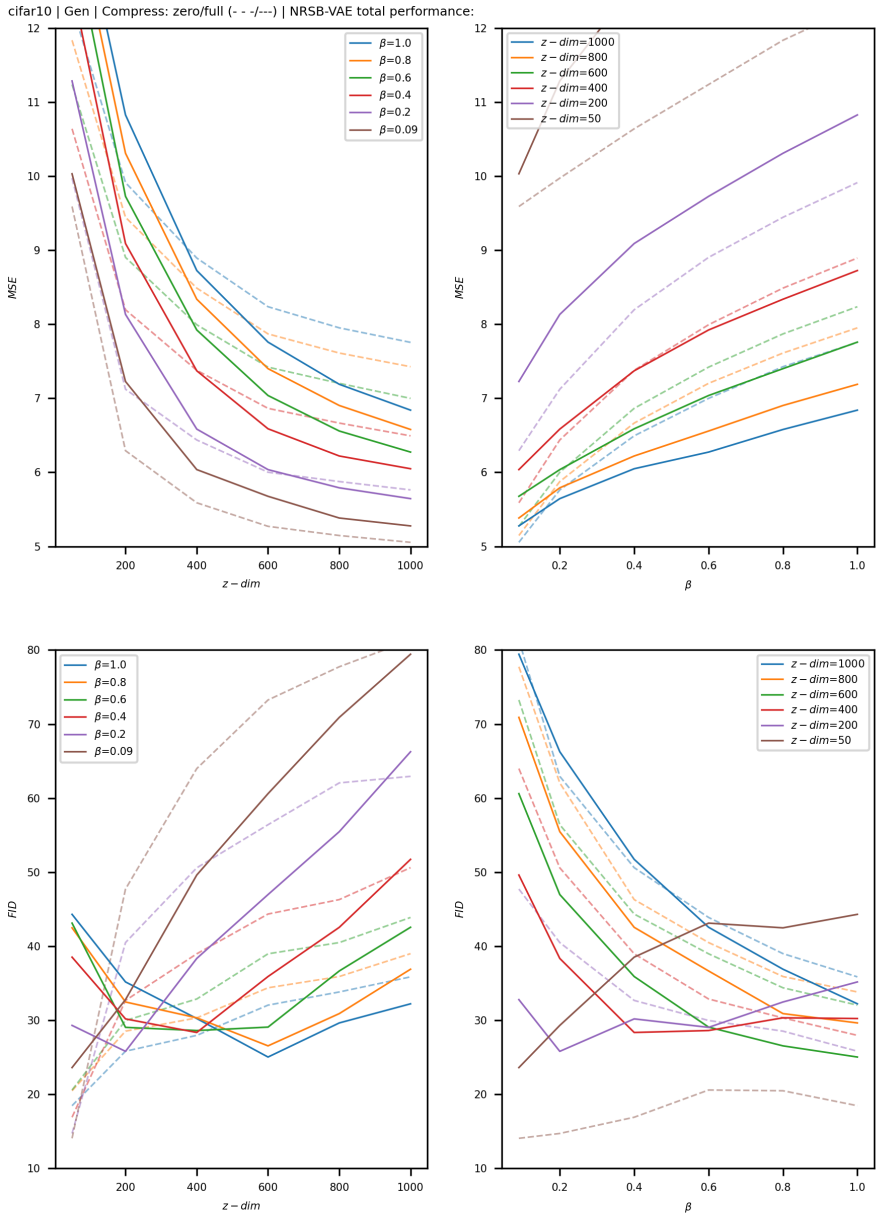


Figure 7: CIFAR10 results breakdown. MSE and FID in terms of both the number of latent space dimensions and the total gain β (cf. Fig.2).

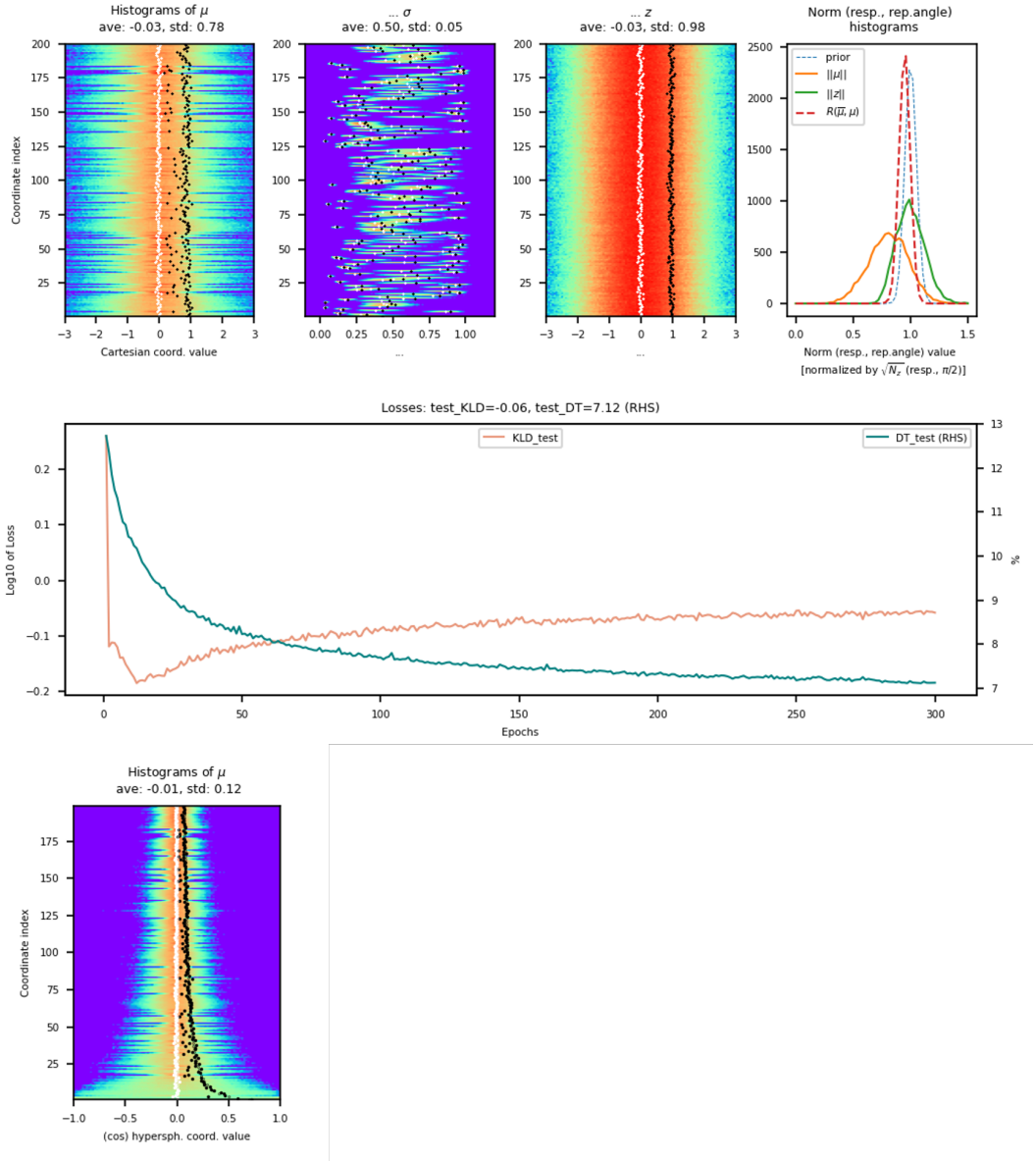


Figure 8: Results of standard VAE training with a balanced β.

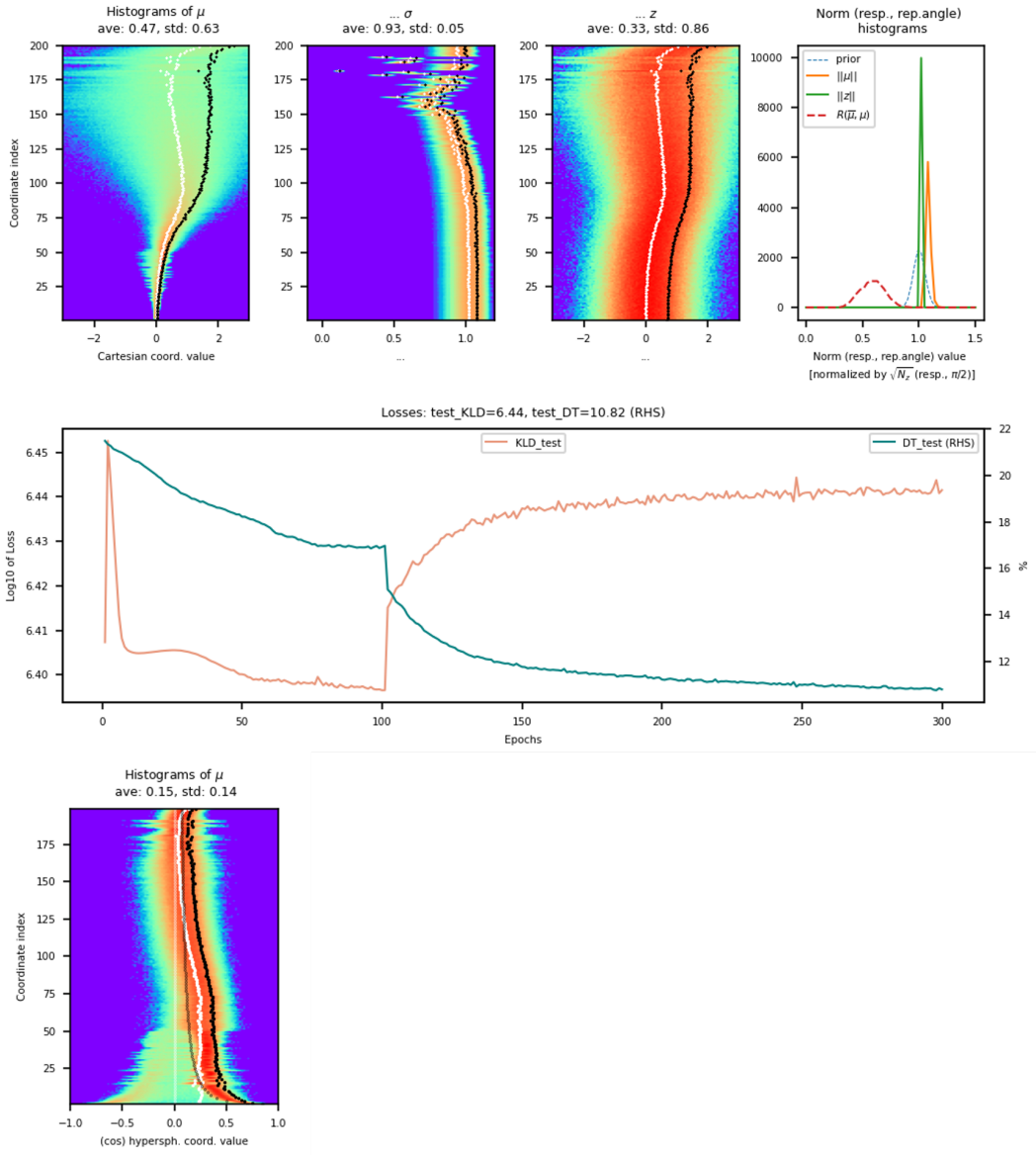


Figure 9: Results of a compressed VAE training.

A.6 THE DIFFERENT REGIMES OF THE STANDARD βVAE IN HD

In this appendix, we illustrate with several examples from our experiments the different regimes in which a βVAE can operate according to the value of the parameter β, while maintaining the dimension of the latent space fixed but high enough. This is important, since it's known (see, e.g., Cinelli et al. (2021), section 5.5.2) that HD VAEs are prone to exhibit a phenomenon known as posterior collapse when β is too high: "[...] [a] state where the variational posterior and true model posterior collapse to the prior, the posterior encodes no information about the input x, and no useful latent representation was learned" (quoted from the mentioned reference). This of course, is a problem, since the collapsed latent dimensions become inoperative for the model and in-utilizable for other tasks. Furthermore, if used, they can introduce errors in those analysis.

In practice, a simple solution to avoid this issue that often works is to simply reduce the value of β, which acts as a gain for the KLD term in the VAE loss function. One can check for any collapse by inspecting the histograms of the means μ of the latent encoding and making sure that the standard deviation (std) there is appreciably away from zero for each latent dimension. A threshold value can be implemented, but we will keep the discussion qualitative in that aspect.

In Fig.10 we show a standard VAE trained with a high β ($= 1.00$) in HD ($n = 200$), it has more than half of its dimensions collapsed yet the FID remains the lowest for the examples (for the standard VAE, that is) in this dimension as we decrease the β (cf. Fig.7, second row, right; this is the case for all the dimensions we checked except the lowest, $n = 50$; see A.12 for this latter case). Thus, posterior collapse here acts as an effective dimensional reduction mechanism for the generation, since the collapse actually improves the FID profile (we believe that what happens here is that the weights of the network corresponding to these dimensions are inactive or close to 0 and, therefore, the decoder simply ignores the dimensions in question). Nevertheless, since many dimensions are ignored, the model's latent space lacks representation capacity, which translates into poor reconstructions ($MSE = 9.92$): the model works similarly to a non-collapsed one with a much more lower latent dimension.

In Fig.11 we show a standard VAE trained with a medium/balanced β ($= 0.20$). In this case, there are more functional dimensions than collapsed or almost collapsed ones. Thus, the model has more representation capacity and this is reflected in a lower reconstruction error ($MSE = 7.12$). Nevertheless, since the decoder now actually operates with a much higher number of dimensions, then the sparsity and high hypervolume of HD spaces becomes an issue, and this is reflected in a worse generative performance (higher FID than the previous case). In Fig.12, we show a standard VAE trained with a low β ($= 0.09$) VAE. In this example, the mentioned trends continue and intensify, now with a much better reconstruction ($MSE = 6.32$), but very poor generation.

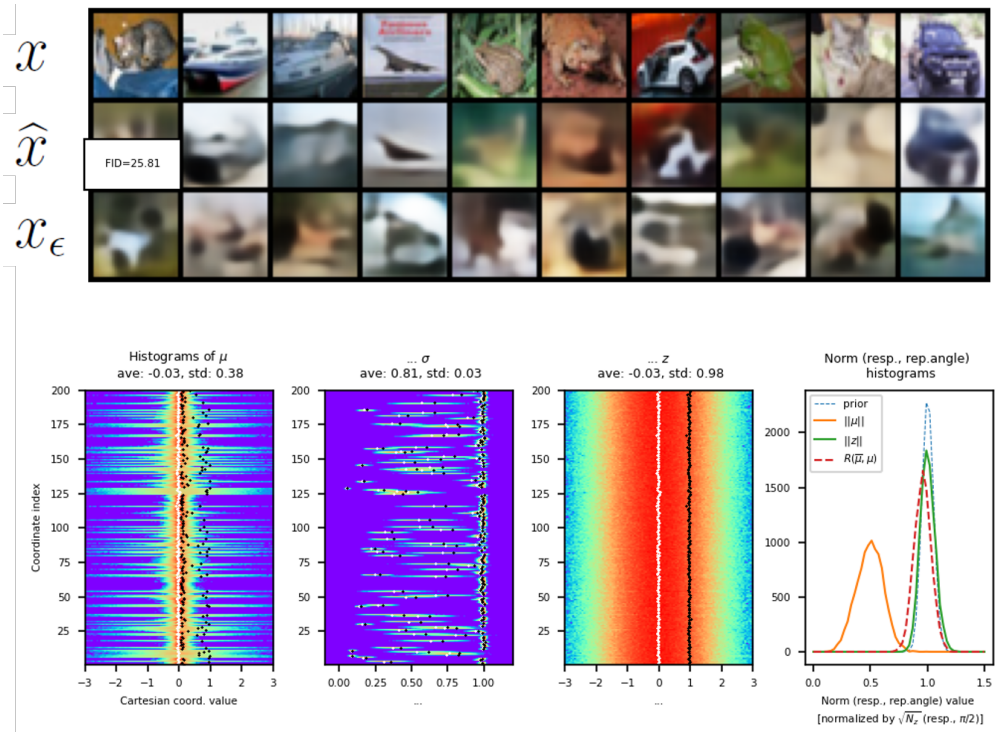


Figure 10: Results of a high β ($= 1.00$) VAE training ($MSE = 9.92$, poor). Notice the collapsed dimensions in the histograms for μ (the variance, black dots, for each of those dimensions is very close to 0). Good generation.

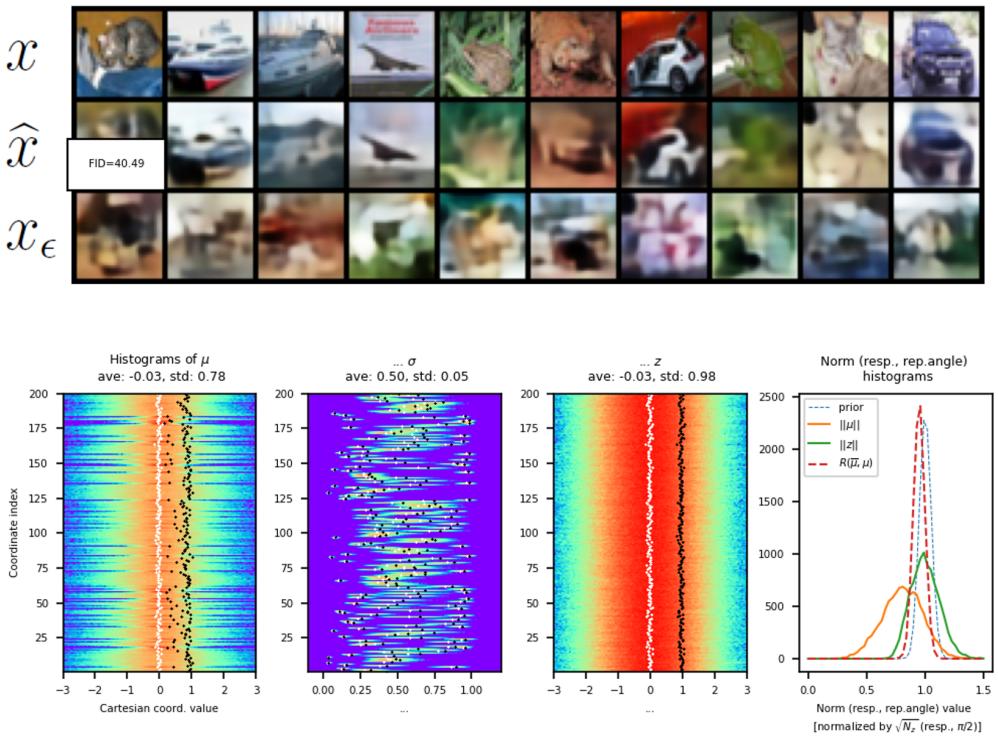


Figure 11: Results of a medium/balanced β ($= 0.20$) VAE training ($MSE = 7.12$, regular). There are more functional dimensions than collapsed or almost collapsed ones. Regular generation.

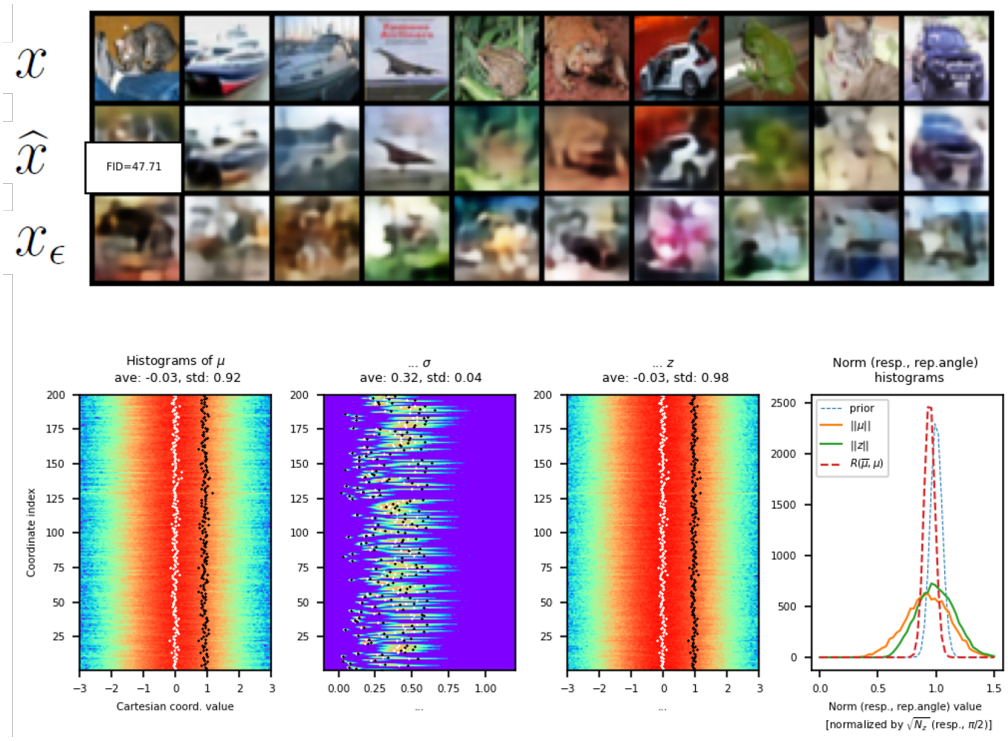


Figure 12: Results of a low β ($= 0.09$) VAE training ($MSE = 6.32$, good). There are no collapsed dimensions, but the model becomes almost an autoencoder (i.e., the VAE's σ is close to 0). Bad generation.

A.7 AVOIDING POSTERIOR COLLAPSE IS NOT ENOUGH TO IMPROVE GENERATION IN A HD VAE

In this appendix, we show an example in which we encourage the mean of the radial coordinate $\overset{\mu}{r}$ of the encoded means μ to lie on the hypersphere of radius $\sqrt{n}$, i.e., $a_{\mu,r} = \sqrt{n}$, and the means of the (cosine) hyperspherical angles $\overset{\mu}{\varphi}_k$ to lie in the equators, i.e., $a_{\mu,k} = 0$, $\forall k$; furthermore, we also balance the variance of the (cosine) angles $\overset{\mu}{\varphi}_k$ by encouraging it to be in the same direction as the vector whose Cartesian coordinates are $(1, \ldots, 1)$, i.e., $b_{\mu,k} = 1/\sqrt{k+1}$, $\forall k$. With this setup, our experiments show that posterior collapse is avoided (in both the Cartesian and hyperspherical coordinates representations), while the distribution of μ is still similar to a uniform distribution on the hypersphere, like in the standard VAE (cf. Bardes et al. (2021)). Nevertheless, as expected from the discussion in the previous section, this is not enough to guarantee good generation (Fig.13).

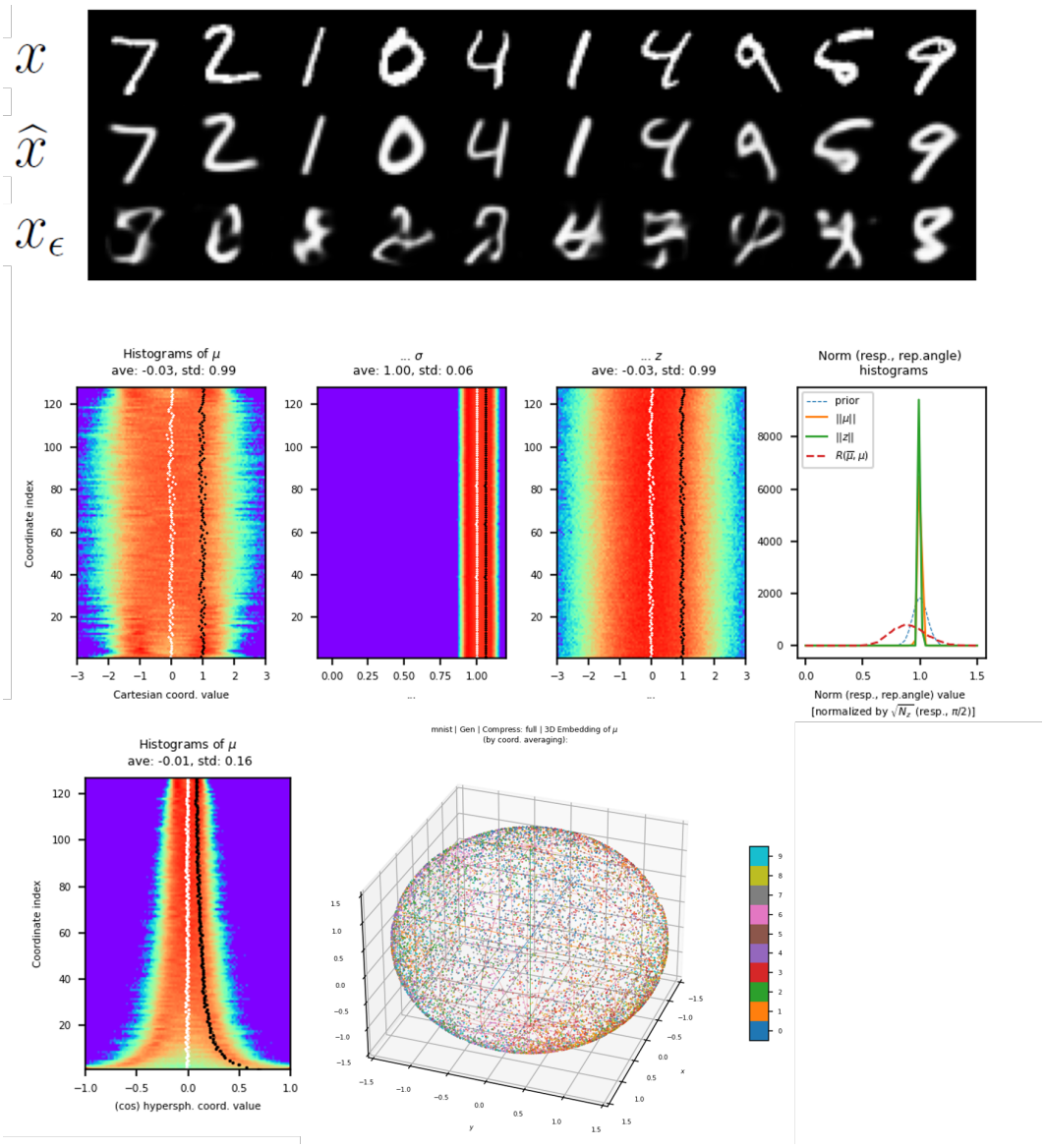


Figure 13: Results of a non-collapsed, non-compressed VAE training. We repeated the experiments for several target values for σ and β, but the results were qualitatively the same as in the present figure.

A.8 High hypervolume compression reduces sparsity and improves generation in HD VAEs

Continuing the analysis of the previous section and figure, then, now that we are sure that we don't have any collapsed latent dimensions and thus are using the full representation capacity of the HD space, we can try to improve the poor generation. If our hypothesis about the sparsity introduced by the exponentially (wrt the dimension) diverging hypervolume in the equators being the root cause of this issue is true, then by implementing our compression via hyperspherical coordinates we should be able to improve this generation while remaining on the hyphersphere, un-collapsed and thus retaining the full expressive capacity of the HD space (unless we compress too much and the excessive overlap hinders the reconstruction).

In Fig.14 we start with a moderate amount of compression by encouraging the mean of the (cosine) angles $\overset{\mu}{\varphi}_k$ to be in the same direction as the vector whose Cartesian coordinates are $(1, \dots, 1)$, i.e., $a_{\mu,k} = 1/\sqrt{k+1}$, $\forall k$. Indeed, recall from appendix A.3 that the closer we get to the north pole, the lower the volume. Nevertheless, this moderate compression is not enough to significantly improve the generation. Thus, in Fig.15 we go to full compression mode by setting $a_{\mu,k} = 1$, $\forall k$, which encourages all the points to converge and condense at the north pole. It's only in this regime of very high compression that we get a significantly appreciably improvement in the generation. Furthermore, we consider this a direct proof of our hypothesis regarding the sparsity of HD spaces and their impact on generation. In Fig.2 of the main text we showed our experimental results for the more challenging dataset CIFAR10 regarding how we can use this to systematically take advantage of the better representation capacity of un-collapsed HD latent spaces to maintain a good and stable reconstruction, while we use our method of volume compression to improve at the same time the quality of the generation. This allowed us to reach more valuable zones of the MSE-FID plane which are not accessible via the standard VAE in any combination of the parameters n (latent dimension) and β.

As an additional comment, by looking at the histogram for μ in Cartesian coordinates in Fig.15, one may think that the lower (in coordinate index) latent dimensions seem heavily collapsed. But this is not the case: the latent data distribution lies exactly on the hypersphere, and this forces correlations in the Cartesian coordinates, reason by which the fact that one or many more Cartesian coordinates (and their variance) are close to 0 is not conclusive of the irrelevance of many of the latent dimensions; indeed, if we now check the histogram for the (cosine) angles $\overset{\mu}{\varphi}_k$ in hyperspherical coordinates (which are a set of uncorrelated coordinates on the hypersphere, by construction), then we see that there's no collapse in any dimension there. Adding to this point, we can see in Fig.8 that, in the standard VAE, the collapse in Cartesian coordinates (e.g., around index 20 in the first histogram to the left in the first row) translates into a collapse in the (cos) hyperspherical coordinates (third row histogram, same index), while this is not the case in our compression VAE in Fig.9, where the apparent collapse in Cartesian coordinates around, e.g., index 20, doesn't translate into an analogous collapse in the (cos) hyperspherical coordinates: we believe that the reason for this is that, in the standard VAE, we are not exactly on the hypersphere (in the fourth histogram to the right in the first row in Fig.8, we can see that the norm of μ, in orange, has a non-zero variance, since the prior is still a multivariate Gaussian, not exactly a uniform distribution on the hypersphere), while our compression VAE is indeed exactly on the hypersphere (analogous norm histogram in Fig.9), since we explicitly encourage the variance of the radial coordinate of μ to be 0. Thus, we emphasize that the improvements in generation by our compression method cannot be explained by selective posterior collapse (as in Fig.10), where the HD collapsed latent representation is effectively equivalent to a non-collapsed one in lower dimensions, since this comes at the cost of loosing reconstruction quality; but our method is able to improve generation while retaining some amount of better reconstruction, and this is why some of the best performative examples in Fig.2 cannot be re-obtained by a standard VAE with a different combination of parameters n and β (possibly in a selective collapsed mode). The improvement in our method is coming from the reduction of the sparsity by compression of the latent hypervolume and by performing this in a key angular way due to the peculiar equatorial nature of the volume in HD spaces.

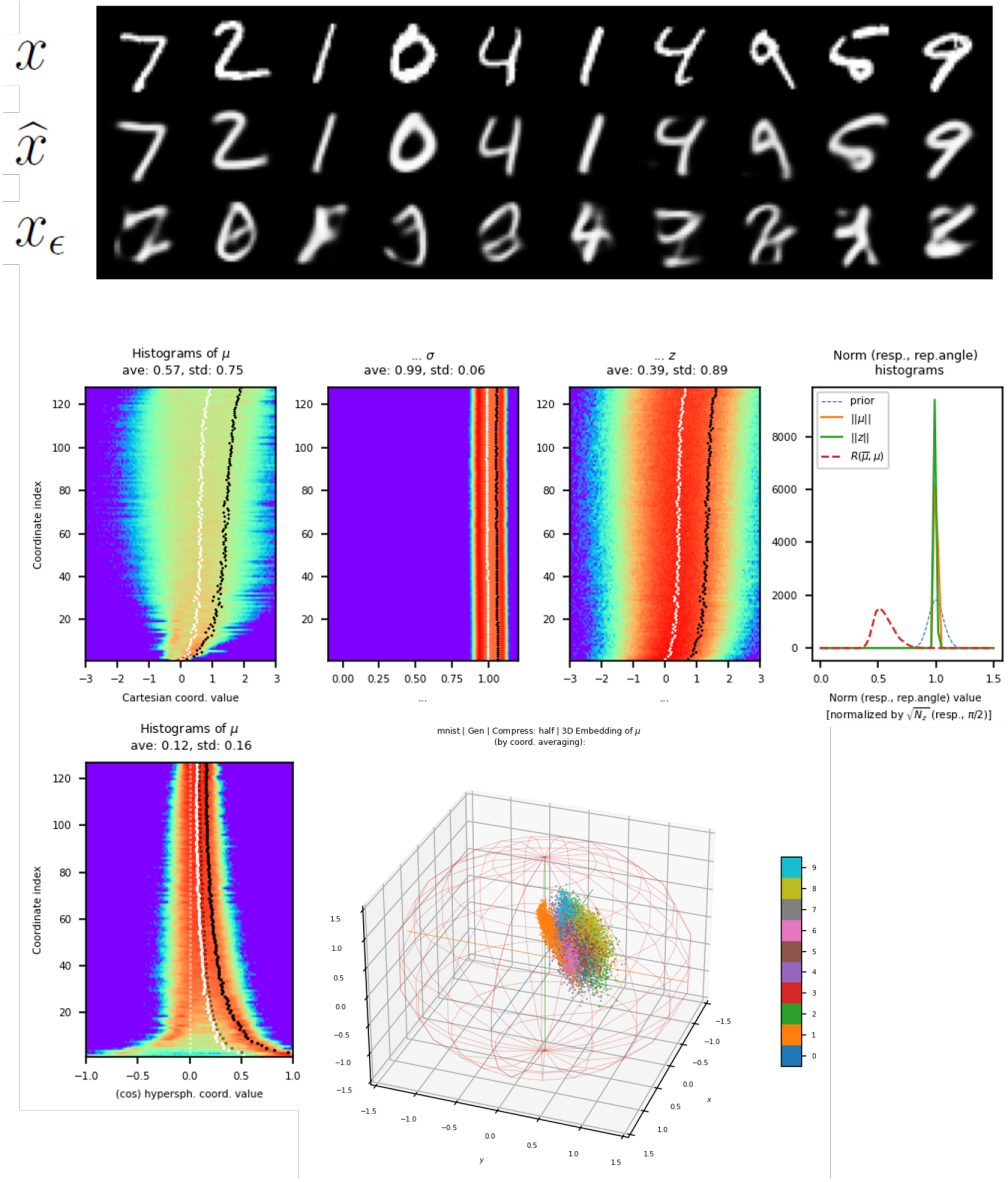


Figure 14: Results of a moderately compressed VAE training.

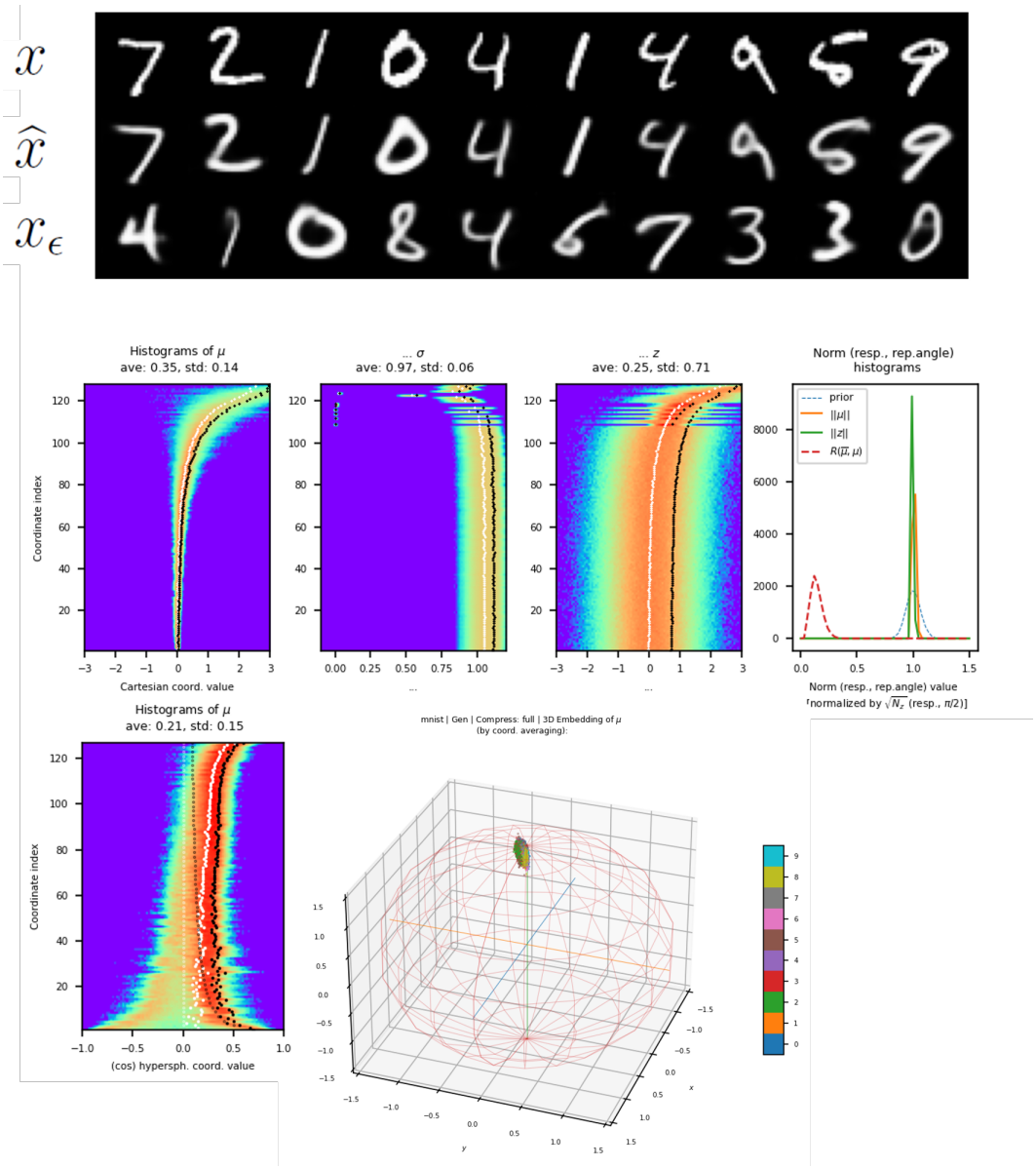


Figure 15: Results of a fully compressed VAE training.

A.9 THE SPIN GLASS ANALOGY DURING TRAINING

We described in Section 4.1 of the main text the following training schedule and the reasons behind this choice: "*[...] we use an annealing schedule for the gain β of the KL-like loss, consisting of an initial stage which increases proportionally with $\sqrt{epoch}$ for the first 100 epochs, and is constant afterwards. This was necessary because we observed that too much compression of the volume was detrimental to the performance, while a strong compression was still necessary at the initial stage[...]*". The gain β here has the role[2] of the inverse temperature, $\beta = 1/T$. In spin glasses and complex systems, the energy function has exponentially many local minima in the equatorial region of the hypersphere. To overcome them, a very strong signal or bias towards the desired region is necessary at the beginning, together with a rapid cooling or quenching. Thus, our initial high β (i.e., very low temperature T) setting, and in the presence of the high intensity (regulated by the β^{-1} factor in front of the MSE) hyperspherical external magnetic fields as bias in directions away from the equator, should make the gradient descent dynamics to quickly tend towards a low temperature distribution with replica symmetry breaking. Indeed, this is what we observed in our experiments, since we check for the replica angle, as mentioned before. This initial strong compression helps escaping those undesirable equatorial minima (Fig.16). Nevertheless, the obtained state shows too much overlapping between samples, so we then perform the annealing (i.e., lower the β, or increase the temperature T, and also lower the intensity of the magnetic fields) in order to allow the system to relax the strong order introduced by the initial bias and, in this way, transition to a replica symmetry breaking state with a bigger angle between replicas (that is, to go back up a bit in the ultrametricity tree/hierarchy of the replica angle values; cf. Mourrat (2024)). This decreases the MSE and makes the decoded images more sharp, at the cost of some generation quality (Fig.17). Note how the replica angle (red dashed lines in fourth histogram to the left in second row) doesn't fully go back to $\pi/2$, even when the KLD term (where the external magnetic fields are) stops optimizing at this stage of the training process (red line in third row), but instead jumps to a different value, higher than the initial one but still below $\pi/2$. This is fully consistent with the spin glass analogy in a quenched and then annealed system, where the glass, always in the replica symmetry breaking phase, jumps from one so-called 'pure state' to a different pure state, i.e., goes back up a bit in the ultrametricity tree/hierarchy of the replica angle values, as mentioned before. But the system has escaped the zone with exponentially many local minima in the equator.

[2] $\mathcal{L} = \beta\left(\beta^{-1}\text{MSE}(x, x_z) + \text{KLD}^{w/Prior}_{\text{HSphCoords}}(\varphi_k, r)\right) = \beta\mathcal{H}$. cf. footnote 1, where $\nabla\mathcal{L} = \beta\nabla\mathcal{H}$ for the gradient descent dynamics on $\mathcal{L}$.

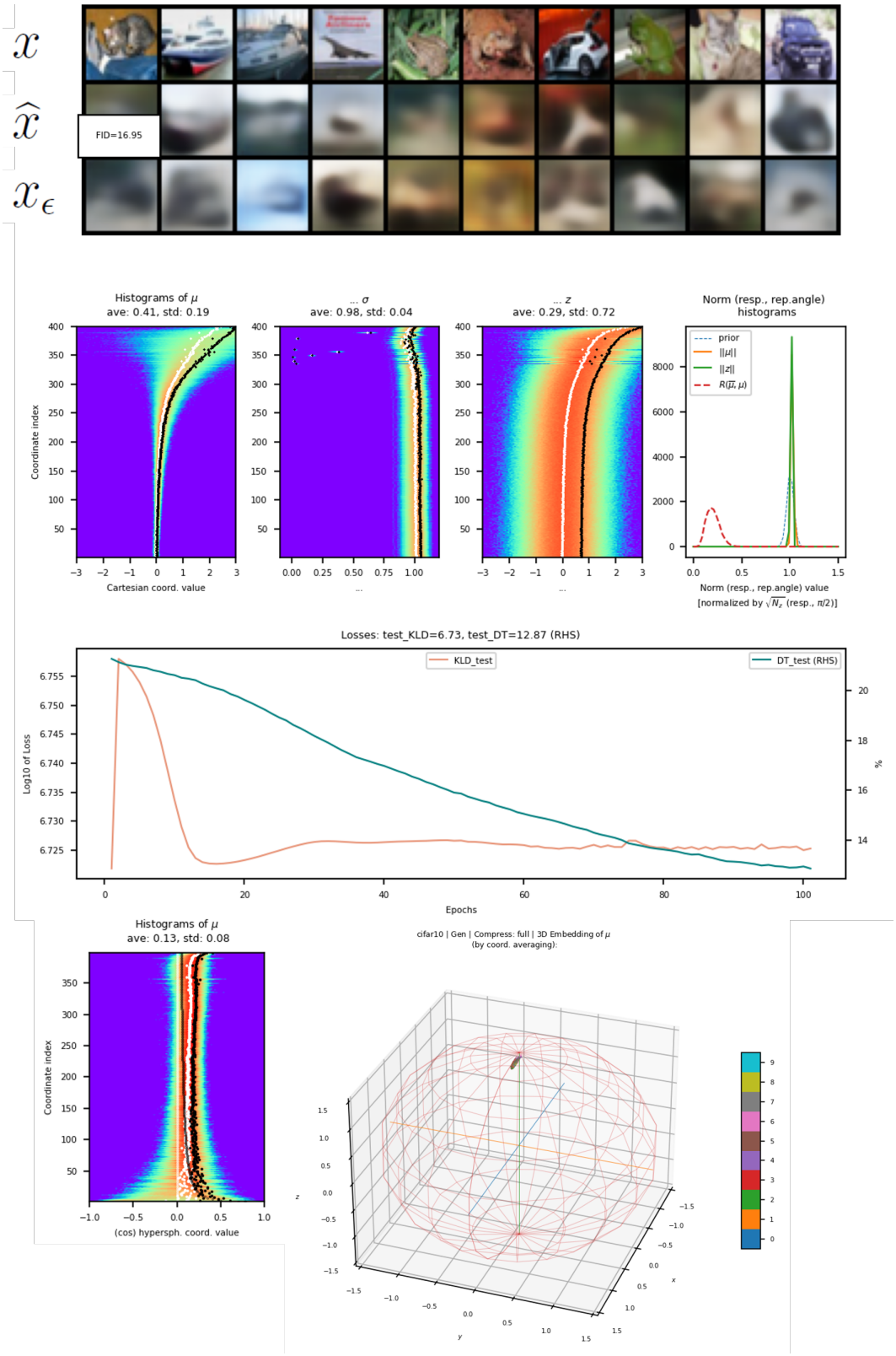


Figure 16: Results of a typical fully compressed VAE training at epoch 100.

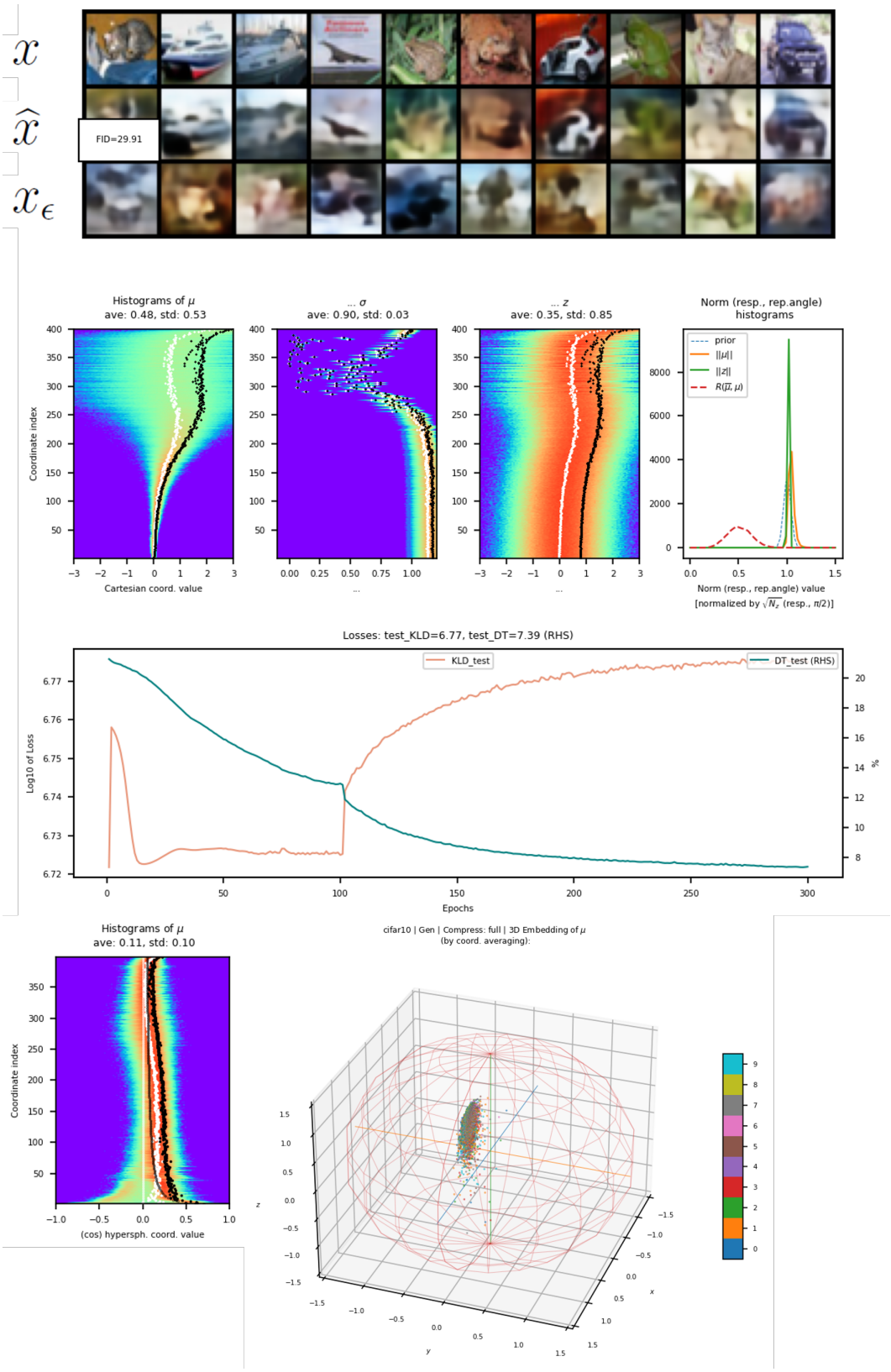


Figure 17: Results of the same fully compressed VAE training at final epoch 300.

A.10 RESULTS ON CELEBA64

In this appendix we include additional experimental results conducted on the dataset CelebA (Liu et al., 2015), resized to a 64×64 image size.

The analysis is of the same type as the one we performed on CIFAR10 (cf. Figs.2, 7), and the results show qualitatively the same trends (Figs.19, 20).

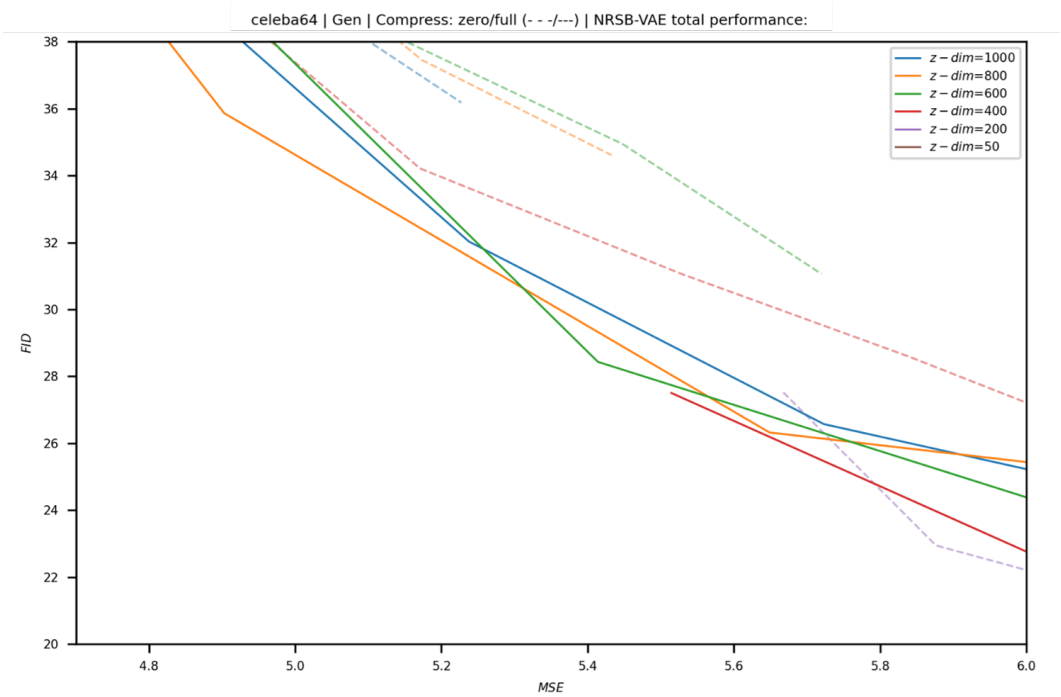


Figure 18: Effect of latent dimension and β on the trade off between reconstruction and generation on CelebA64 (as in CIFAR10, solid lines closer to the bottom left corner than the dashed lines).

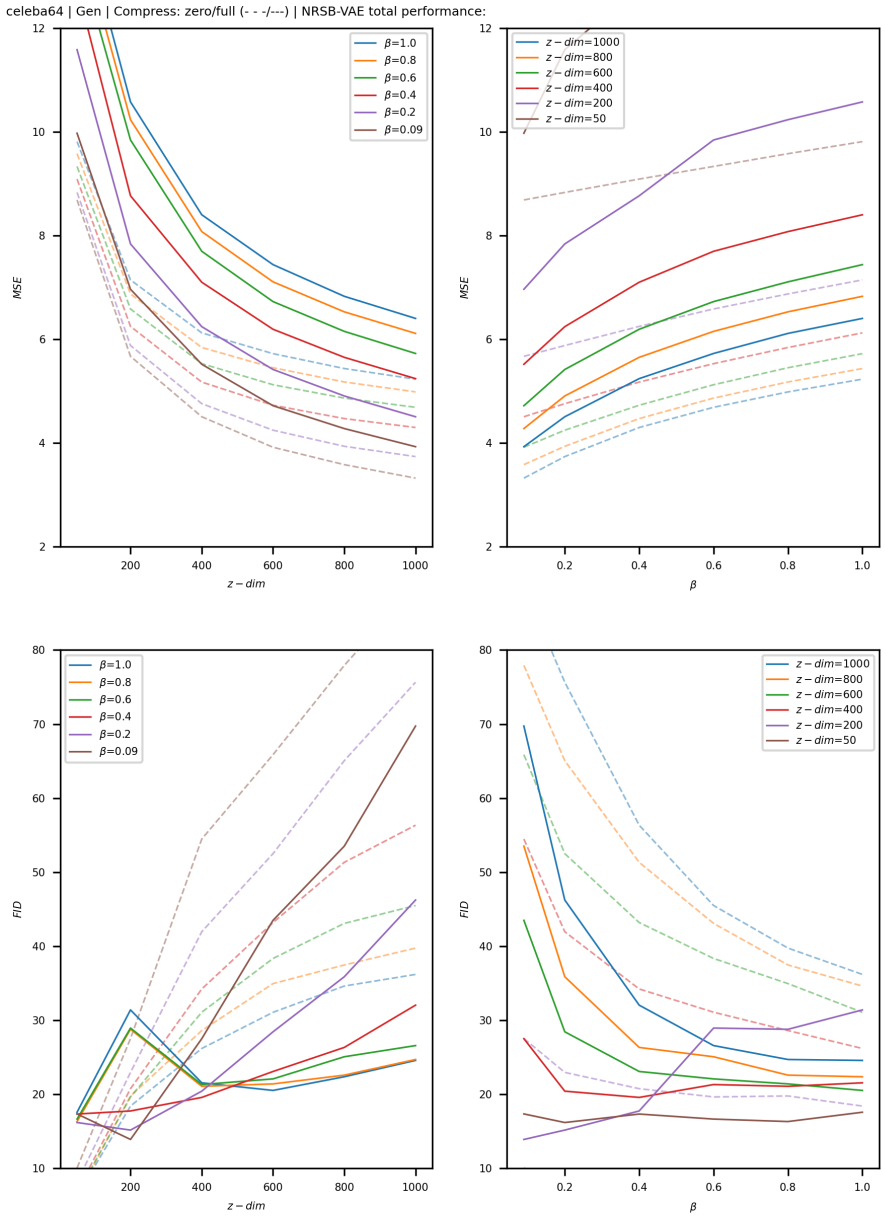


Figure 19: CelebA64 results breakdown. MSE and FID in terms of both the number of latent space dimensions and the total gain β.

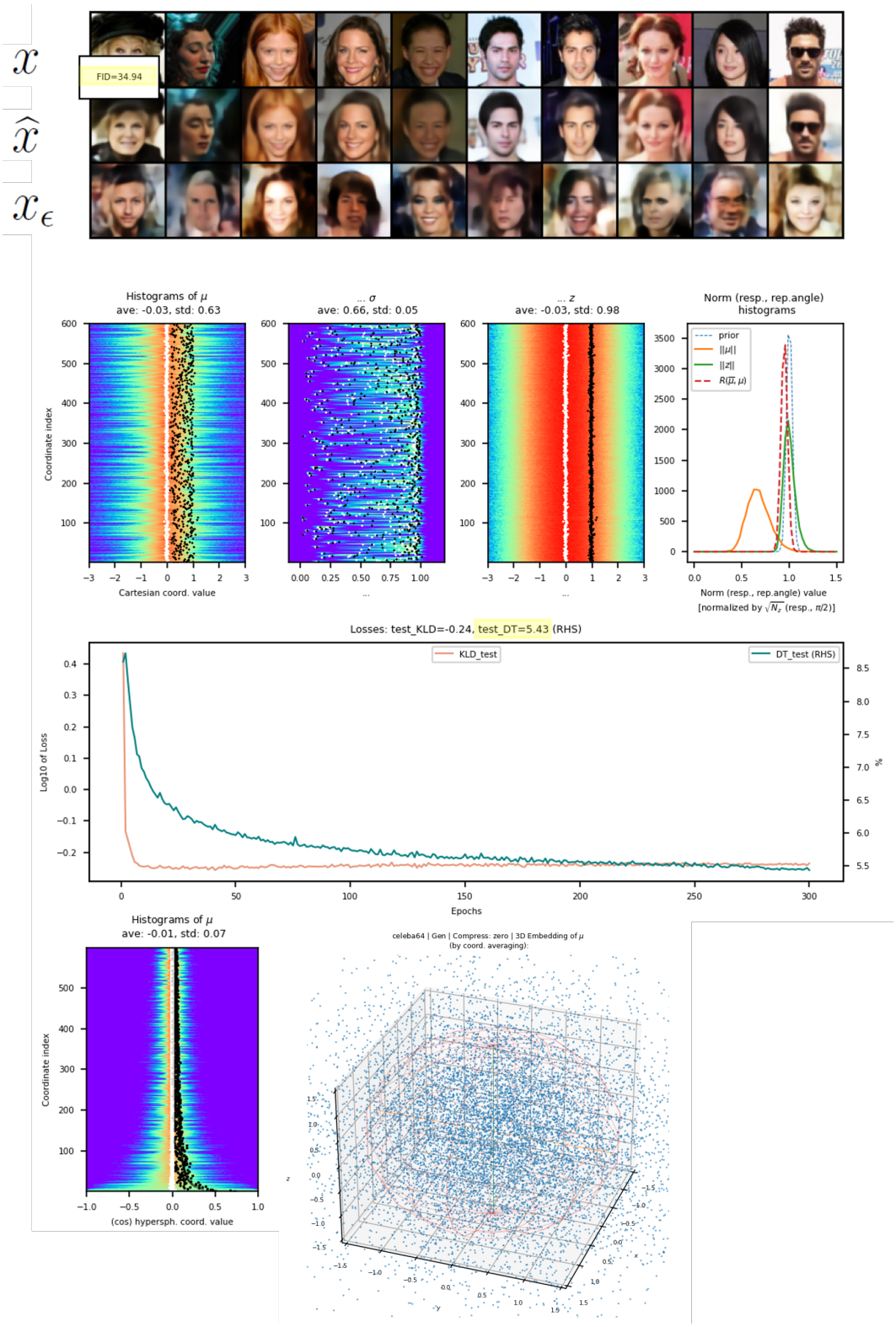


Figure 20: Results of standard VAE training with a balanced β (in the 3-D embedding diagram, the samples are normalized by the overall mean of the radial coordinate, rather than set exactly to the sphere; thus, rather than looking like a uniform-like distribution on the 2-D sphere, it looks like a normal distribution in 3-D, but this difference is only merely in the convention being used regarding the radial normalization for the 3-D embedding). $MSE = 5.43$ and $FID = 34.94$, $n = 600$.

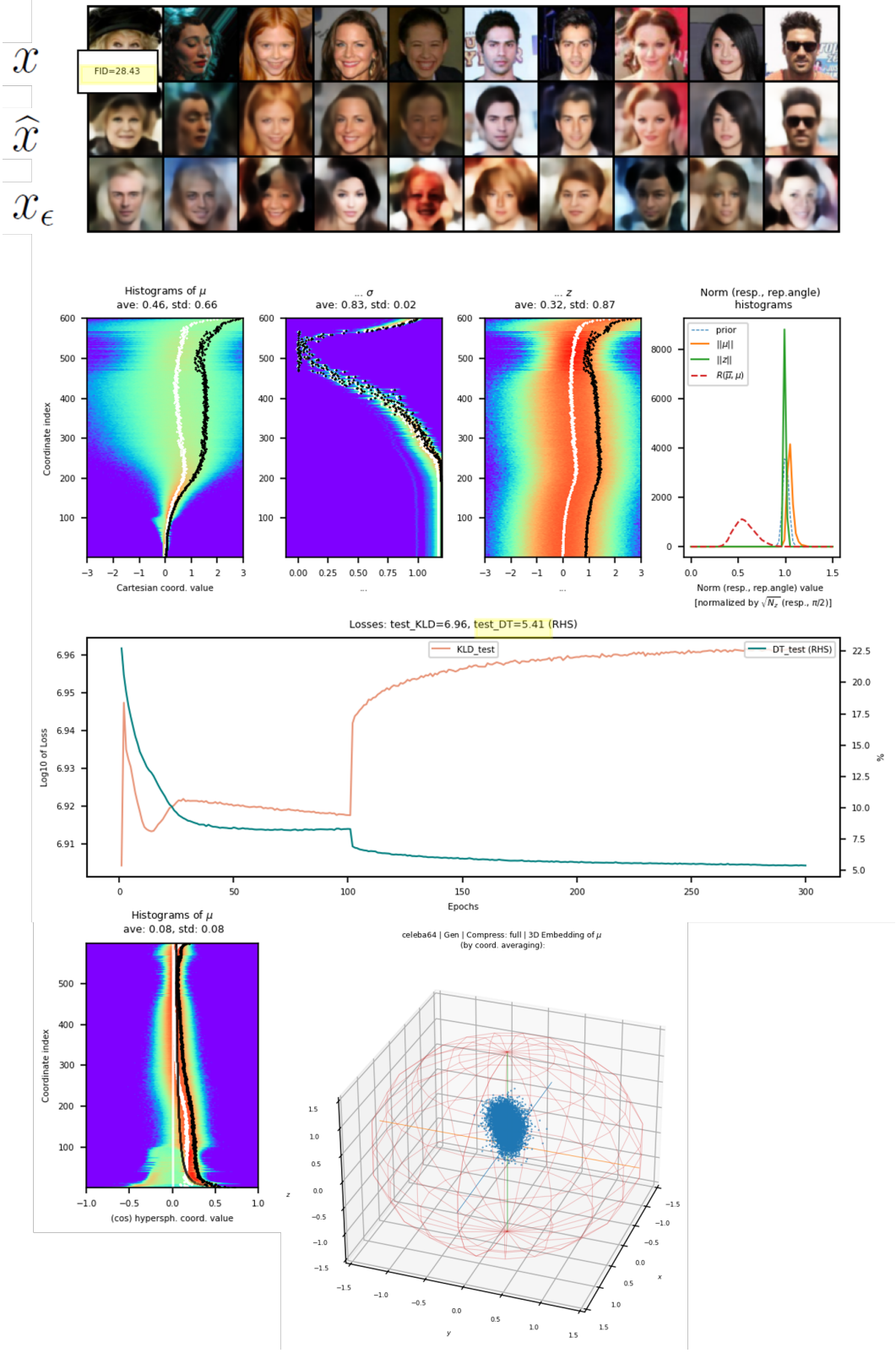


Figure 21: Results of a compressed VAE training. $MSE = 5.41$ and $FID = 28.43$, $n = 600$.

A.11 INTERPOLATIONS ON MNIST

Here we complement the results and associated claims of Fig.1 with interpolations experiments on the same models. They highlight the lack of continuity in the standard VAE case (Fig.22), while they show the gained continuity and how densely packed the clusters are in our compressed version (Fig.23).

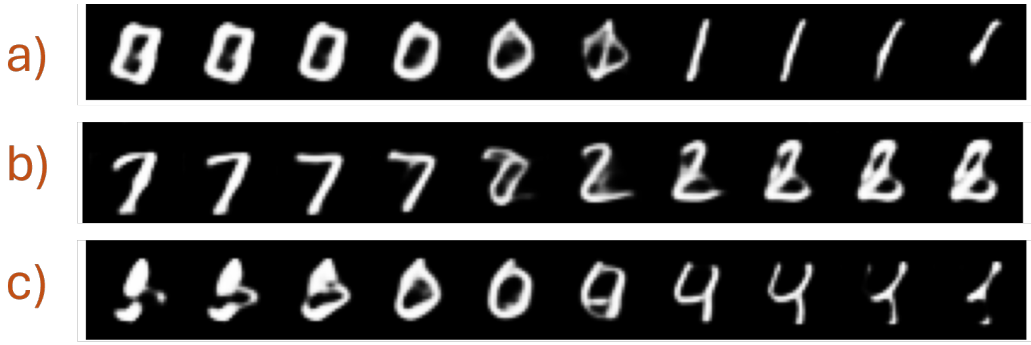


Figure 22: Interpolations on the standard VAE. a) from 0 to 1; b) from 7 to 2; c) from 0 to 4.

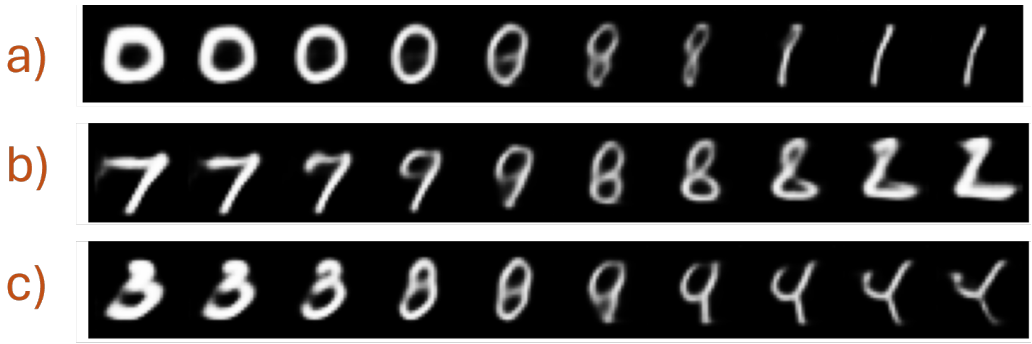


Figure 23: Interpolations on the compression VAE. a)-b) Idem as previous figure.

A.12 THE DIFFERENT REGIMES OF THE STANDARD βVAE IN LOW DIMENSIONS

In this appendix, we perform a similar analysis as the one in A.6, but now for the model with the lowest latent dimension ($n = 50$).

In this situation, the trends actually reverse: de-collapsing the model (that is, going from Figs.24 to 25 and so on) improves the generation as measured by the FID (cf. Figs.10 to 11 and so on, in the HD case, where it becomes worse). See also Fig.7, second row, right.

Nevertheless, these cases are pathological and not very useful, since all of them have very high MSE, that is, the images are too 'blurry'. Thus, both the collapsed and the non-collapsed cases fall into the bottom far right of Fig.2, way outside the more useful area of the MSE-FID plane.

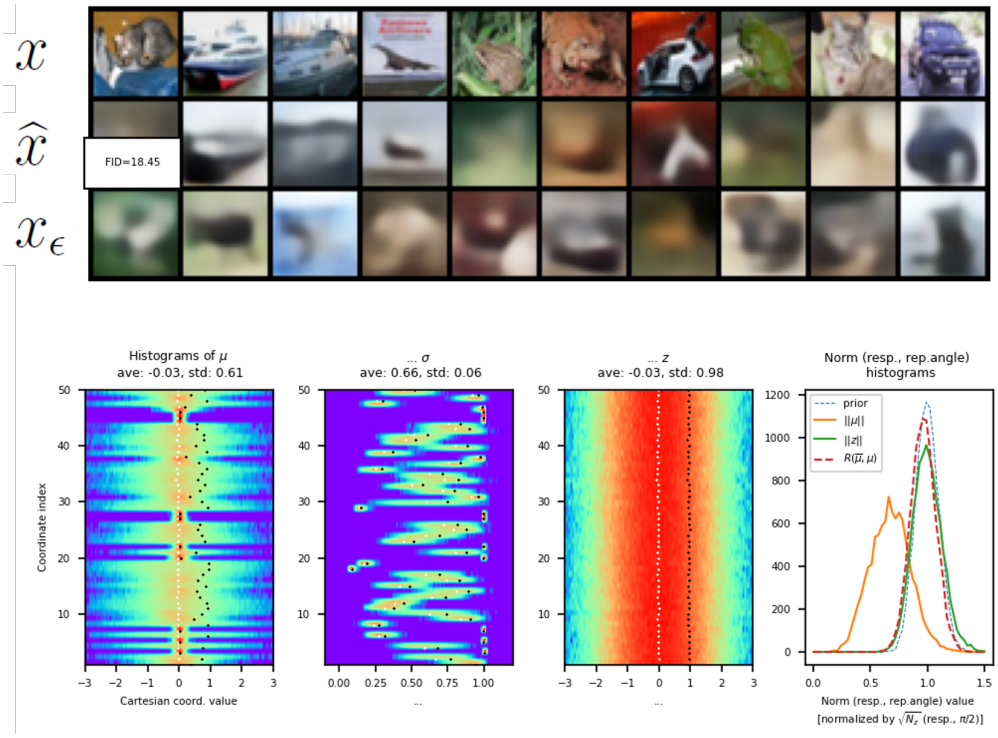


Figure 24: Results of a high β ($= 1.00$) VAE training ($MSE = 12.27$, very poor). Notice the collapsed dimensions in the histograms for μ (the variance, black dots, for each of those dimensions is very close to 0). Good generation.

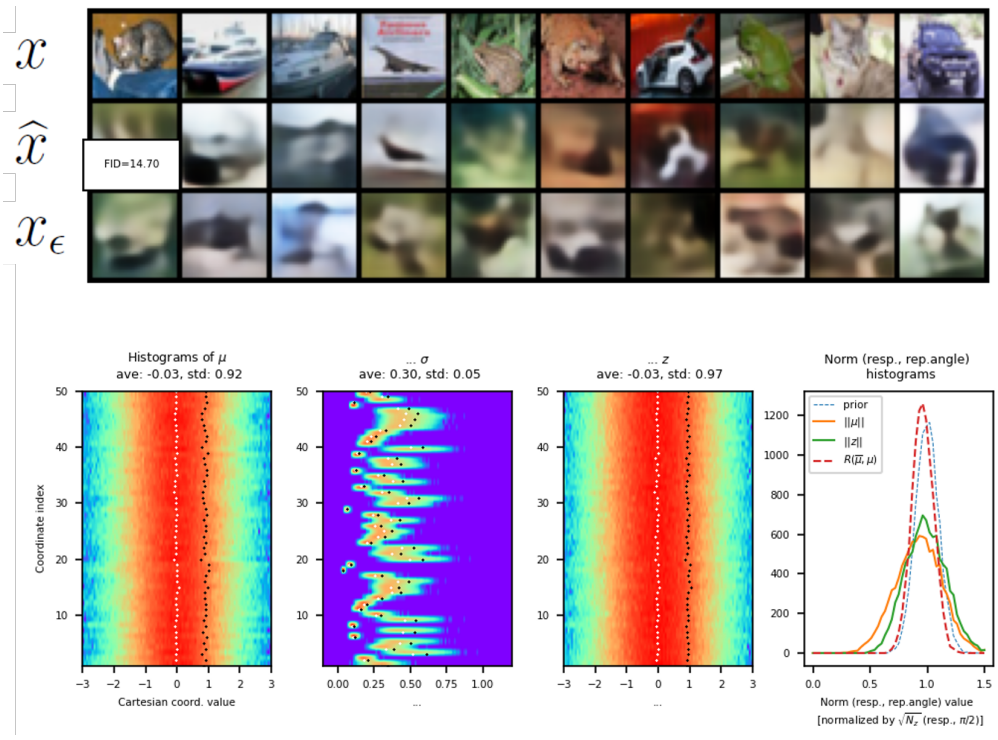


Figure 25: Results of a medium/balanced β ($= 0.20$) VAE training ($MSE = 9.97$, poor). There are more functional dimensions than collapsed or almost collapsed ones. Better generation than previous figure.

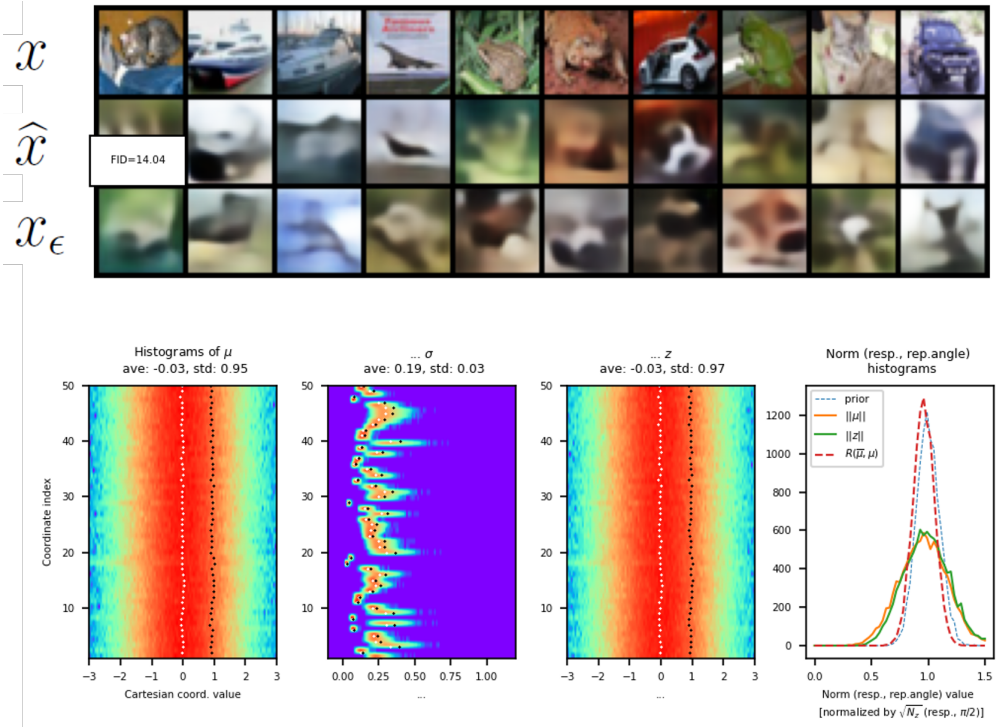


Figure 26: Results of a low β ($= 0.09$) VAE training ($MSE = 9.59$, poor). There are no collapsed dimensions, but the model becomes almost an autoencoder (i.e., the VAE's σ is close to 0). Even better generation than previous figure.