Variational Autoencoders: A Harmonic Perspective

Alexander Camuto
University of Oxford

Matthew Willetts
UCL & The Alan Turing Institute

Abstract

In this work we study Variational Autoencoders (VAEs) from the perspective of harmonic analysis. By viewing a VAE's latent space as a Gaussian Space, a variety of measure space, we derive a series of results that show that the encoder variance of a VAE controls the frequency content of the functions parameterised by the VAE encoder and decoder neural networks. In particular we demonstrate that larger encoder variances reduce the high frequency content of these functions. Our analysis allows us to show that increasing this variance effectively induces a soft Lipschitz constraint on the decoder network of a VAE, which is a core contributor to the adversarial robustness of VAEs. We further demonstrate that adding Gaussian noise to the input of a VAE allows us to more finely control the frequency content and the Lipschitz constant of the VAE encoder networks. Finally, we show that the KL term of the VAE loss serves as single point of action for modulating the frequency content of both encoder and decoder networks; whereby upweighting this term decreases the high-frequency content of both networks. To support our theoretical analysis we run experiments using VAEs with small fully-connected neural networks and with larger convolutional networks, demonstrating empirically that our theory holds for a variety of neural network architectures.

1 Introduction

Variational autoencoders (VAEs) are deep latent variable models that typically use Gaussian priors and Gaussian posteriors in their latent spaces (Rezende et al., 2014; Kingma and Welling, 2014). VAEs have become a work-horse method in modern machine learning, but still their theoretical properties are not fully understood. In particular, we do not yet fully understand the regularising effect of latent space sampling during training.

While the effect of the latent space sampling (a.k.a latent noise) on VAEs has been studied from an information-theoretic standpoint (Shu et al., 2018) and through Taylor analysis (Kumar and Poole, 2020), here we take a different tack and study the impact that latent Gaussian samples have on the *harmonic properties* of the underlying neural networks used to implement the model.

Related to our inquiry is the study of the *Gaussian noise injections* in neural networks trained on supervised tasks (Yin et al., 2019; Camuto et al., 2020, 2021a). Adding (standard) Gaussian noise to the input layer has long been known to induce regularisation (Webb, 1994; Bishop, 1995; Burger and Neubauer, 2003), and recently has been used to induce robustness to adversarial attacks (Cohen et al., 2019). Recently, by studying the effects of these Gaussian noise injections from a functional analysis perspective, (Camuto et al., 2020) shows that these injections penalise functions which learn high frequency components in Fourier-space. These ideas form the starting point for our work.

By considering the latent space of a VAE as a measure space, equipped with a *Gaussian measure*, we can consider the decoder of the VAE as being a member of a *Gaussian Space*, a type of L_2 function space equipped with the Gaussian measure. In our analysis, we view the latent variable's posterior, a Gaussian, as being the Gaussian measure with which the latent space is equipped on a per-datapoint basis. This posterior is located around a particular *location*, the mean, with a particular *scale*, the (often-diagonal) standard deviation.

Previous analysis in Gaussian spaces has been done for spaces equipped with the standard Gaussian mea-

Proceedings of the 25$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

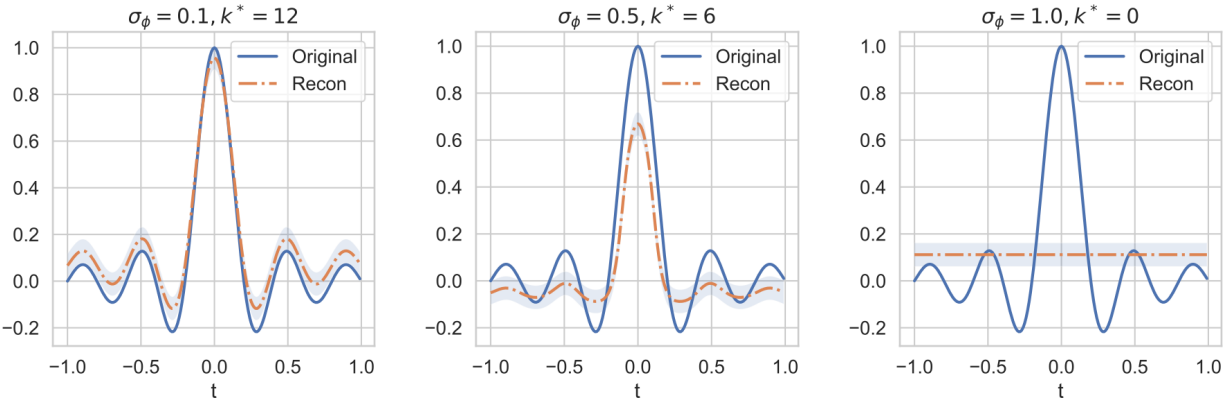


Figure 1: We plot the reconstructed function of a VAE, with the Sigmoid activation function networks (3 dense layer each with 256 units), trained on the function $\text{sinc}(5t)$, $t \in [-1, 1]$. We use a one dimensional latent space $\mathcal{Z}$ with a fixed encoder variance $\sigma_\phi \in [0.1, 0.5, 1.0]$. As σ_ϕ increases we can see qualitatively that the frequency of the reconstructed function increasingly skews towards lower frequencies. We fit polynomial regression to the reconstructed function, picking the optimal polynomial degree k^* via cross-validation on 10 randomly chosen train-test splits. As σ_ϕ increases, the lower frequency content of the decoder function corresponds, as one might expect, to a lower optimal polynomial degree k^*.

sure (zero mean and unit variance). To directly apply the theory of Gaussian spaces to VAEs, we extend the theory of these spaces for standard Gaussian measures to be applicable to *general* Gaussian spaces with arbitrary mean and covariance.

Using this newly developed theory we consider a basis-expansion of the decoder of a VAE in terms of the eigenfunctions of a general Gaussian space, the *Hermite polynomials*. This then enables us to see how higher degrees of these basis functions, these polynomials, are implicitly more-strongly penalised as the variance of the underlying Gaussian space increases, leading the decoder of VAE to preferentially learn functions that can be represented as lower degree polynomials. As higher-order polynomials are naturally associated with higher frequency content in the Fourier domain, this means that a higher variance Gaussian measure in the latent space necessarily leads to decoders with lower-frequency functional representations.

Further, the encoder, which in our analysis parameterises the Gaussian measure of the latent space, is also affected by changes in the latent measure's variance. By studying the Fourier transform of Gaussian measures we can show very simply that larger-variance measures have less high frequency content in the Fourier domain. This implies that encoders that learn high-variance posteriors have lower-frequency representations.

To modulate the frequency content of the *individual networks* that constitute the VAE encoder, we study the effect of adding Gaussian noise to VAE inputs during training. This noising operation effectively turns the input space into a Gaussian space on a per-datapoint basis. In the context of generative models, this process is reminiscent both of Spread Divergences (Zhang et al., 2020) and dequantisation (Dinh et al., 2017; Salimans et al., 2017; Ho et al., 2019). As we demonstrate both theoretically and empirically, these noise additions enable finer-grained control over the harmonic content of the encoder mean network.

Further, we demonstrate that the Kullback-leibler (KL) divergence between the VAE latent prior and variational posterior in the VAE evidence lower bound (ELBO) serves as a single point of action to modulate the frequency content of both encoder and decoder networks; whereby a larger proportional weighting of the KL during training induces VAE networks with lower frequency components.

Finally, using this method of harmonic analysis, we extend Nash's Poincaré inequality to general Gaussian spaces. This shows that Gaussian spaces implicitly induce a soft constraint on the Lipschitz constant of their constituent function. In the VAE setting, we use these results to show that the Lipschitz constant of the VAE decoder decreases as the posterior variance increases.

This result links our work both to recent advances in the Lipschitz penalisation and regularisation of deep generative models (Adler and Lunz, 2018; Terjék, 2020) and more generally to the adversarial robustness literature where control of the Lipschitz constant im-

proves the robustness of models (Gouk et al., 2018; Yang et al., 2020; Hein and Andriushchenko, 2017; Tsuzuku et al., 2018). Our novel theoretical viewpoint on VAEs unifies the two emerging strands in the empirical and theoretical study of the robustness of VAEs to adversarial attack. The first strand aims to tune the noise of the VAE latent space by up-weighting various regularisation terms (Willetts et al., 2021; Camuto et al., 2021b). The second aims to directly control the Lipschitz constants of the underlying networks (Barrett et al., 2021). In our framework we link these two strands: the Gaussian noise of a VAE's latent space affects the Lipschitz constants of the VAE's constituent networks.

2 Background

Variational Autoencoders: VAEs (Kingma et al., 2014; Kingma and Welling, 2014), and the models they have inspired (Alemi et al., 2017; Kumar and Poole, 2020; Chen et al., 2018; Willetts et al., 2021), are deep latent variable models. Using $\mathbf{x} \in \mathcal{X}$ to denote data and $\mathbf{z} \in \mathcal{Z}$ to denote the latents with associated prior $p(\mathbf{z})$, a VAE simultaneously learns both a forward generative model, $p_\theta(\mathbf{x}|\mathbf{z})$, and an amortised approximate posterior distribution, $q_\phi(\mathbf{z}|\mathbf{x})$ (where θ and ϕ correspond to their respective parameters) which are typically implemented using neural networks [1]. These models are referred to as the decoder and encoder respectively, and a VAE can be thought of as a deep stochastic autoencoder. Under this autoencoder framework, one typically takes the reconstructions as deterministic, corresponding to the mean of the decoder, namely $g_\theta(\mathbf{z}) := \mathbb{E}_{p_\theta(\mathbf{x}|\mathbf{z})}[\mathbf{x}]$, a convention we adopt.

A VAE is trained by maximizing the evidence lower bound (ELBO) $\mathcal{L} = \mathbb{E}_{p_\mathcal{D}(\mathbf{x})}[\mathcal{L}(\mathbf{x})]$, where

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) \tag{2.1}$$

and $p_{\mathcal{D}(\mathbf{x})}$ is the empirical data distribution. The optimisation is carried out using stochastic gradient ascent with Monte Carlo samples to evaluate expectations over $\mathbf{z}$, typically employing the reparameterisation trick (Kingma and Welling, 2014). For example, for a Gaussian $q_\phi(\mathbf{z}|\mathbf{x})$, we draw samples as $\mathbf{z} = \boldsymbol{\mu}_\phi(\mathbf{x}) + \boldsymbol{\epsilon} \circ \boldsymbol{\sigma}_\phi(\mathbf{x}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, where $\circ$ is the element-wise product.

Previous work (Camuto et al., 2020, 2021b; Rezende and Viola, 2018) has shown the regularising effects of increasing the noisiness of VAE encodings, by increasing the variance of the variational posterior. VAEs are less likely to overfit under these settings and can even be more robust to adversarial attack. Our contribution here is to offer a simple framework that allows the study of VAEs from a functional and frequency domain perspective, and gives more precise insights into the polynomial order and spectral properties of encoder and decoder functions (rather than some abstract measure of model complexity (Rezende and Viola, 2018)) that different encoder variances induce.

Gaussian Spaces: A *Gaussian Space*, denoted $L_2(\mathbb{R}^n, \gamma)$ is an L_2 space, the space of square-integrable functions $f : \mathbb{R}^n \to \mathbb{R}$, equipped with the Gaussian measure $\gamma(x) = \prod_i \mathcal{N}(x_i|0, 1)$. This space has an inner product between two functions f and g: $\langle f, g\rangle = \mathbb{E}_{\gamma(x)}\left[f(x)g(x)\right]$.

On the real line, *the Hermite polynomials* form an *orthogonal basis* for the space $L_2(\mathbb{R}, \gamma)$. These are the polynomials of degree k that satisfy (Janson, 1997)

$$\mathbb{E}_{\gamma(x)}[H_k(x)H_m(x)] = k!\mathbf{1}\{m = k\}. \tag{2.2}$$

These polynomials can also be defined recursively, in a manner that allows a more intuitive understanding of their polynomial nature:

$$\begin{aligned} H_{k+1}(x) &= xH_k(x) - kH_{k-1}(x) \\ H_0(x) &= 1, H_1(x) = x. \end{aligned} \tag{2.3}$$

We have, for instance,

$$H_2(x) = x^2 - 1, H_3(x) = x^3 - 3x, H_4(x) = x^4 - 6x^2 + 3.$$

A function $f \in L_2(\mathbb{R}, \gamma)$ in this space can be expressed as a weighted sum of these polynomial functions, where $\hat{f}(k)$ are the *Hermite coefficients*:

$$f = \sum_{k \in \mathbb{N}} \frac{1}{k!} \hat{f}(k) H_k \tag{2.4}$$

$$\hat{f}(k) = \langle f, H_k\rangle = \mathbb{E}_{\gamma(x)}\left[f(x)H_k(x)\right]. \tag{2.5}$$

3 Gaussian Spaces and VAEs

We want to bring the tools of Gaussian space analysis to bear on VAEs, by viewing the latent space of the VAE as being equipped with a Gaussian measure. The VAE encoder parameterises an amortised (per-datapoint) posterior Gaussian distribution with mean $\mu_\phi(\mathbf{x})$ and *diagonal variance* $\mathrm{diag}(\sigma^2_\phi(\mathbf{x}))$. Thus we can view the latent space of a VAE as being equipped with a Gaussian measure that varies on a *per-datapoint basis*. This measure is a multivariate general Gaussian measure with mean $\mu_\phi(\mathbf{x})$ and *diagonal variance* $\mathrm{diag}(\sigma^2_\phi(\mathbf{x}))$. The encoder in some sense 'indexes' over

[1] Notation: We use bold letters to denote vectors or matrices and non-bolded letters to denote scalars. Matrices are capitalised.

the range of possible Gaussian measures in a data-dependent manner. The decoder, which acts on this latent space, would then be a member of some Gaussian space (the measure for which depends on $\mathbf{x}$).

As we were unable to find a comprehensive resource for spaces equipped with non-standard Gaussian measures we must first derive results for Gaussian spaces equipped with a *general* Gaussian measure (with mean μ and variance σ^2) before we apply Gaussian space analysis to VAEs. We start with results for functions acting on a univariate space, $f : \mathbb{R} \to \mathbb{R}$.

Proposition 1. *If $x \sim \mathcal{N}(\mu, \sigma^2)$, then we have that $\hat{x} = (x - \mu)/\sigma \sim \mathcal{N}(0, 1)$ and the Hermite polynomials are the polynomials of degree k that satisfy*

$$\mathbb{E}_{\gamma(\hat{x})}[H_k(\hat{x})H_m(\hat{x})] = k!\mathbf{1}\{m = k\},$$

where γ is the standard Gaussian measure. The family $\left\{\frac{1}{\sqrt{k!}}H_k(\hat{x}) : k \geq 0\right\}$ is then an orthonormal basis for $L_2(\mathbb{R}, \mathcal{N}(\mu, \sigma^2))$.

For the proof see Appendix A.1[2]. As such the polynomials $H_n(\hat{x})$ are the orthogonal polynomials for the $\mathcal{N}(\mu, \sigma^2)$ measure. We can calculate each of these polynomials by substituting x for $\hat{x} = (x - \mu)/\sigma$ in Equation (2.3). In higher dimensions ($\mathbb{R}^n$), for a function $f : \mathbb{R}^n \to \mathbb{R}$, we assume that we have a *diagonal* standard deviation $\mathbf{S}$, as this form of the standard deviation is most directly applicable to how VAEs are used in practice.

We denote the multivariate Gaussian measure as $\mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2)$ with covariance matrix populated by the element-wise square of $\mathbf{S}$. In this case the basis can be expressed using a multi-index $\boldsymbol{\alpha} \in \mathbb{N}^n$ (the sets of size n of non-negative integers), where

$$|\boldsymbol{\alpha}| = \sum_{i=1}^{n} \alpha_i, \qquad \mathbf{v}^{\boldsymbol{\alpha}} = \prod_{i=1}^{n} v_i^{\alpha_i}$$
$$\boldsymbol{\alpha}! = \prod_{i=1}^{n} \alpha_i!, \qquad g^{(\boldsymbol{\alpha})}(\mathbf{x}) = \frac{\partial^{|\boldsymbol{\alpha}|} g}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}}.$$

The $\boldsymbol{\alpha}^{\text{th}}$ multivariate Hermite polynomial (denoted $\mathcal{H}_{\boldsymbol{\alpha}}$) for the measure $\mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2)$ can then be expressed as the product of univariate polynomials indexed by $\boldsymbol{\alpha}$:

$$\mathcal{H}_{\boldsymbol{\alpha}}((\mathbf{x} - \boldsymbol{\mu})\mathbf{S}^{-1}) = \mathcal{H}_{\boldsymbol{\alpha}}(\hat{\mathbf{x}}) = \prod_i H_{\alpha_i}((x_i - \mu_i)/\sigma_i),$$

This stems from the fact that each H_{α_i} forms a basis in the univariate case, and that we assume a diagonal covariance, meaning that we can obtain the basis for $\mathbb{R}^n$ simply by tensorisation $H_{\alpha_1} \otimes \dots \otimes H_{\alpha_n}$. For the sake of completeness, in Appendix B we derive the Hermite polynomials for a Gaussian space with a *full covariance matrix*.

Because the set of $\mathcal{H}_{\boldsymbol{\alpha}}$ form an orthogonal basis, we can express functions $f \in L_2(\mathbb{R}^n, \mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2))$ as a weighted sum of these functions:

$$f = \sum_{\boldsymbol{\alpha} \in \mathbb{N}^n} \frac{1}{\boldsymbol{\alpha}!} \hat{f}(\boldsymbol{\alpha}) \mathcal{H}_{\boldsymbol{\alpha}} \tag{3.1}$$
$$\hat{f}(\boldsymbol{\alpha}) = \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f(\mathbf{x})\mathcal{H}_{\boldsymbol{\alpha}}(\hat{\mathbf{x}})\right]$$
$$\gamma(\hat{\mathbf{x}}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{S}^2).$$

Using these results, we can express the Hermite coefficients for a general Gaussian space in terms of the underlying measure's standard deviation $\mathbf{S}$ and the derivatives of a function f in the space.

Proposition 2. *Assume we have a function f in Gaussian space with diagonal covariance, $f \in L_2(\mathbb{R}^n, \mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2))$. Further assume that f is in C^∞, the class of infinitely differentiable functions. For $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2)$, then we have that $\hat{\mathbf{x}} = (\mathbf{x} - \boldsymbol{\mu})\mathbf{S}^{-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the Hermite coefficients can be expressed as*

$$\hat{f}(\boldsymbol{\alpha}) = (\text{diag}(\mathbf{S}))^{\boldsymbol{\alpha}} \, \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right], \gamma(\hat{\mathbf{x}}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{S}^2).$$

We can now express the variance of a function $L_2(\mathbb{R}^n, \mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2))$ as a sum of function derivatives, which in turns allows us to connect this variance to the Fourier domain.

Theorem 3. *Assume we have a function f in Gaussian space with diagonal covariance, $f \in L_2(\mathbb{R}^n, \mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2))$. Further assume that f is in C^∞, the class of infinitely differentiable functions and is L_2 integrable with respect to the Lebesgue measure. For $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2)$, then we have that $\hat{\mathbf{x}} = (\mathbf{x} - \boldsymbol{\mu})\mathbf{S}^{-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ such that $\gamma(\hat{\mathbf{x}}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{S}^2)$. The variance of f can be expressed as*

$$\text{Var}(f) = \sum_{|\boldsymbol{\alpha}| \geq 1} \frac{(\text{diag}(\mathbf{S}))^{2\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \left|\mathbb{E}_{\gamma(\hat{x})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right]\right|^2$$
$$= \sum_{|\boldsymbol{\alpha}| \geq 1} \frac{(\text{diag}(\mathbf{S}))^{2\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \left|\int_{\mathbb{R}^n} (\boldsymbol{\omega})^{\boldsymbol{\alpha}} \mathcal{F}(\boldsymbol{\omega}) \overline{\mathcal{P}(\boldsymbol{\omega})} d\boldsymbol{\omega}\right|^2,$$

where $\mathcal{P}$ is the Fourier transform of the Gaussian measure $\mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2)$ given by $\mathcal{P}(\boldsymbol{\omega}) = \det(\mathbf{S})\mathcal{G}(\boldsymbol{\omega}\mathbf{S})e^{-i\boldsymbol{\omega}\boldsymbol{\mu}\mathbf{S}^{-1}}$ (where $\mathcal{G}$ is the Fourier transform of the standard Gaussian measure γ) and $\mathcal{F}$ is the Fourier transform of f, and $\boldsymbol{\alpha} \in \mathbb{N}^n$.

We also note that even if a function is *not* infinitely differentiable, we can also express $\text{Var}(f)$ as a weighted

[2] All proofs are presented in Appendix A

sum of the Hermite coefficients (see the Proof of Theorem 3):

$$\mathrm{Var}(f) = \sum_{|\boldsymbol{\alpha}|\geq 1} \frac{1}{\boldsymbol{\alpha}!} \left|\hat{f}(\boldsymbol{\alpha})\right|^2. \tag{3.2}$$

If we now assume that f is infinitely differentiable, then we obtain the intuitive result that high-frequency functions correspond to larger Hermite coefficients associated with higher degree Hermite polynomials. This can be deduced by combining Proposition 2 and Theorem 3:

$$\left|\hat{f}(\boldsymbol{\alpha})\right|^2 = \left|\int_{\mathbb{R}^n} (\mathrm{diag}(\mathbf{S}))^{\boldsymbol{\alpha}} (\boldsymbol{\omega})^{\boldsymbol{\alpha}} \mathcal{F}(\boldsymbol{\omega}) \overline{\mathcal{P}(\boldsymbol{\omega})} d\boldsymbol{\omega}\right|^2. \tag{3.3}$$

As the variance of the underlying Gaussian measure increases, larger degree polynomials contribute more heavily to $\mathrm{Var}(f)$. This is captured in the terms $(\mathrm{diag}(\mathbf{S}))^{2\boldsymbol{\alpha}}$ of Equation (3.3), meaning that increases in $|(\mathrm{diag}(\mathbf{S}))|$ disproportionately increase large $\boldsymbol{\alpha}$ terms, i.e the coefficients associated with higher degree polynomial terms. More succinctly, Gaussian spaces with large variances naturally induce larger Hermite coefficients associated with higher degree polynomials, which themselves are associated with higher frequency components in the Fourier domain.

Lipschitzness Finally we redevelop Nash's Poincaré inequality for general Gaussian spaces to show that the variance of a function $f \in L_2(\mathbb{R}^n, \mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2))$, assuming that this function is Lipschitz continuous, can be upper-bounded by using its Lipschitz constant.

Proposition 4. *Let $f \in L_2(\mathbb{R}^n, \mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2))$ be a function that is Lipschitz continuous with Lipschitz constant L. Further assume that $f^{(|\boldsymbol{\alpha}|=1)} \in L_2(\mathbb{R}^n, \mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2))$, we have*

$$\mathrm{Var}(f) \leq L^2 \|\mathbf{S}\|_2^2.$$

This bound is relatively tight in that for a measure $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and a function $f(\mathbf{x}) = \frac{1}{n}\sum x_i$ the bound becomes an equality: $L = n^{-1}$ and $\mathrm{Var}(f) = n^{-2}$. We can now use these results for a general Gaussian space to study VAE decoders.

3.1 Gaussian Spaces and VAE decoders

The amortised posterior distributions of a VAE are typically chosen as Gaussians, where, to allow for the backpropagation of gradients, samples are reparameterised as

$$\mathbf{z} = \boldsymbol{\mu}_\phi(\mathbf{x}) + \boldsymbol{\sigma}_\phi(\mathbf{x}) \circ \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{3.4}$$

This sample is then fed into the decoder to calculate the likelihood term of the ELBO (see Eq (2.1)). Implicit in this arrangement is that the VAE decoder operates on samples $\mathbf{z}$ in latent space $\mathcal{Z}$ equipped with a Gaussian measure, captured by the $\boldsymbol{\epsilon}$ term in Eq (3.4).

Assuming the function parameterised by the decoder g, for a given point $\mathbf{x}$, is L_2 integrable with respect to the general Gaussian measure, i.e $g \in L_2(\mathbb{R}^n, \mathcal{N}(\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi(\mathbf{x})))$ (this is reasonable as we are effectively training the decoder to be in this space) then we can directly apply the theory of Gaussian spaces to the VAE decoder. Assuming we have a Gaussian likelihood with a fixed standard deviation σ_θ across dimensions, we express the likelihood for a single point using a bias-variance decomposition, which gives the sum of an error term captured by the bias, and a regularisation term captured by the variance:

$$\begin{aligned}
&\frac{1}{\sigma_\theta^2} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[(g_\theta(\mathbf{z}) - \mathbf{x})^2\right] \\
&= \frac{1}{\sigma_\theta^2} \left(\left(\mathrm{Bias}_{q_\phi(\mathbf{z}|\mathbf{x})} \left(g_\theta(\mathbf{z})\right)\right)^2 + \mathrm{Var}_{q_\phi(\mathbf{z}|\mathbf{x})}(g_\theta(\mathbf{z}))\right)
\end{aligned} \tag{3.5}$$

$$\begin{aligned}
&\mathrm{Bias}_{q_\phi(\mathbf{z}|\mathbf{x})} \left(g_\theta(\mathbf{z})\right) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[g_\theta(\mathbf{z})\right] - \mathbf{x} \\
&\mathrm{Var}_{q_\phi(\mathbf{z}|\mathbf{x})} \left(g_\theta(\mathbf{z})\right) \\
&\qquad = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\left(g_\theta(\mathbf{z}) - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[g_\theta(\mathbf{z})\right]\right)^2\right].
\end{aligned}$$

We can directly apply Theorem 3 to the decoder function if we assume that the decoder function is infinitely differentiable. This holds for networks with the Sigmoid activation function for instance (Hornik, 1991). We can then swap out $\mathrm{diag}(\mathbf{S})$ for the encoder standard deviation $\boldsymbol{\sigma}_\phi(\mathbf{x})$ and repeatedly apply the Theorem to each output $g_{\theta,i}(\mathbf{z}), i \in 1, \ldots, d$ of the decoder. For an output i of the decoder and $\boldsymbol{\alpha} \in \mathbb{N}^n$, we have that

$$\begin{aligned}
\mathrm{Var}_{q_\phi(\mathbf{z}|\mathbf{x})}(g_{\theta,i}) &= \sum_{|\boldsymbol{\alpha}|\geq 1} \frac{(\boldsymbol{\sigma}_\phi(\mathbf{x}))^{2\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \left|g_{\theta,i}^{(\boldsymbol{\alpha})}(\mathbf{z})\right|^2 \\
&= \sum_{|\boldsymbol{\alpha}|\geq 1} \frac{(\boldsymbol{\sigma}_\phi(\mathbf{x}))^{2\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \left|\int_{\mathbb{R}^n} (\boldsymbol{\omega})^{\boldsymbol{\alpha}} \mathcal{F}_{\theta,i}(\boldsymbol{\omega}) \overline{\mathcal{P}_\phi(\boldsymbol{\omega})} d\boldsymbol{\omega}\right|^2.
\end{aligned} \tag{3.6}$$

$\mathcal{P}_\phi$ is the Fourier transform of the Gaussian measure $\mathcal{N}(\mu_\phi(\mathbf{x}), \sigma_\phi^2(\mathbf{x}))$ and $\mathcal{F}_{\theta,i}$ is the Fourier transform of $g_{\theta,i}$.

As $\boldsymbol{\sigma}_\phi(\mathbf{x})$ increases, particularly for $\boldsymbol{\sigma}_\phi(\mathbf{x}) \geq 1$, larger $\boldsymbol{\alpha}$ terms in the sum, and large frequencies to the power $\boldsymbol{\alpha}$ will begin to be disproportionately penalised in the Var term of the bias variance decomposition of the VAE likelihood. Thus larger $\boldsymbol{\sigma}_\phi(\mathbf{x})$ will result in a larger penalisation of the high-frequency components of the decoder function. We note that this penalisation is modulated on a *per-datapoint basis* by $\boldsymbol{\sigma}_\phi(\mathbf{x})$, but

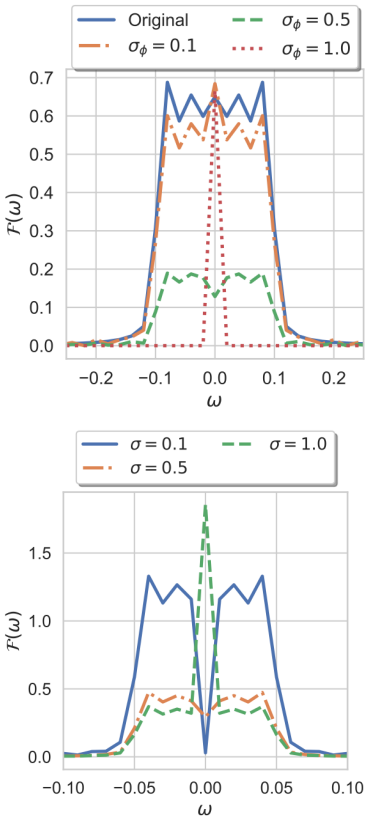


Figure 2: [top] FFT of the VAE reconstructions in Figure 1. For larger $\boldsymbol{\sigma}_\phi$, the spectrum loses high frequencies. [bottom] FFT of μ_ϕ of VAEs trained on $\text{sinc}(5t) + \mathcal{N}(0, \sigma^2\mathbf{I})$. For larger σ the spectrum loses high frequency content.

that VAEs which on average have larger encoder variances over training inputs will learn lower frequency decoders. Because of the link between the decoder variance, the Hermite polynomials, and the frequency domain (see Equation (3.3)), the lower frequency function learned by the decoder, for each $\mathbf{z}|\mathbf{x}$, *also* corresponds to a function that can be described as a lower degree polynomial.

To demonstrate these findings, in Figure 1 we use a fixed encoder variance σ_ϕ^2, uniform across dimensions, on VAEs, with Sigmoid activation function in their networks, trained on data from the $\text{sinc}(5t), t \in [-1, 1]$ function. As σ_ϕ increases, we can qualitatively ascertain that the decoder function loses higher frequency components and that the optimal polynomial to describe the decoder function (approximated by polynomial regression) *decreases* in its degree. Quantitatively measuring the Fourier transform of the decoder here is simple, we collect the reconstruction of all inputs, ordering by increasing t. We then take the fast Fourier transform (FFT) of this aggregate reconstructed function. We show this in Figure 2 where as σ_ϕ increases, the decoder learns a lower frequency Fourier representation.

Though our theoretical results hold for classes of infinitely differentiable neural networks, we empirically confirm they still hold for more complex neural network architectures with ReLU activation functions, trained on large multivariate datasets. In Figure 3 we show that for a fixed encoder variance σ_ϕ^2, as σ_ϕ increases reconstructed CelebA images from a convolutional VAE become more similar and start to lose diversity – the images lose mid to high level frequencies as measured by a 2D-FFT – suggesting that the decoder learns a lower frequency function. To support this, we plot the mean 1D-Non-uniform discrete Fourier transform (NUDFT) (Bagchi and Mitra, 1999) across the d dimensions of the output, which shows that models trained with larger σ_ϕ learn functions that on average are of much lower frequency content than smaller σ_ϕ models. This demonstrates that the multi-dimensional output decreases in its high-frequency content over the d output dimensions, as predicted by our theory. In Figure D.6 of the Appendix we show that this also holds for fully-connected VAEs trained on multivariate sinusoids.

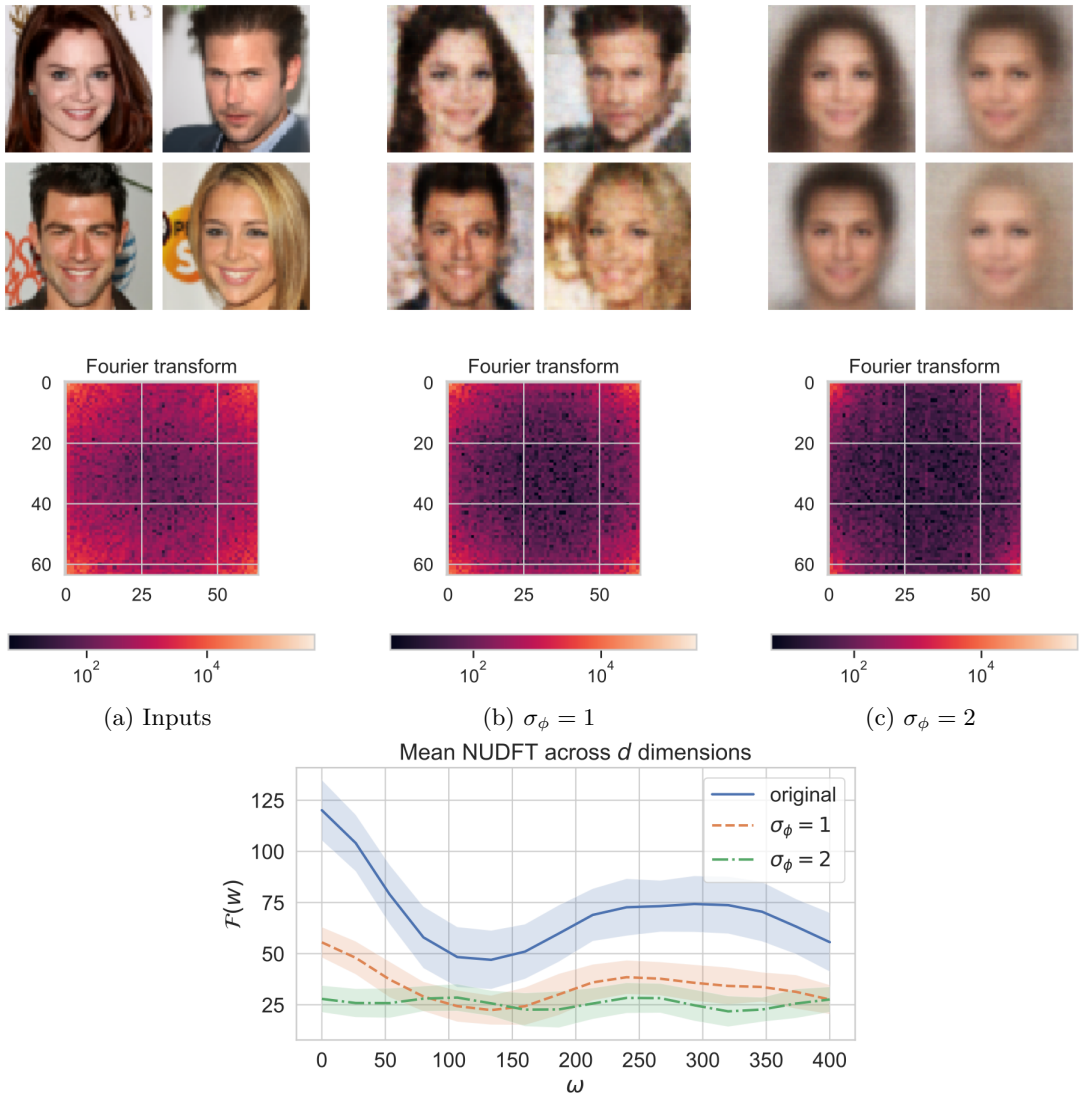
3.2 Gaussian Spaces and VAE encoders

Here we use the analysis developed in previous section to analyze the Harmonic content of the VAE encoder. In the typical VAE setup, we don't usually consider the case of Gaussian noise on inputs and we make no assumptions as to the underlying measure of the input space; such that we cannot directly apply the previous Gaussian space analysis to the VAE encoder.

The perspective we've established, however, of the latent space as a Euclidean space equipped with a general Gaussian measure, and the perspective of the decoder as belonging to Gaussian space equipped with this same measure, can inform our analysis. The encoder of the VAE effectively parameterises the latent space measure for each input $\mathbf{x}$ such that the function the encoder is learning is the measure $\mathcal{N}(\boldsymbol{\mu}_\phi(\mathbf{x}), \boldsymbol{\sigma}_\phi(\mathbf{x}))$. In Theorem 3 we have already established that the Fourier transform of this measure is given by

$$\mathcal{P}(\boldsymbol{\omega}) = \det(\boldsymbol{\Sigma}_\phi(\mathbf{x}))\mathcal{G}(\boldsymbol{\omega}\boldsymbol{\Sigma}_\phi(\mathbf{x}))e^{-i\boldsymbol{\omega}\boldsymbol{\mu}_\phi(\mathbf{x})\boldsymbol{\Sigma}_\phi^{-1}(\mathbf{x})},$$

where $\boldsymbol{\Sigma}_\phi(\mathbf{x})$ is a diagonal matrix with the elements of $\boldsymbol{\sigma}_\phi(\mathbf{x})$ on its diagonal. Much like the decoder, as the encoder variance decreases, the function parameterised by the encoder increases in amplitude in its high-frequency components. However, here the frequency content of the encoder function is *intrinsic* to



(a) Inputs (b) $\sigma_\phi = 1$ (c) $\sigma_\phi = 2$

Figure 3: We train convolutional VAEs (see Appendix C for network details), with ReLU activation functions in their networks, on the CelebA dataset, with a 64 dimension latent space $\mathcal{Z}$. As before we fix the encoder σ_ϕ. **[first + second rows]** We show 4 images and the 2D-FFT of the image in the upper right quadrant for a) the training images used b) VAE reconstructions of the 4 images when $\sigma_\phi = 1$ c) VAE reconstructions for $\sigma_\phi = 2$. As σ_ϕ increases mid-high level frequencies are increasingly dampened relative to the original image. **[bottom row]** Positive components of the mean 1D-NUDFT of the d dimensions of the output of these models (calculated across 2000 images for each of the $d = 12288$ dimensions). Shading corresponds to the std. dev. over d dimensions.

the Gaussian measure, whereas the frequency content of the decoder is a result of the relative weighting of the decoder variance in the VAE likelihood, and as such is enforced by optimisation. Note that $e^{-i\boldsymbol{\omega}\boldsymbol{\mu}_\phi(\mathbf{x})\boldsymbol{\Sigma}_\phi^{-1}(\mathbf{x})}$ only modulates the phase of the Fourier transform and does not alter the amplitude of its frequency content. As such the mean does not affect the frequency spectrum of the posterior Gaussian. In Figure D.5 of the Appendix we demonstrate that for a univariate Gaussian, decreasing the variance increases the magnitude of high-frequency components in the Fourier domain, whereas altering the mean has no such effect.

This perspective gives us an idea of how to alter the harmonic content of the measure parameterised by the encoder *overall*. If instead we want to modulate the frequency content of the *individual networks* that are used to parameterise the Gaussian measure we need a different perspective. In the next section we show that using the theory developed previously, adding Gaussian noise to the input data offers a simple and effec-

tive method for modulating the frequency content of the encoder mean.

3.2.1 Noisy inputs

Let us now assume that we add Gaussian noise to the data $\mathbf{x}$ such that our input for each point $\mathbf{x}$ is a Gaussian space with a Gaussian measure with mean $\mathbf{x}$, variance σ^2:

$$\tilde{\mathbf{x}} = \mathbf{x} + \sigma\boldsymbol{\nu}, \boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Under this noisy input, the expectation of the KL between the amortised posterior distribution and the unit Gaussian prior can be written as

$$\begin{aligned}
&\mathbb{E}_{\gamma(\boldsymbol{\nu})}\left[\mathrm{KL}(q_\phi(\mathbf{z}|\tilde{\mathbf{x}})\|p(\mathbf{z}))\right] \\
&= \frac{1}{2}\,\mathbb{E}_{\gamma(\boldsymbol{\nu})}\left[\log\left(\frac{1}{\prod_{i=1}^{n}\sigma_{\phi,i}(\tilde{\mathbf{x}})}\right)\right. \\
&\qquad \left. - n + \sum_{i=1}^{n}\sigma_{\phi,i}(\tilde{\mathbf{x}}) + \left\|\boldsymbol{\mu}_\phi(\tilde{\mathbf{x}})\right\|_2^2\right]. \qquad (3.7)
\end{aligned}$$

We now have a Gaussian space variance $\mathrm{Var}(\boldsymbol{\mu}_\phi) = \mathbb{E}_{\gamma(\boldsymbol{\nu})}\left[\left\|\boldsymbol{\mu}_\phi(\tilde{\mathbf{x}})\right\|_2^2\right]$ penalised at each iteration. As before for the decoder we can directly apply Theorem 3 to the encoder mean function, by swapping out $\mathrm{diag}(\mathbf{S})$ for the data standard deviation $\sigma\mathbf{I}$ and repeatedly applying the Theorem to each output $\mu_i(\tilde{\mathbf{x}}), i \in 1, \ldots, n$ of the encoder. We can expect two things to happen as σ increases. The frequency content of the encoder mean should decrease, and generally the function for each $\mathbf{x}$ can be described using an increasingly lower degree polynomial. Thus we can add Gaussian noise to a VAE's inputs to modulate the frequency content of the encoder mean function. We demonstrate these findings in Figure 2. Note that though our theory is not directly applicable to the encoder variance network, if we fix σ_ϕ so as to modulate the variance of the decoder, then we have full control over the variance of the encoder by adding noise to the VAE inputs.

3.3 The KL and frequency content

In previous experiments we fixed the variance of the encoder for analytical purposes. In practice however we often want to learn this variance, and as such, need tools by which we can modify the frequency content of the encoder and decoder whilst learning the encoder variance. As we now show, the frequency content of both the encoder mean and decoder networks can be controlled by altering the weighting of KL in the ELBO.

Consider the β-VAE setting (Higgins et al., 2017), whereby the KL component of the ELBO is premultiplied by a factor β. The strength of this penalisation changes the optimal encoder variance $\sigma_\phi(x)$: as β increases we can expect $\sigma_\phi(x)$ to tend to 1 – see Theorem 2 of Camuto et al. (2020) for a formal statement of this. This then means that we can modulate the scale of the latent noise experienced by the model during training, where $\sigma_\phi(x)$ is learnt and no longer fixed as in previous experiments. Because increasing β tends to increase $\sigma_\phi(x)$, our theory would predict that gradual increases in β would be accompanied by the decoder losing higher-frequency content in the Fourier domain.

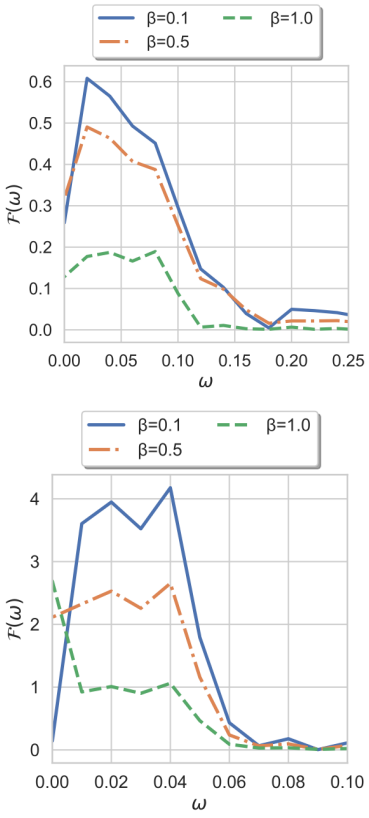


Figure 4: Positive components of Figure 2 with variable β penalisation on the KL term of the ELBO. As β increases higher frequencies are increasingly dampened for the decoder [top] and encoder [bottom]

We also note that the variance of the encoder mean network $\mathrm{Var}(\boldsymbol{\mu}_\phi) = \mathbb{E}_{\gamma(\boldsymbol{\nu})}\left[\left\|\boldsymbol{\mu}_\phi(\tilde{\mathbf{x}})\right\|_2^2\right]$ experiences a greater penalisation as the weight β increases, as seen in the decomposition of the KL in section 3.2.1. This means that we expect larger KL weightings to induce encoder mean functions with lower frequency content (when the input space is a Gauss space). See Figure 4 for a demonstration of both these claims.

3.4 VAE Lipschitzness

We know that by Proposition 4 the variance of a function (that is once differentiable) provides a relatively

tight lower bound on the Lipschitz constant of a neural network. As the encoder variance $\boldsymbol{\sigma}^2_\phi(\mathbf{x})$ increases, the variance of the decoder function $\mathrm{Var}_{q_\phi(\mathbf{z}|\mathbf{x})}(g_\theta(\mathbf{z}))$ decreases, which by Proposition 4 will lead to a smaller lower bound on the Lipschitz constant of the decoder. Due to the computational costs of estimating the Lipschitz constants of networks, we restrict our empirical analysis here to fully-connected VAEs and use *layer-wise* LipSDP (Fazlyab et al., 2019) to obtain estimates. In Appendix D.3 we confirm that regulating the encoder variance allows for us to impose a soft constraint of the Lipschitzness of the VAE decoder. We also show that adding Gaussian noise on data, gives us finer control on the Lipschitz constant of the encoder. Finally, we also demonstrate that this modulation of the encoder and decoder Lipschitzness is a purveyor of adversarial robustness for VAEs.

3.5 Potential Limitations

Our results on Lipschitzness would be stronger if Proposition 4 offered an upper bound on the Lipschitz constant, in that we could certify a maximum Lipschitz constant. Also, the encoder variance does not *directly* affect the frequency content of the encoder, only secondarily. Noise injections on data are needed to provide full control over the frequency content and Lipschitzness of the VAE networks, whereas a single parameter to control would have been a stronger result. We also recognise that the results in Figure D.7 would have been stronger if the variance of the likelihood degradation results was such that the shading of the curves did not overlap for the different values of σ and σ_ϕ.

4 Conclusion

Using the lens of Gaussian spaces, we demonstrated that the variance of the latent encodings features as an important parameter for regularising VAE decoders, controlling their Lipschitz constants, and improving their adversarial robustness. By applying the same framework to the VAE encoder we show that simply adding Gaussian noise to VAE inputs offers the same control over the VAE encoder. This work lays the foundation for analysing the effect of Gaussian priors on VAEs and offers a novel framework from which to understand the functions learned by a VAE's networks. It also paves the way for the study of the effect of *non-Gaussian* priors on VAE networks.

5 Acknowledgements

We thank Professor Svante Janson, Professor Wilfreda Urbina-Romero, and Professor Tom Alberts for their invaluable guidance on Gaussian Spaces. This research was directly funded by the Alan Turing Institute under Engineering and Physical Sciences Research Council (EPSRC) grant EP/N510129/1. Alexander Camuto was supported by an EPSRC Studentship.

References

Jonas Adler and Sebastian Lunz. Banach Wasserstein GAN. In *NeurIPS*, 2018.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. *ICLR*, 2017.

Sonali Bagchi and Sanjit K. Mitra. *The Nonuniform Discrete Fourier Transform and Its Applications in Signal Processing*. Kluwer Academic Publishers, USA, 1999.

Ben Barrett, Alexander Camuto, Matthew Willetts, and Tom Rainforth. Certifiably Robust Variational Autoencoders. *arXiv*, 2021.

Chris M. Bishop. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Computation*, 1995.

Martin Burger and Andreas Neubauer. Analysis of Tikhonov regularization for function approximation by neural networks. *Neural Networks*, 2003.

Alexander Camuto, Matthew Willetts, Umut Şimşekli, Stephen Roberts, and Chris Holmes. Explicit Regularisation in Gaussian Noise Injections. In *NeurIPS*, 2020.

Alexander Camuto, Xiaoyu Wang, Lingjiong Zhu, Chris Holmes, Mert Gürbüzbalaban, and Umut Şimşekli. Asymmetric Heavy Tails and Implicit Bias in Gaussian Noise Injections. In *ICML*, 2021a.

Alexander Camuto, Matthew Willetts, Stephen Roberts, Chris Holmes, and Tom Rainforth. Towards a Theoretical Understanding of the Robustness of Variational Autoencoders. *AISTATS*, 2021b.

Tian Qi Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *NeurIPS*, 2018.

Jeremy Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. *ICML*, 2019.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *ICLR*, 2017.

Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George J. Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks, 2019.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael Cree. Regularisation of neural networks by enforcing lipschitz continuity. *arXiv*, 2018.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*, pages 2266–2276, 2017.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design. *ICML*, 2019.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 1991.

Svante Janson. *Gaussian Hilbert Spaces*. Cambridge Tracts in Mathematics. Cambridge University Press, 1997.

Diederik P Kingma and Jimmy Lei Ba. Adam: A Method for Stochastic Optimisation. In *ICLR*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

Diederik P. Kingma, Danilo Jimenez Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. *CoRR*, 2014.

Abhishek Kumar and Ben Poole. On Implicit Regularization in β-VAEs. *ICML*, 2020.

Danilo Jimenez Rezende and Fabio Viola. Taming vaes, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *ICML*, 2014.

Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. PixelCNN++: Improving the PixelCnn with discretized logistic mixture likelihood and other modifications. In *ICLR*, 2017.

Rui Shu, Hung H. Bui, Shengjia Zhao, Mykel J. Kochenderfer, and Stefano Ermon. Amortized inference regularization. In *NeurIPS*, 2018.

Dávid Terjék. Adversarial lipschitz regularization, 2020.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In *NeuRIPS*, 2018.

Andrew R. Webb. Functional Approximation by Feed-Forward Networks: A Least-Squares Approach to Generalization. *IEEE Transactions on Neural Networks*, 1994.

Matthew Willetts, Alexander Camuto, Tom Rainforth, Stephen Roberts, and Chris Holmes. Improving VAEs' Robustness to Adversarial Attack. *ICLR*, 2021.

Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. Adversarial robustness through local lipschitzness. *arXiv*, 2020.

Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *NeurIPS*, 2019.

Mingtian Zhang, Peter Hayes, Thomas Bird, Raza Habib, and David Barber. Spread Divergence. *ICML*, 2020.

Appendix for Variational Autoencoders: a Harmonic Perspective

A Technical Proofs

Before we begin the proofs we give an alternative definition of the Hermite polynomials which uses the k^{th} divergence operator (δ^k, (Janson, 1997)). This definition greatly simplifies some of our later proofs.

$$H_k = \delta^k 1(x), \tag{A.1}$$

where 1 is shorthand for the function that is identically 1. The divergence operator is defined using $\text{Dom}(\delta^k)$, which is the subset of functions $g \in L_2(\mathbb{R}, \gamma)$ for which there exists a $c > 0$ such that for all functions for which the derivative up to degree k obeys $f^{(k)} \in L_2(\mathbb{R}, \gamma)$:

$$\left| \int_{\mathbb{R}} f^{(k)}(x) g(x) d\gamma(x) \right| \leq c \sqrt{\left| \int_{\mathbb{R}} f^{(k)}(x) d\gamma(x) \right|}.$$

$1(x)$ for example, as defined in Equation (A.1) is in $\text{Dom}(\delta^k)$. The k^{th} divergence can then be defined as follows. If $g \in \text{Dom}(\delta^k)$ then $\delta^k g$ is the unique element of $L_2(\mathbb{R}, \gamma)$ such that for all functions for which the derivative up to degree $\boldsymbol{\alpha}$ obeys $f^{(k)} \in L_2(\mathbb{R}, \gamma)$ we have:

$$\int_{\mathbb{R}} f(x) \delta^k g(x) d\gamma(x) = \int_{\mathbb{R}} f^{(k)}(x) d\gamma(x). \tag{A.2}$$

For the general Gaussian space, this becomes by substitution of variables:

$$H_k = \delta^k 1(\hat{x}), \qquad \hat{x} = (x - \mu)/\sigma. \tag{A.3}$$

In the multivariate case, for the general Gaussian space with diagonal standard deviation, we have that $\mathcal{H}_{\boldsymbol{\alpha}} = \delta^{\boldsymbol{\alpha}} 1(\hat{\mathbf{x}})$, $\hat{\mathbf{x}} = (\mathbf{x} - \boldsymbol{\mu})\mathbf{S}^{-1}$, where δ now is the vector-valued divergence operator indexed by the multi-index $\boldsymbol{\alpha}$.

A.1 Proof of Proposition 1

Proof. Due to the scaling and centering relation for Gaussians a simple change of variables $\hat{x} = (x - \mu)/\sigma$ in Equation (2.2) gives us the first part of the proof. Thus

$$\mathbb{E}_{\gamma(\hat{x})}[H_k(\hat{x}) H_m(\hat{x})] = k! \mathbf{1}\{m = k\},$$

and thus $\left\{ \frac{1}{k!} H_k : k \geq 0 \right\}$ is orthonormal in $L_2(\mathbb{R}, \mathcal{N}(\mu, \sigma^2))$ because $\gamma(\hat{x}) = \mathcal{N}(\mu, \sigma^2)$.

We know that for $k \geq 0$ the polynomial H_k has degree k, hence it suffices to demonstrate that the mononomials of degree k, $\{y^k : k \in \mathbb{N}, \hat{x} = (x - \mu)/\sigma\}$ are dense in $L_2(\mathbb{R}, \gamma(\hat{x}))$ to show that $\left\{ \frac{1}{\boldsymbol{\alpha}!} H_k : k \geq 0 \right\}$ is a basis for $L_2(\mathbb{R}, \gamma(\hat{x}))$.

The Elementary Hahn-Banach theorem shows that if $g \in L_2(\mathbb{R}, \gamma)$ is such that $\int_{\mathbb{R}} g(\hat{x}) \hat{x}^k d\gamma(\hat{x}) = 0, \forall k \in \mathbb{N}^+$ then it suffices to show that $g = 0$ almost everywhere to demonstrate that the mononomials $\hat{x}^k$ are a dense subspace of $L_2(\mathbb{R}, \gamma(\hat{x}))$. This proof generally follows that used for the standard Gaussian measure. Assume we have a g that satisfies $\int_{\mathbb{R}} g(\hat{x}) \hat{x}^k d\gamma(\hat{x}) = 0$. By the series expansion of the exponential function we know that for all $t \in \mathbb{R}$

$$\int_{\mathbb{R}} g(\hat{x}) e^{ity} d\gamma(\hat{x}) = \lim_{m \to \infty} \sum_{k=0}^{m} \frac{(it)^k}{k!} \int_{\mathbb{R}} g(\hat{x}) \hat{x}^k d\gamma(\hat{x}) = 0.$$

Thus we have that $\int_{\mathbb{R}} g(\hat{x}) e^{ity} d\gamma(\hat{x}) = 0, \forall t$. Because e^{ity} is injective we have that $g(\hat{x}) = 0, \forall \hat{x} \in \mathbb{R}$.

□

A.2 Proof of Proposition 2

Proof. We begin with the univariate case, from which we build up to the multivariate case. Here we use the divergence operator definition of the Hermite polynomials, seen in Equation (A.1). In our case by substitution of variables $H_k(\hat{x}) = \delta^k 1(\hat{x})$ and we can express x as $x = y\sigma + \mu$, thus:

$$\int_{\mathbb{R}} f(y\sigma+\mu)H_k(\hat{x})d\gamma(\hat{x}) = \int_{\mathbb{R}} f(y\sigma+\mu)\delta^k 1(\hat{x})d\gamma(\hat{x}) = \int_{\mathbb{R}} \sigma^k f^k(\hat{x}\sigma+\mu)d\gamma(\hat{x}).$$

As mentioned in the main paper, in the multivariate case, the divergence operator is an almost exact analogue to that of the univariate case. Due to the tensorisation of bases to form $\mathcal{H}$ the indices α simply become a multi-index and we recover that $\hat{f}(\boldsymbol{\alpha}) = \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[\mathbf{S}^{\boldsymbol{\alpha}} f^{(\boldsymbol{\alpha})}(\mathbf{x})\right]$.

□

A.3 Proof of Theorem 3

Proof. By Proposition 2 we know that we can express a function $f \in L_2(\mathbb{R}^n, \mathcal{N}(\boldsymbol{\mu}, \mathbf{S}^2))$ that is infinitely differentiable as:

$$f = \sum_{\boldsymbol{\alpha}\in\mathbb{N}^n} \frac{1}{\boldsymbol{\alpha}!}(\text{diag}(\mathbf{S}))^{\boldsymbol{\alpha}}\, \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right] \mathcal{H}_{\boldsymbol{\alpha}}, \qquad \gamma(\hat{\mathbf{x}}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{S}^2).$$

Now note that the expectation of f is simply the first Hermite coefficient with degree 0:

$$\mathbb{E}_{\gamma(\hat{\mathbf{x}})}[f] = \sum_{|\boldsymbol{\alpha}|=0} \frac{1}{\boldsymbol{\alpha}!}(\text{diag}(\mathbf{S}))^{\boldsymbol{\alpha}}\, \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right] \mathcal{H}_{\boldsymbol{\alpha}}, \qquad \gamma(\hat{\mathbf{x}}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \mathbf{S}^2).$$

Now taking the variance of f:

$$\begin{aligned}
\text{Var(f)} &= \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[\left(f - \mathbb{E}_{\gamma(\hat{\mathbf{x}})}[f]\right)^2\right] \\
&= \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[\left(\sum_{\boldsymbol{\alpha}\in\mathbb{N}^n} \frac{1}{\boldsymbol{\alpha}!}(\text{diag}(\mathbf{S}))^{\boldsymbol{\alpha}}\, \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right] \mathcal{H}_{\boldsymbol{\alpha}} - \sum_{|\boldsymbol{\alpha}|=0} \frac{1}{\boldsymbol{\alpha}!}(\text{diag}(\mathbf{S}))^{\boldsymbol{\alpha}}\, \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right] \mathcal{H}_{\boldsymbol{\alpha}}\right)^2\right] \\
&= \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[\left(\sum_{|\boldsymbol{\alpha}|\geq 1} \frac{1}{\boldsymbol{\alpha}!}(\text{diag}(\mathbf{S}))^{\boldsymbol{\alpha}}\, \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right] \mathcal{H}_{\boldsymbol{\alpha}}\right)^2\right].
\end{aligned}$$

Recall that the bases $\mathcal{H}_{\boldsymbol{\alpha}}$ are orthogonal such that

$$\mathbb{E}_{\gamma(\hat{\mathbf{x}})}[\mathcal{H}_{\boldsymbol{\alpha}}(\hat{\mathbf{x}})\mathcal{H}_m(\hat{\mathbf{x}})] = \boldsymbol{\alpha}!\{m = \boldsymbol{\alpha}\}.$$

As such we have that:

$$\begin{aligned}
\text{Var(f)} &= \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[\left(\sum_{|\boldsymbol{\alpha}|\geq 1} \frac{1}{\boldsymbol{\alpha}!}(\text{diag}(\mathbf{S}))^{\boldsymbol{\alpha}}\, \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right] \mathcal{H}_{\boldsymbol{\alpha}}\right)^2\right] \\
&= \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[\sum_{|\boldsymbol{\alpha}|\geq 1}\left(\frac{1}{\boldsymbol{\alpha}!}(\text{diag}(\mathbf{S}))^{\boldsymbol{\alpha}}\, \mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right] \mathcal{H}_{\boldsymbol{\alpha}}\right)^2\right] \\
&= \sum_{|\boldsymbol{\alpha}|\geq 1} \frac{(\text{diag}(\mathbf{S}))^{2\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}\left|\mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right]\right|^2.
\end{aligned}$$

This completes the first part of the proof.

For the second part we focus on terms of the form

$$\mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right] = \int_{\mathbb{R}^n} f^{(\boldsymbol{\alpha})}(\hat{\mathbf{x}}\mathbf{S}+\boldsymbol{\mu})d\gamma(\hat{\mathbf{x}}) = \int_{\mathbb{R}^n} f^{(\boldsymbol{\alpha})}(\hat{\mathbf{x}}\mathbf{S}+\boldsymbol{\mu})\gamma(\hat{\mathbf{x}})d\hat{\mathbf{x}}.$$

Both f and γ are L_2 integrable with respect to the Lebesgue measure. As such we can directly apply Plancherel's theorem to demonstrate that

$$\begin{aligned}
\mathbb{E}_{\gamma(\hat{\mathbf{x}})}\left[f^{(\boldsymbol{\alpha})}(\mathbf{x})\right] &= \int_{\mathbb{R}^n} f^{(\boldsymbol{\alpha})}(\hat{\mathbf{x}}\mathbf{S}+\boldsymbol{\mu})\gamma(\hat{\mathbf{x}})d\hat{\mathbf{x}} \\
&= \int_{\mathbb{R}^n} f^{(\boldsymbol{\alpha})}(\mathbf{x})\gamma((\mathbf{x}-\boldsymbol{\mu})\mathbf{S}^{-1})d\mathbf{x} \\
&= \int_{\mathbb{R}^n} (i\boldsymbol{\omega})^{\boldsymbol{\alpha}}\mathcal{F}(\boldsymbol{\omega})\overline{\det(\mathbf{S})\mathcal{G}(\boldsymbol{\omega}\mathbf{S})e^{-i\boldsymbol{\omega}\boldsymbol{\mu}\mathbf{S}^{-1}}}d\boldsymbol{\omega} \\
&= \int_{\mathbb{R}^n} (i\boldsymbol{\omega})^{\boldsymbol{\alpha}}\mathcal{F}(\boldsymbol{\omega})\overline{\mathcal{P}(\boldsymbol{\omega})}d\boldsymbol{\omega}.
\end{aligned}$$

where $\mathcal{P}$ is the Fourier transform of the Gaussian measure $\mathcal{N}(\mu,\sigma^2)$ given by $\overline{\mathcal{P}(\boldsymbol{\omega})} = \det(\mathbf{S})\mathcal{G}(\boldsymbol{\omega})e^{-i\boldsymbol{\omega}\boldsymbol{\mu}\mathbf{S}^{-1}}$ (where $\mathcal{G}$ is the Fourier transform of the standard Gaussian measure γ) and $\mathcal{F}$ is the Fourier transform of f. This stems from the fact that the Fourier transform of the $\boldsymbol{\alpha}^{\text{th}}$ derivative of a function is simply $(i\boldsymbol{\omega})^{\boldsymbol{\alpha}}\mathcal{F}(\boldsymbol{\omega})$

From this we obtain that

$$\begin{aligned}
\text{Var(f)} &= \sum_{|\boldsymbol{\alpha}|\geq 1} \frac{(\text{diag}(\mathbf{S}))^{2\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}\left|\int_{\mathbb{R}^n}(i\boldsymbol{\omega})^{\boldsymbol{\alpha}}\mathcal{F}(\boldsymbol{\omega})\overline{\mathcal{P}(\boldsymbol{\omega})}d\boldsymbol{\omega}\right|^2 \\
&= \sum_{|\boldsymbol{\alpha}|\geq 1} \frac{(\text{diag}(\mathbf{S}))^{2\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}\left|\int_{\mathbb{R}^n}(\boldsymbol{\omega})^{\boldsymbol{\alpha}}\mathcal{F}(\boldsymbol{\omega})\overline{\mathcal{P}(\boldsymbol{\omega})}d\boldsymbol{\omega}\right|^2, \boldsymbol{\alpha}\in\mathbb{N}^n.
\end{aligned}$$

□

A.4 Proof of Proposition 4

Proof. We begin with a proof for a univariate Gaussian space. We once again use the definition of the Hermite polynomials in terms of the divergence operator. Note that $H_\alpha = \delta H_{\alpha-1}$. The variance of a function $f \in L_2(\mathbb{R},\mathcal{N}(\mu,\sigma^2))$ can be expressed as the sum of Hermite coefficients squared:

$$\text{Var}(f) = \sum_{\alpha=1}^{\infty}\frac{1}{\alpha!}|\hat{f}(\alpha)|^2 = \sum_{\alpha=1}^{\infty}\frac{1}{\alpha!}\left|\mathbb{E}_{\gamma(\hat{x})}\left[f(\sigma\hat{x}+\mu)H_\alpha(\hat{x})\right]\right|^2. \tag{A.4}$$

Using the definition of the divergence operator we have that:

$$\text{Var}(f) = \sum_{\alpha=1}^{\infty}\frac{1}{\alpha!}\left|\mathbb{E}_{\gamma(\hat{x})}\left[\sigma f^{(1)}(\sigma\hat{x}+\mu)H_{\alpha-1}(\hat{x})\right]\right|^2 \tag{A.5}$$

$$\leq \sum_{\alpha=1}^{\infty}\frac{1}{(\alpha-1)!}\left|\mathbb{E}_{\gamma(\hat{x})}\left[\sigma f^{(1)}(\sigma\hat{x}+\mu)H_{\alpha-1}(\hat{x})\right]\right|^2 \tag{A.6}$$

$$= \sum_{\alpha=0}^{\infty}\frac{\sigma^2}{\alpha!}\left|\mathbb{E}_{\gamma(\hat{x})}\left[f^{(1)}(\sigma\hat{x}+\mu)H_{\alpha}(\hat{x})\right]\right|^2. \tag{A.7}$$

As $f^{(1)}$ is a member of the Gaussian space, the last equation is simply the dot product of $f^{(1)}$ with itself, weighted by σ^2. Thus:

$$\text{Var}(f) = \sigma^2\langle f^{(1)}, f^{(1)}\rangle. \tag{A.8}$$

By definition of the Lipschitz constant:

$$L \leq \sup_{x,x'\in\mathbb{R}^\times, x\neq x'}\frac{|f(x)-f(x')|}{\|x-x'\|}.$$

We then have

$$\text{Var}(f) \leq \sigma^2 L^2. \tag{A.9}$$

For the multivariate case, we work in reverse. We know that $L^2 \geq \|Df\|^2$ where Df is the Jacbobian of the function with respect to the input such that

$$\left[\|Df\|^2\right] = \sum_{|\boldsymbol{\alpha}|=1} < f^{(\boldsymbol{\alpha})}, f^{(\boldsymbol{\alpha})} >$$

As by assumption each $f^{(\boldsymbol{\alpha})}, |\boldsymbol{\alpha}| = 1$ is a member of the Gaussian space we have that

$$\|Df\|^2 = \sum_{|\boldsymbol{\alpha}|=1} \sum_{\beta \in \mathbb{N}^n} \frac{1}{\beta!} \left| \mathbb{E}_{\gamma(\hat{\mathbf{x}})} \left[f^{(\boldsymbol{\alpha})}(\mathbf{S}\hat{\mathbf{x}} + \boldsymbol{\mu}) \mathcal{H}_\beta(\hat{\mathbf{x}}) \right] \right|^2 \tag{A.10}$$

$$= \sum_{|\boldsymbol{\alpha}|=1} \sum_{\beta \in \mathbb{N}^n} \frac{1}{\beta!} \left| \mathbb{E}_{\gamma(\hat{\mathbf{x}})} \left[\mathbf{S}^{-1} f(\mathbf{S}\hat{\mathbf{x}} + \boldsymbol{\mu}) \delta^{\boldsymbol{\alpha}} \mathcal{H}_\beta(\hat{\mathbf{x}}) \right] \right|^2 \tag{A.11}$$

$$= \sum_{|\boldsymbol{\alpha}|=1} \sum_{\beta \in \mathbb{N}^n} \frac{1}{\beta!} \left| \mathbb{E}_{\gamma(\hat{\mathbf{x}})} \left[\mathbf{S}^{-1} f(\mathbf{S}\hat{\mathbf{x}} + \boldsymbol{\mu}) \mathcal{H}_{\beta + \boldsymbol{\alpha}}(\hat{\mathbf{x}}) \right] \right|^2 \tag{A.12}$$

$$\geq \sum_{|\beta| \geq 1} \frac{1}{\beta!} \left| \mathbb{E}_{\gamma(\hat{\mathbf{x}})} \left[\mathbf{S}^{-1} f(\mathbf{S}\hat{\mathbf{x}} + \boldsymbol{\mu}) \mathcal{H}_\beta(\hat{\mathbf{x}}) \right] \right|^2, \beta \in \mathbb{N}^n. \tag{A.13}$$

This last term is simply $\text{Var}(f)/\|\mathbf{S}\|^2$ (see the proof of Theorem 3 for why this is). Thus we have that $L^2\|\mathbf{S}\|^2 \geq \text{Var}(f)$. This concludes the proof.

□

B Multivariate Hermite Coefficients for a Gaussian measure with full Covariance

Let our Gaussian space be equipped with Gaussian measure $\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$, where $\mathbf{C}$ is a full covariance matrix. By definition $\mathbf{C}$ is symmetric and positive definite, thus there exists an orthogonal matrix $\mathbf{O}$ such that:

$$\mathbf{O}\mathbf{C}\mathbf{O}^T = \mathbf{D},$$

where $\mathbf{D}$ is some diagonal matrix. Let $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$, we know that:

$$\hat{\mathbf{x}} = (\mathbf{x} - \boldsymbol{\mu})\mathbf{O}^T\mathbf{D}^{-\frac{1}{2}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

By substitution of variables for the Hermite polynomials in the case of a diagonal covariance, we have that the Hermite polynomials for the measure $\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ are given by:

$$\mathcal{H}_{\boldsymbol{\alpha}}((\mathbf{x} - \boldsymbol{\mu})\mathbf{O}^T\mathbf{D}^{-\frac{1}{2}}) = \mathcal{H}_{\boldsymbol{\alpha}}(\hat{\mathbf{x}}) = \prod_i H_{\alpha_i}(\hat{x}_i).$$

C Implementation Details

The architecture of VAEs with dense layers is presented in text. For convolutional networks the encoder layers had the following number of filters in order: $\{64, 64, 128, 128, 512\}$.

The mean and variance of the amortised posteriors are the output of dense layers acting on the output of the purely convolutional network, where the number of neurons in these layers is equal to the dimensionality of the latent space $\mathcal{Z}$.

Similarly, for the decoders ($p(\mathbf{x}|\mathbf{z})$) of all our models we also used purely convolutional networks with 6 deconvolutional layers. The layers had the following number of filters in order: $\{512, 128, 128, 64, 64, 3\}$. The mean of the likelihood $p(\mathbf{x}|\cdot)$ was directly encoded by the final de-convolutional layer. The variance of the decoder was fixed to 0.1.

To train models we used ADAM (Kingma and Lei Ba, 2015) with default parameters, a learning rate of 0.001, and a batch size of 1024. All data was preprocessed to fall on the interval -1 to 1.

All models were trained with an *Azure Cloud Standard NC6* machine with a single *NVIDIA Tesla K80*.

D Additional Results

D.1 Fourier Transform of Gaussian Measure

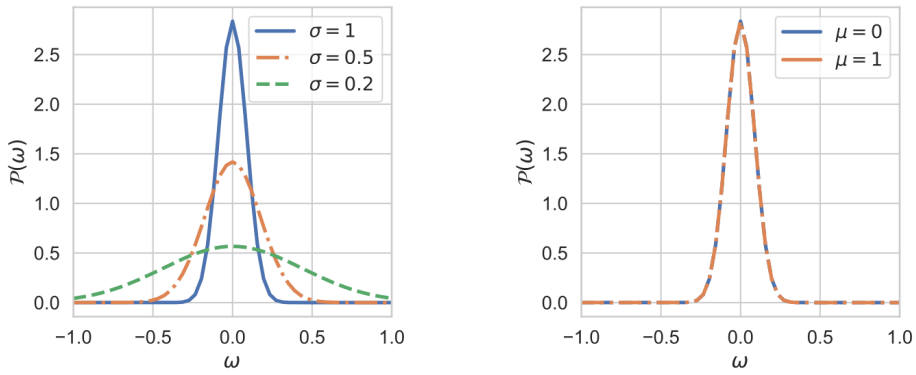


Figure D.5: [left] Fourier transform of the Gaussian measure $\mathcal{N}(0, \sigma^2)$ for varying σ. [right] Fourier transform of the Gaussian measure $\mathcal{N}(\mu, 1)$ for varying μ. Clearly as σ decreases the spectrum gains high frequencies, whereas altering μ has no effect.

D.2 Fourier Spectrum of Multivariate VAEs

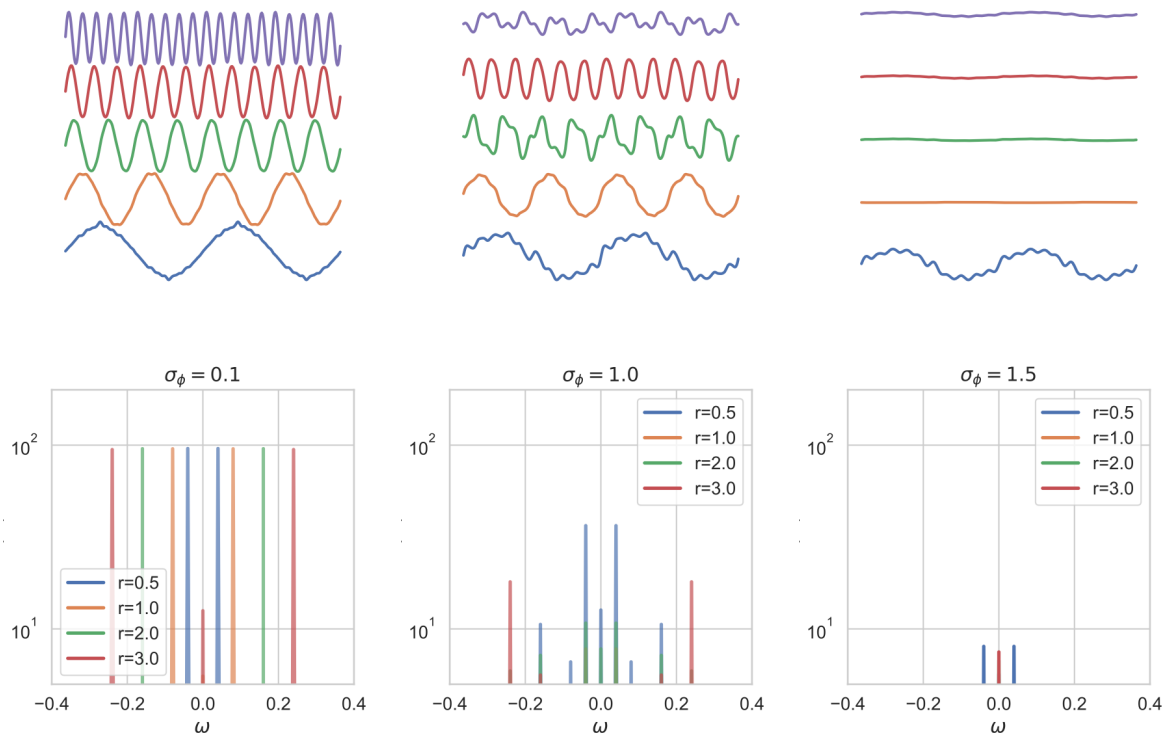


Figure D.6: We train VAEs with 3-dense layers (each with 256 units), with ReLU activation functions in their networks, on multivariate data consisting of 5 sinusoids $y = \sin(2\pi r t)$, $r \in \{0.5, 1.0, 2.0, 3.0, 5.0\}$, $t \in [-1, 1]$, with a 1 dimension latent space $\mathcal{Z}$. As before we fix the encoder σ_ϕ. **[top row]** We show plots of the regressed sinusoids for different values of σ_ϕ. **[bottom row]** Below we show an FFT for each regressed sinusoid. As σ_ϕ increases we clearly lose some of the mid to high level frequencies.

D.3 Gaussian Spaces and VAE Lipschitzness

We know that by Proposition 4 the variance of a function (that is once differentiable) provides a relatively tight lower bound on the Lipschitz constant of a neural network. This lower bound also increases as the variance of the Gaussian measure decreases. In the Supplement we demonstrate that this modulation of the encoder and decoder Lipschitzness is a purveyor of adversarial robustness for the VAE networks.

In previous sections we demonstrated that larger encoder variances induce smaller variance decoder functions, with lower frequency components. As the encoder variance $\boldsymbol{\sigma}_\phi^2(\mathbf{x})$ increases, the variance of the decoder function $\mathrm{Var}_{q_\phi(\mathbf{z}|\mathbf{x})}(g_\theta(\mathbf{z}))$ decreases, which by Proposition 4 will lead to a smaller lower bound on the Lipschitz constant of the decoder. Due to the computational costs of estimating the Lipschitz constants of networks, we restrict our empirical analysis here to fully-connected VAEs and use *layer-wise* LipSDP (Fazlyab et al., 2019) to obtain estimates. In Table D.1 we confirm that regulating the encoder variance allows for us to impose a soft constraint of the Lipschitzness of the VAE decoder. We also show that adding Gaussian noise on data, as motivated in the previous sections by its effect on $\mathrm{Var}(\mu_\phi)$, gives us finer control on the Lipschitz constant of the encoder.

Table D.1: The Lipschitz constants (L) of VAEs' **[left]** decoder networks (g_θ) when trained with fixed encoder scale (σ_ϕ) and **[right]** encoder networks (μ_ϕ) when trained with σ-scale Gaussian noise injection on inputs and fixed encoder variance σ_ϕ=0.5. We train fully-connected VAEs with the Sigmoid activation in their networks; on sinc($5t$) ($\dim(\mathcal{Z}) = 1$), on CelebA ($\dim(\mathcal{Z}) = 64$), and CIFAR10 ($\dim(\mathcal{Z}) = 64$). Each network has 3 dense layers each with 256 units. As σ_ϕ decreases the Lipschitz constant (L) increases, as predicted by Proposition 4. As σ decreases, the Lipschitz constant (L) of the encoder mean (μ_ϕ) increases. ($\pm$) is the std. dev. over 3 random seeds.

Data	$L(g_\theta;\sigma_\phi)$			$L(\mu_\phi;\sigma)$		
	$\sigma_\phi = 1.0$	$\sigma_\phi = 0.5$	$\sigma_\phi = 0.1$	$\sigma = 1.0$	$\sigma = 0.5$	$\sigma = 0.0$
sinc	2.2 ± 0.2	5.2 ± 0.3	17.9 ± 3.2	13.9 ± 2.7	24.6 ± 1.7	29.8 ± 2.2
CelebA(10^4)	7.5 ± 1.1	12.0 ± 0.5	13.7 ± 1.2	1.4 ± 0.1	1.6 ± 0.1	1.8 ± 0.1
CIFAR10(10^2)	17.9 ± 1.2	19.1 ± 1.2	27.3 ± 0.3	4.7 ± 0.2	5.6 ± 0.6	8.5 ± 0.8

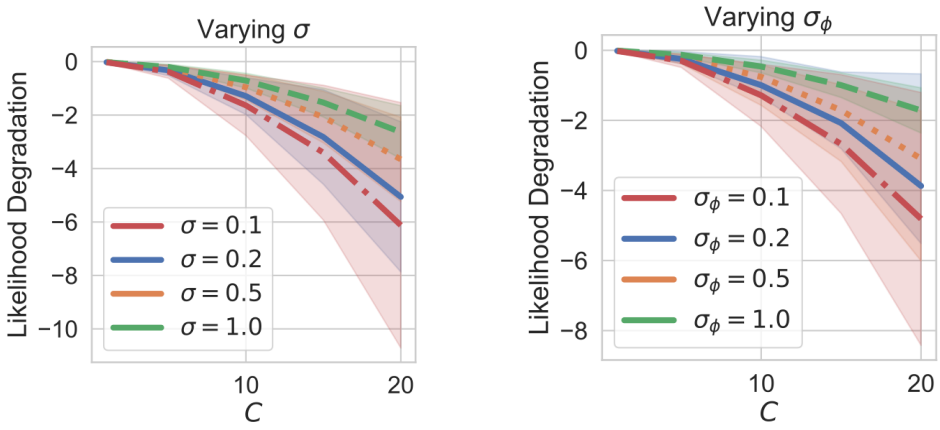


Figure D.7: Both plots show the relative log likelihood degradation resulting from a 'maximum-damage' (Camuto et al., 2020) adversarial attack against the maximum attack norm C. As σ, the variance of isotropic Gaussian noise added to data, and σ_ϕ, the fixed posterior variance, increase, the degradation lessens as C increases, highlighting that both these parameters improve robustness. Shading corresponds to the variance of the likelihood degradation over 25 points from CIFAR10.

Lipschitzness and Robustness Recent work shows that larger encoder variances and smaller encoder and decoder network Lipschitz constants are core contributors to the robustness of VAEs to adversarial attack (Camuto et al., 2021b; Barrett et al., 2021). Whereas this theory has viewed the network Lipschitz constants and the encoder variance as distinct parameters to control to attain robustness; our harmonic analysis shows that they are in fact intrinsically linked. This means that the encoder variance can serve as a single parameter on which we can act to improve the robustness of VAEs to adversarial attack, in that it also affects the Lipschitz

constant of the decoder. Further, adding noise to the VAE input data gives us similar control on the Lipschitz constant of the encoder mean μ_ϕ. As such, fixing and modulating both the encoder variance and the variance of the noise on data allows for the imposition of soft Lipschitz constraints on the VAE networks that improve adversarial robustness.

We consider an adversary trying to distort the input data to maximally disrupt a VAE's output, as in *maximum damage* attacks (Camuto et al., 2020; Barrett et al., 2021). The adversary maximises, with respect to a perturbation on data $\boldsymbol{\delta}_x$, constrained in norm by a constant C, the distance between the VAE reconstruction and data $\mathbf{x}$

$$\boldsymbol{\delta}_x^* = \underset{\|\boldsymbol{\delta}_x\|_2 \leq C}{\arg\max} \left(||g_\theta(\boldsymbol{\mu}_\phi(\mathbf{x} + \boldsymbol{\delta}_x) + \boldsymbol{\eta}\sigma_\phi(\mathbf{x} + \boldsymbol{\delta}_x)) - g_\theta(\boldsymbol{\mu}_\phi(\mathbf{x}))||_2 \right). \tag{D.1}$$

Given an embedding $\mathbf{z}^*$ formed from the mean encoding of $\mathbf{x} + \boldsymbol{\delta}_x$, we measure the likelihood of the original point $\mathbf{x}$ and quantify the degradation in model performance as the relative log likelihood degradation $(|\log p_\theta(\mathbf{x}|\mathbf{z}^*) - \log p_\theta(\mathbf{x}|\mathbf{z})|/\log p_\theta(\mathbf{x}|\mathbf{z}))$, where $\mathbf{z}$ is the embedding of $\mathbf{x}$.

Figure D.7 shows that as the variance of noise on data (σ) and the fixed encoder variance (σ_ϕ) increase, this degradation lessens for the VAEs trained on the CIFAR10 dataset in Table D.1, indicating less damaging attacks. In tandem, Table D.1 also shows that the increase of both these parameters decreases the Lipschitz constants of the encoder and decoder network for these VAEs. Thus we confirm the analysis above (and the recent theory on the adversarial robustness of VAEs (Camuto et al., 2021b; Barrett et al., 2021)), empirically demonstrating that both σ_ϕ and σ reduce the Lipschitz constant of both the encoder and decoder networks and simultaneously improve the robustness of VAEs trained on CIFAR10.