Spectral Smoothing Unveils Phase Transitions in Hierarchical Variational Autoencoders

Pervez, A.; Gavves, E.

Publication date
2021

Document Version
Final published version

Published in
Proceedings of Machine Learning Research

License
Other

Link to publication

Citation for published version (APA):
Pervez, A., & Gavves, E. (2021). Spectral Smoothing Unveils Phase Transitions in Hierarchical Variational Autoencoders. *Proceedings of Machine Learning Research*, *139*, 8536-8545. https://proceedings.mlr.press/v139/pervez21a.html

Spectral Smoothing Unveils Phase Transitions in Hierarchical Variational Autoencoders

Adeel Pervez [1] **Efstratios Gavves** [1]

Abstract

Variational autoencoders with deep stochastic hierarchies are known to suffer from the problem of posterior collapse, where the top layers fall back to the prior and become independent of input. We suggest that the hierarchical VAE objective explicitly includes the variance of the function parameterizing the mean and variance of the latent Gaussian distribution which itself is often a high variance function. Building on this we generalize VAE neural networks by incorporating a smoothing parameter motivated by Gaussian analysis to reduce higher frequency components and consequently the variance in parameterizing functions. We show this helps to solve the problem of posterior collapse. We further show that under such smoothing the VAE loss exhibits a phase transition, where the top layer KL divergence sharply drops to zero at a critical value of the smoothing parameter that is similar for the same model across datasets. We validate the phenomenon across model configurations and datasets.

1. Introduction

Variational autoencoders (VAE) (Kingma & Welling, 2014) are a popular latent variable model for unsupervised learning that simplifies learning by the introduction of a learned approximate posterior. Given data x and latent variables z, we specify the conditional distribution $p(x|z)$ by parameterizing the distribution parameters by a neural network. Since it is difficult to learn such a model directly, another conditional distribution $q(z|x)$ is introduced to approximate the posterior distribution. During learning the goal is to maximize the evidence lower bound (ELBO), which lower bounds the log likelihood, $\log p(x) \geq \mathbb{E}_{q(z|x)}\big[\log p(x|z) + \log p(z) - \log q(z|x)\big]$. In their simplest form, the generative model $p(x|z)$ and the approximate posterior $q(z|x)$ are Gaussian distributions optimized in unison.

A natural way to increase the modeling capacity of VAE is to incorporate a hierarchy of stochastic variables. Such models, however, turn out to be difficult to train and higher levels in the hierarchy tend to remain independent of input data – a problem termed *posterior collapse*. Posterior collapse in VAEs manifests itself by the latent distribution tending to fall back to the prior. With hierarchical VAEs the effect is found to be more pronounced in the top layers farther from the output. For the purpose of the paper and for clarity of exposition, we focus on the simplest extension of hierarchical variational autoencoders where stochastic layers are stacked serially on top of each other (Burda et al., 2016; Sønderby et al., 2016) , $p(x, z) = p(x|z_1)p(z_L)\prod_{i=1}^{L-1} p(z_i|z_{i+1})$ and $q(z|x) = q(z_1|x)\prod_{i=1}^{L-1} q(z_{i+1}|z_i)$. The intermediate distributions in this model are commonly taken to be Gaussian distributions parameterized by neural network functions, so that $p(z_i|z_{i+1}) = \mathcal{N}(z_i|\mu(z_{i+1}), \sigma(z_{i+1}))$, where $\mu(z), \sigma(z)$ are neural networks computing the mean and variance of the Gaussian distribution. We refer to them as *vanilla hierarchical variational autoencoders*. For each stochastic layer in this model there is a corresponding KL divergence term in the objective given by

$$\mathbb{E}[KL(q(z_i|z_{i-1})||p(z_i|z_{i+1})]. \tag{1}$$

As described later, expression 1 can be easily decomposed to show an explicit dependence on the *variance* of the parameterizing functions $\mu(z_i), \sigma(z_i)$ of the intermediate Gaussian distribution. We further show the KL divergence term to be closely related to the harmonics of the parameterizing function. For complex parameterizing functions the KL divergence term has large high frequency components (and thus high variance) which leads to unstable training causing posterior collapse.

Building on this, we suggest a method for training the simplest hierarchical extension of VAE that avoids the problem of posterior collapse without introducing further architectural complexity (Maaløe et al., 2019; Sønderby et al., 2016). Given a hierarchical variational autoencoder, our training method incorporates a smoothing parameter (we denote this by ρ) in the neural network functions used to parameterize

[1] QUVA Lab, Informatics Institute, University of Amsterdam, The Netherlands. Correspondence to: Adeel Pervez <a.a.pervez@uva.nl>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

the intermediate latent distributions. The smoothing is done such that expected values are preserved, the higher frequencies are attenuated and the variance is reduced. Next, the gradients computed with the smooth functions are used to train the original hierarchical variational autoencoder.

For the construction of the smoothing transformations for VAEs with Gaussian latent spaces we use of ideas from the analysis of Gaussian spaces. We analyze the stochastic functions in vanilla hierarchical VAEs as Hermite expansions on Gaussian spaces (Janson et al., 1997). The Ornstein-Uhlenbeck (OU) semigroup from Gaussian analysis is a set of operators that we show to smoothly interpolate between a random variable and its expectation. The OU semigroup provides the appropriate set of smoothing operators which enable us to control variance and avoid posterior collapse.

We further show that by smoothing the intermediate parameterizing functions $\mu(z), \sigma(z)$ in the proposed manner, the KL divergence of the top layer sees a sudden sharp drop toward zero as the amount of smoothing is decreased. This behaviour is retained when we evaluate the KL divergence on the original *unsmoothed* variational autoencoder. This behaviour is reminiscent of phase transitions from statistical mechanics. Our experiments suggest that the phenomenon is general across datasets and commonly used architectures. Furthermore, the *critical value* of the smoothing parameter ρ at which the transition occurs is fixed for a given model configuration and varies with stochastic depth and width.

We make the following contributions. First, we establish a connection between higher harmonics, variance, posterior collapse and phase transitions in hierarchical VAEs. Second, we show that by using the Ornstein-Uhlenbeck semigroup of operators on the generative stochastic functions in VAEs we reduce higher frequencies, and consequently the variance, mitigating posterior collapse. We extensively corroborate our findings experimentally and with simple architectures we obtain likelihoods in CIFAR-10 that are competitive to methods requiring complex architectural solutions.

2. Spectral Smoothing for Variational Autoencoders

2.1. Analysis on Gaussian spaces

The analysis of Gaussian spaces studies functions of Gaussian random variables. These are real-valued functions defined on $\mathbb{R}^n$ endowed with the Gaussian measure. Many functions employed in machine learning are instances of such functions: decoders for variational autoencoders, as is the case in this work, and generators for generative adversarial networks being two examples.

By way of summary, the main facts we use from this field are that a function on a Gaussian space can be expanded in an orthonormal basis, where the basis functions are the Hermite polynomials. This orthonormal expansion is akin to a Fourier transform in this space. The second fact is that the coefficients of such an expansion can be modified to reduce the variance of the expanded function by applying an operator from the Ornstein-Uhlenbeck semigroup of operators. Next, we give a brief introduction. For more details on Gaussian analysis we refer to (Janson et al., 1997).

Gaussian Spaces: Let $L^2(\mathbb{R}^n, \gamma)$ be the space of square integrable functions, $f : \mathbb{R}^n \to \mathbb{R}$, with the Gaussian measure $\gamma(z) = \prod_i \mathcal{N}(z_i|0,1)$. Given functions f, g in this space, the inner product is given by $\langle f, g\rangle = \mathbb{E}_{\gamma(z)}[f(z)g(z)]$.

Basis functions for $L^2(\mathbb{R}, \gamma)$: Taking the space of *univariate* functions $L^2(\mathbb{R}, \gamma)$, it is known that the polynomial functions $\phi_i(z) = z^i$ are a basis for this space. By a process of orthonormalization we obtain the *normalized* Hermite polynomial basis for this space. The first few Hermite polynomials are the following: $h_0(z) = 1, \quad h_1(z) = z, \quad h_2 = \frac{z^2-1}{\sqrt{2}}, \ldots$.

Basis functions for $L^2(\mathbb{R}^n, \gamma)$: Letting $\alpha \in \mathbb{N}^n$ be a multi-index, the basis functions for $L^2(\mathbb{R}^n, \gamma)$ are obtained by multiplying the univariate basis functions across dimension, $h_\alpha(z) = \prod_i h_{\alpha_i}(z_i)$.

Hermite expansion: A function in $L^2(\mathbb{R}^n, \gamma)$ can be expressed as $f = \sum_{\alpha\in\mathbb{N}^n} \hat{f}(\alpha)h_\alpha$, where $\hat{f}(\alpha)$ are the Hermite coefficients of f and are computed as $\hat{f}(\alpha) = \langle f, h_\alpha\rangle = \mathbb{E}_{\gamma(z)}[f(z)h_\alpha(z)]$. By the orthnormality of the basis functions, Plancherel's theorem connects the norm of f with the Hermite coefficients as $\langle f, f\rangle = \sum_\alpha \hat{f}(\alpha)^2$.

Ornstein-Uhlenbeck (OU) Semigroup: Given a parameter $\rho \in [0, 1]$ and a Gaussian variable z, we construct a correlated variable z' as $z' = \rho z + \sqrt{1-\rho^2} z_\omega$, where $z_\omega \sim \mathcal{N}(0, 1)$ is a random standard Gaussian sample. The OU semigroup is a set of operators, denoted U_ρ and parameterized by $\rho \in [0, 1]$. The action of U_ρ on f at z is to average the function values on correlated z's around z,

$$U_\rho f(z) = \mathbb{E}_{z'|z}[f(z')] = \mathbb{E}_{z_\omega}[f(\rho z + \sqrt{1-\rho^2} z_\omega)] \quad (2)$$

The action of the U_ρ operators on the Hermite expansion of $f(z)$ is to decay Hermite coefficients according to their degree, $U_\rho f(z) = \sum_{\alpha\in\mathbb{N}^n} \rho^{|\alpha|}\hat{f}(\alpha)h_\alpha$. where $|\alpha| = \sum_i \alpha_i$.

If z is reparameterized as $z = \sigma\epsilon_1 + \mu$, the correlated OU sample is given by $z' = \sigma(\rho\epsilon_1 + \sqrt{1-\rho^2}\epsilon_2) + \mu$, where ϵ_1, ϵ_2 are standard Gaussian variables. This can also be expressed in terms of z as

$$z' = \rho z + (1-\rho)\mu + \sigma\sqrt{1-\rho^2}\epsilon_2, \quad (3)$$

2.2. Hermite expansions for VAEs

We propose a new training procedure for the vanilla hierarchical variational autoencoder that builds upon Hermite expansions of Gaussian functions and properties of the OU semigroup. In the context of hierarchical variational autoencoders, the Gaussian function f is the generative model $\mu_i(z_{i+1})$ and $\sigma_i(z_{i+1})$ that receives as inputs the latent variable z_{i+1} to return the Gaussian latent variable of the next layer, $z_i \sim \mathcal{N}(\mu_i(z_{i+1}), \sigma_i(z_{i+1}))$.

We make use of the following properties of the OU semigroup to construct Gaussian functions of lower variance.

The first property, described in the following two propositions, is that the OU semigroup of operators interpolates between a random variable ($\rho = 1$) and its expectation ($\rho = 0$). Parameter ρ controls the extent of interpolation.

Proposition 1 *The operators U_ρ retain the expected value of the operated function, $\mathbb{E}[f] = \mathbb{E}[U_\rho f]$.*

Proposition 2 *The operators U_ρ interpolate between a random variable and its expectation. In particular, as $\rho \to 1$, $U_\rho f = f$. and as $\rho \to 0$, $U_\rho f = \mathbb{E}[f]$*

The second property is that the new random variable $U_\rho f(z)$ has lower variance compared to the original variable $f(z)$ and is in general a smoother function than $f(z)$.

The smoothing properties of the operator U_ρ can be understood by examining the Hermite expansion of $U_\rho f$. First, we note that we can express the expectation and variance of a function f in terms of its Hermite coefficients. Specifically, $\mathbb{E}[f] = \hat{f}(0)$ and $\mathrm{Var}(f) = \mathbb{E}[(f - \mathbb{E}[f])^2] = \mathbb{E}[(f - \hat{f}(0))^2] = \sum_{\alpha:|\alpha|>0} \hat{f}(\alpha)^2$, which follows from Plancherel's theorem. Replacing f with $U_\rho f$ and using the Hermite expansion of $U_\rho f$ from equation 2, the mean remains the same, $\mathbb{E}[U_\rho f] = \rho^0 \hat{f}(0) = \hat{f}(0)$, and variance reduces like

$$\begin{aligned}\mathrm{Var}[U_\rho f] &= \mathbb{E}[(U_\rho f - \mathbb{E}[f])^2] = \mathbb{E}[(f - \hat{f}(0))^2] \\ &= \sum_{\alpha:|\alpha|>0} \rho^{2|\alpha|} \hat{f}(\alpha)^2. \qquad (4)\end{aligned}$$

Equation equation 4 indicates that the contribution to the variance by $\hat{f}(\alpha)$ decays by an amount $\rho^{2|\alpha|}$ when $\rho \in (0, 1)$. By using the Ornstein-Uhlenbeck semigroup U_ρ on the generative model $\mu_i(z_{i+1})$ and $\sigma_i(z_{i+1})$ we obtain smoother VAE functions of lower variance by dampening higher frequency components.

Algorithm. In essence, OU-smoothed variational autoencoders are similar to variational autoencoders, save for applying the OU semigroup to the latent distributions $p(z_i|z_{i+1})$ of the generator to compute gradients during *training* only. Specifically, we apply these operators to the functions parameterizing the mean and variance of the latent Gaussian distributions. For each distribution $p(z_i|z_{i+1})$ we substitute $\mathcal{N}(z_i|\mu_i(z_{i+1}), \sigma_i(z_{i+1}))$ with $\mathcal{N}(z_i|U_\rho\mu_i(z_{i+1}), U_\rho\sigma_i(z_{i+1}))$. The new functions result in latent distributions with parameters with lower variance but the same expected value relative to the conditional input latent distribution. In practice, we compute $U_\rho\mu_i(z_{i+1})$ and $U_\rho\sigma_i(z_{i+1})$ by Monte Carlo averaging. As for a function f, $U_\rho f = \mathbb{E}_{z'|z}[f(z')]$, where z' are the correlated samples, we estimate the expectation by Monte Carlo averaging over z'. Experiments show that 5 to 10 samples suffice.

It is important to emphasize that the substitution of the lower variance functions for parameterizing the distributions is *only done when computing gradients during training*. All evaluations, training or test, are still done on the original hierarchical variational autoencoder model. Thus, the new training procedure has an additional computational cost only for the intermediate distributions in the generator, proportional to the number of correlated samples during training.

Alternative parameterization. We can also apply the OU semigroup to different functions in the variational autoencoder. Applying U_ρ to the ratio of the mean and variance functions, $U_\rho \frac{\mu_i}{\sigma_i}(z_{i+1})$, for instance, provides a convenient form to analyze the KL divergence in hierarchical VAEs, as we shall see next. Moreover, the OU semigroup operators can be applied on approximate posterior functions. Experimentally, we observed little benefit and we left that variant out of the comparisons.

Complexity. We apply the OU sampling operation *only in the intermediate stochastic layers in the generator network* of the smoothed VAE. We *do not apply* OU sampling in the inference network. Also, we *do not apply* OU sampling in the last stochastic layer of the decoder computing $p(x|z_1)$, typically upsampled to match image dimensions in deep VAEs for images. Overall, OU-smoothed VAEs are more memory and parameter efficient than models like IWAE, LVAE and BIVA, as also shown in our experiments (see table 10).

3. KL Divergence Analysis

By casting the generative functions of variational autoencoders with Gaussian latent variables on Hermite basis, we can analyze the KL divergence from equation 1 in terms of the hierarchical VAE objective in terms of bias-variance trade-off. The bias-variance trade-off arises by rewriting the KL divergence in terms of the variance of the functions $\mu(z), \sigma(z)$ parameterizing the intermediate distributions. This variance is related to spectral complexity. Reducing spectral complexity leads to a reduction in variance.

For clarity, we start with two stochastic layers, z_1 and z_2, and also assume that the standard deviation for p is fixed and independent of z_2, that is $\sigma(z_2) = \sigma_p$. Without loss of generality we think of z_1 as a scalar. The general result for multivariate z_1 follows from summing for all dimensions.

We focus on the terms in the KL divergence including the conditionals $p(z_1|z_2)$, which are prone to causing collapse,

$$\mathbb{E}_{q(z_1,z_2|x)}[\log p(z_1|z_2)] = \mathbb{E}_{q(z_1|x)}\mathbb{E}_{q(z_2|z_1)}[\log p(z_1|z_2)], \tag{5}$$

where $p(z_1|z_2) = \mathcal{N}(z_1|\mu_p(z_2), \sigma_p)$ and $q(z_1|x) = \mathcal{N}(z_1|\mu_q(x), \sigma_q(x))$.

For Gaussian p we write equation 5 as $\mathbb{E}_{q(z_1,z_2|x)}[-\log\sqrt{2\pi\sigma_p^2} - \frac{1}{2\sigma_p^2}(z_1 - \mu_p(z_2))^2]$. From the inner term $\mathbb{E}_{q(z_1,z_2|x)}[(z_1 - \mu_p(z_2))^2]$ we focus on the quadratic $\mu_p(z_2)^2$ which is expanded as

$$\frac{1}{2\sigma_p^2}\mathbb{E}_{q(z_1,z_2|x)}[\mu_p(z_2)^2] = \frac{1}{2\sigma_p^2}\mathbb{E}_{q(z_1|x)}[\mathbb{E}[\mu_p(z_2)]^2 + \mathrm{Var}(\mu_p)] \tag{6}$$

By Plancherel's theorem we have

$$\frac{1}{2\sigma_p^2}\mathbb{E}_{q(z_1,z_2|x)}[\mu_p(z_2)^2] = \frac{1}{2\sigma_p^2}\mathbb{E}_{q(z_1|x)}\left[\hat{\mu_p}(0)^2 + \sum_{\alpha:|\alpha|>0}\hat{\mu_p}(\alpha)^2\right]. \tag{7}$$

That is, for σ_p independent of z_2 the KL divergence term in the ELBO contains the variance of the parameterizing function $\mu_p(z_2)$.

In our proposal we replace $\mu_p(z_2)$ by $U_\rho[\mu_p(z_2)]$ and the right side of equation 7 becomes

$$\frac{1}{2\sigma_p^2}\mathbb{E}_{q(z_1|x)}[\mathbb{E}[U_\rho\mu_p]^2 + \mathrm{Var}(U_\rho\mu_p)] = \frac{1}{2\sigma_p^2}\mathbb{E}_{q(z_1|x)}\left[\hat{\mu_p}(0)^2 + \sum_{\alpha:|\alpha|>0}\rho^{2|\alpha|}\hat{\mu_p}(\alpha)^2\right], \tag{8}$$

since $\mathbb{E}[U_\rho f] = \mathbb{E}[f]$. The new variance is of order $O(\rho^2)$. Comparing this objective with the original VAE objective, for the second term we get a bias proportional to the difference of the variance

$$\text{bias} = \frac{1}{2\sigma_p^2}(\mathrm{Var}(\mu_p) - \mathrm{Var}(U_\rho\mu_p)).$$

Comparing equations 7 and 8 we see the bias to be $O(1-\rho^2)$. In figure 4 in the appendix we show the reduction in gradient variance with OU smoothing compared to regular VAEs.

More than 2 layers, σ_p dependent on z_2, choice of ρ. The analysis above assumes that σ_p is independent of z_2. For σ_p dependent on z_2 we can repeat the analysis for the variance of $(z_1 - \mu(z_2))/\sigma(z_2)$ and for the ratio $\mu(z_2)/\sigma(z_2)$ by expanding the square and considering the terms separately.

During training the decoder layers $p(z_i|z_{i+1})$ use samples from the respective encoder layer $q(z_{i+1}|z_i)$ and not the previous decoder layers $p(z_{i+1}|z_{i+2})$. The bias-variance decomposition when smoothing $p(z_{i+1}|z_{i+2})$ does not depend on the bias-variance decomposition of the smoothing on $p(z_i|z_{i+1})$. Thus, the analysis is easy to derive when cascading multiple stochastic layers ($L > 2$).

The above analysis introduces a bias-variance trade-off. Since we have a bias-variance trade-off, by throttling ρ we are now in position to exchange some variance for some bias. As a result, we can now train robustly and efficiently deep stochastic architectures, even with very simple neural network architectures. Experiments indeed show a consistent behavior and phase transitioning as we change ρ. Specifically, experiments corroborate that there exists a single value of ρ for which models consistently undergoes posterior collapse for different datasets given the same architecture. Deriving this critical ρ that prevents collapse is an interesting direction for future work.

4. Related Work

To increase the stochastic depth of VAEs (Kingma & Welling, 2014; Rezende et al., 2014), Sønderby et al. (2016) propose Ladder VAEs. With an architecture that shares a top-down dependency between the encoder and the decoder, Ladder VAEs allow for interactions between the bottom-up and top-down signals and enable training with several layers deep VAEs. Extending Ladder VAEs, Maaløe et al. (2019) proposed the bidirectional-inference VAEs, adding skip connections in the generative model and a bidirectional stochastic path in the inference model.

Bowman et al. (2016) observed that the latent distribution collapses to the prior in deep stochastic hierarchies – a phenomenon now called *posterior collapse*. Posterior collapse appears in different contexts including images or text, and is strongly associated with the presence of powerful decoders, be it LSTMs (Bowman et al., 2016) for text or strong autoregressive models for images (Oord et al., 2016), where although the model may produce good reconstructions, it does not learn a meaningful generative distribution. A prevalent hypothesis behind posterior collapse is that when the decoder is strong enough to generate very low cross entropy losses, the optimization may find it easier to simply set the KL divergence term to 0 to minimize the ELBO (Bowman et al., 2016). Making an association with probabilistic PCA, Lucas et al. (2019b) hypothesize that posterior collapse is caused by local optima in the optimization landscape due to high variance, even without powerful decoders. High variance was identified as a potential culprit also by (Sønderby et al., 2016) for posterior collapse.

Several proposals have attempted to address posterior collapse. Bowman et al. (2016); Higgins et al. (2017); Sønderby et al. (2016); Maaløe et al. (2016) anneal the KL divergence between the approximate posterior to the prior from 0 to 1. This solution does not optimize the original ELBO formulation and is shown (Yang et al., 2017; Chen et al., 2016) to cause instabilities, especially with large datasets and complex decoders. Kingma et al. (2016) introduce the concept of *free bits* forcing a minimum KL divergence. Razavi et al. (2019) proposed δ-VAEs, which constrain the latent distribution to have a minimum distance to the prior. He et al. (2019) monitor the mutual information between the latent and the observed variable to aggressively optimize the inference model before every model update. (Burda et al., 2016) suggest that using a tighter multi-sample ELBO can help alleviate collapse to some extent.

While some (Bowman et al., 2016; Sønderby et al., 2016) suggested a connection between posterior collapse and variance reduction, no real solution using variance reduction has been proposed. One reason may be the low variance the reparameterization trick (Kingma & Welling, 2014; Rezende et al., 2014) already offers. Empirically, while the reparametrization is successful with producing low variance forward and backward propagations in shallow models, it has not been enough for deeper and wider ones. Another reason suggested by the approach of this paper is that variance appears as a side effect of spectral complexity. A discrete version of the OU semigroup is used in Pervez et al. (2020) for gradients for binary latent variable models.

To reduce the variance in reparameterization gradients, Roeder et al. (2017) suggest removing a mean zero score function term from the total derivative while Miller et al. (2017) build a control variate using a linear approximation for variance reduction. Burda et al. (2016) propose importance weighted gradients and Tucker et al. (2018) extend Miller et al. (2017) to multiple samples to obtain an estimator with improved signal-to-noise ratio. Other approaches to increase the power of VAE models include normalizing flows (Rezende & Mohamed, 2015), better posterior or prior distributions (Tomczak & Welling, 2016), adding autoregressive components (Gulrajani et al., 2017) or a combination of both (Kingma et al., 2016).

We theoretically argue and empirically validate that damping higher frequency components, thus lowering variance, allows for training deeper latent hierarchies while addressing posterior collapse. We rely on tools from the field of analysis on Gaussian spaces, amenable to the analysis of stochastic processes (Janson et al., 1997).

5. Experiments

We perform an extensive array of evaluations with state-of-the-art benchmarks, methods, and architectures. Unless stated otherwise, we use the same architectures in the respective comparisons for VAE with and without OU-smoothing.

Our primary focus is posterior collapse, a fundamental problem, linked to high variance when stacking multiple stochastic layers (Lucas et al., 2019a). We investigate OU-smoothed VAEs in the context of posterior collapse and compare with methods that help with it. Then, we evaluate OU-smoothed VAEs on binary MNIST, OMNIGLOT and CIFAR-10 with various convolutional and MLP architectures. We compute validation ELBOs with importance-weighted samples (Burda et al., 2016) ($\mathcal{L}_{100}$ with 100 and $\mathcal{L}_{5000}$ with 5000 samples).

5.1. Posterior collapse in Hierarchical VAEs

Posterior collapse happens when the approximate posterior in the VAE falls back to the prior, yielding extremely low KL divergence and “dying” stochastic neurons showing consistently no activation no matter the input. With posterior collapse the model effectively learns to ignore the encoder or some portion thereof. Posterior collapse is observed in models with powerful generators, such as autoregressive models, and with models with deep stochastic hierarchies. In hierarchical VAEs and for higher layers posterior collapse is not simply a form of feature selection, as models suffering from “dying” neurons also show bad validation ELBO, see table 8 in the appendix.

We test OU-smoothed VAEs for posterior collapse on basic MLP network architectures, using on the static and dynamically binarized MNIST against various standard methods for mitigating posterior collapse. Following (Burda et al., 2016), we say a neuron is active when its activity variance across a batch of inputs is more than 0.01.

Dynamic and Static MNIST. For dynamically binarized MNIST we choose two models: the first is a 4 stochastic layer model with 64,32,16,8 latent variables. Between any two stochastic layers we add two deterministic layers of 512,256,128,64 units respectively from bottom to top. The second model has 4 stochastic layers VAE with 40 units per stochastic layer and 2 layers of 200 units per stochastic layer. For static MNIST we only use the second model described above. All models have a simple stacked architecture with no skip connections. We train the VAE with the standard training method and OU-smoothed VAEs with $\rho \in \{0.8, 0.9\}$.

We report the validation and the KL divergence curves of the top stochastic layer, $KL(q(z_4|z_3)||\mathcal{N}(0,1)$, in figure 2. For the vanilla VAE the approximate posterior collapses imme-

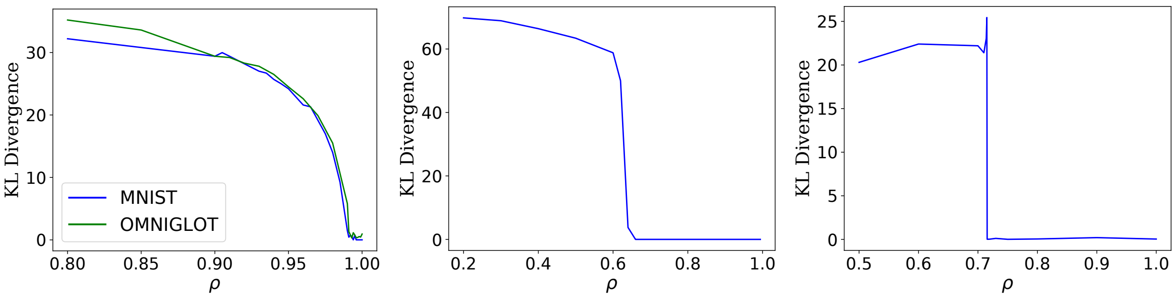

Figure 1. VAE KL divergence undergoes as a phase transition as the model as we decrease the amount of smoothing to recover the original model at $\rho = 1$. The figure shows top layer KL divergence *vs.* ρ, where each point is a different model after 100000 steps of training with a fixed ρ. The plots show the KL divergence values for the original unsmoothed model after training on the smoothed version. The left figure shows this for same 4 stochastic (40 units per layer) layer MLP model on MNIST and OMNIGLOT. It can be seen that the critical value where the KL divergence drops toward 0 is the same for both datasets on this model. The next two figures show phase transition for a 3 stochastic layer (200 units per layer) MLP model on OMNIGLOT (middle) and a convolutional model (8x7x7 latent dimension) on MNIST (right) showing that the phase transition becomes especially pronounced with the width of the latent layer.

diately and we obtain a poor validation ELBO. In contrast, OU smoothing avoids posterior collapse by maintaining a good KL divergence and a better validation ELBO.

It is worth noting that while posterior collapse is more visible in deeper stochastic models, OU smoothing can also be applied to models with a single stochastic layer (see table 7 in the appendix).

Further comparisons. We compare against other methods designed to mitigate posterior collapse: KL annealing (Kingma et al., 2016), free bits (Kingma et al., 2016) and importance weighted objectives (Burda et al., 2016) with the same architectures as above. For KL annealing the annealing coefficient is set to 0 for the first 10,000 steps and is linearly annealed to 1 over the next 500,000 steps. For free bits, we apply the same free bits value to each stochastic layer, as recommended in IAF (Kingma et al., 2016). The free bits values are chosen from $\{0.5, 1.0, 2.0, 3.0\}$. We find that training slows down considerably when using free bits and values of free bits of ≥ 4.0 made training unstable.

We show results on dynamic MNIST in table 1, including the total KL divergence, the top layer KL divergence as well as the number of active units in each of the 4 layers. KL annealing and free bits help in mitigating posterior collapse for shallow hierarchies of stochastic variables, annealing being more effective. Both methods lead to more active units, especially in the lower levels, and a somewhat larger top layer KL divergence than a standard VAE. However these techniques are not very effective at overcoming collapse in deeper hierarchies, which is the motivation of our work. Compared to IWAE, KL annealing and free bits, OU-smoothed VAEs maintain significant activity across the layers, a considerably higher KL divergence in the top layer, and in the end a better validation ELBO. Interestingly, unlike OU-smoothed VAEs, existing methods seem to not only lead to considerably fewer active units when moving higher in the hierarchy (further away from the output layer), but also the active units depend on the architecture (the 64-32-16-8 variant maintains more active units). This suggests that training dynamics and the architecture affect the extent of posterior collapse with standard methods, while having less of an effect with the proposed method.

We obtain similar results for static MNIST, see table 6 in the appendix. We also experimented with combining OU-smoothed VAEs with other posterior collapse mitigation techniques. KL annealing with the 4 layer OU-smoothed VAE improves validation ELBO, although with more complex architectures or complex datasets like CIFAR-10, we observed little benefit.

5.1.1. Phase Transitions

We saw that attenuating the higher frequency components of parameterizing functions by smoothing is a justifiable mitigation against posterior collapse. Here we show that as the amount of smoothing is reduced (as ρ approaches 1) to recover the original VAE gradients, the top level KL divergence shows a sudden decline at a critical value of the smoothing parameter. The sharpness of this decline depends on the model configuration (stochastic layers, latent dimension). On the other hand, our experiments suggest that the sharpness of the decline is independent of dataset.

An example of this phenomenon is in figure 1 and more in figures 5 and 6 in the appendix. We show the top layer KL divergence after training for 100k steps with varying ρ for 3 different architectures. The phase transition becomes more pronounced with greater stochastic dimension. Figure 3 further shows how OU smoothing prevents collapse across a spectrum of stochastic depths and widths while maintaining good validation performance.

Table 1. Posterior collapse on dynamic MNIST. The table shows top layer KL divergence and active units (top-to-bottom) for various method on the 4 stochastic layer models. The 64-32-16-8 latent dimension models have two layers of 512, 256, 128, 64 hidden units in each stochastic layer respectively. The 40-40-40-40 latent dimension models have two layers of 200 units per stochastic layer. '+KL' indicates KL annealing. All models were trained for 1M steps with the same hyperparameters.

Model	V. ELBO ($\mathcal{L}_{100}$)	Reconstruction	KLD	Top KLD	Active Units	Relative Activity (%)
IWAE (64-32-16-8)	-84.46	-65.3	23.98	4.88	64-30-15-3	100-94-94-38
IWAE (40-40-40-40)	-84.63	-65.5	23.98	1.18	40-37-4-0	100-93-10-0
VAE+KL (64-32-16-8)	-84.6	-60.3	28.8	6.2	49-25-11-6	79-78-69-75
VAE+KL (40-40-40-40)	-84.7	-60.8	28.07	1.13	40-15-6-1	100-38-15-2.5
VAE+Freebits (64-32-16-8)	-85.5	-64.2	25.1	3.8	21-9-4-2	33-28-25-25
VAE+Freebits (40-40-40-40)	-86.0	-65.8	23.6	2.46	18-8-2-1	45-20-5-2.5
OU-VAE ($\rho = 0.95$) (64-32-16-8)	-81.6	-59.78	26.1	8.99	54-32-16-8	84-100-100-100
OU-VAE ($\rho = 0.9$) (64-32-16-8)	-81.7	-60.0	25.6	9.56	43-32-16-8	67-100-100-100
OU-VAE ($\rho = 0.95$) (40-40-40-40)	-84.4	-65.7	23.7	9.34	40-40-40-40	100-100-100-100

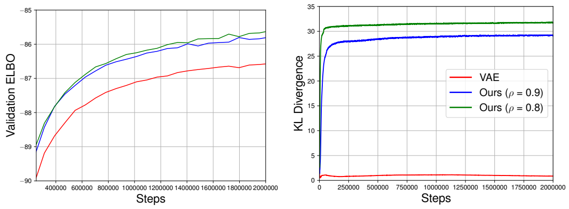

Figure 2. Training 4 layer VAE models on static MNIST with validation ELBO on the left and $KL(q(z_4|z_3)||\mathcal{N}(0,1))$ on the right. The VAE shows posterior collapse, while OU-VAE avoids it alongside improved validation ELBO.

5.2. Benchmark Comparisons

5.2.1. MNIST & OMNIGLOT

MLPs. We report test ELBO for MLPs for static and dynamic MNIST comparing with methods using their best reported settings. As OU-smoothed VAEs scale easily up to multiple stochastic layers, we experiment with 4 stochastic layers for static MNIST, and 4 and 5 stochastic layers for dynamic MNIST, with the same architecture details as earlier.

Table 2. Test ELBO on static MNIST with MLP

Model	ELBO
VAE (L=2)	-86.05
VAE (L=1)+NF (Rezende & Mohamed, 2015)	-85.10
IWAE (L=2) (Burda et al., 2016)	-85.32
VampPrior (L=2) (Tomczak & Welling, 2017)	-83.19
OU-VAE (L=4), $\mathcal{L}_{5000}$	-83.42

Table 3. Test ELBO on dynamic MNIST with MLP

Model	ELBO
Ladder VAE (L=5) (Sønderby et al., 2016)	-81.7
VampPrior (L=2) (Tomczak & Welling, 2017)	-81.24
OU-VAE (L=4), $\mathcal{L}_{5000}$	-81.2
OU-VAE (L=5), $\mathcal{L}_{5000}$	-81.1

On static and dynamic MNIST, including IWAE (Burda et al., 2016), VAE with normalizing flow (Rezende & Mohamed, 2015), VampPrior (Tomczak & Welling, 2017), and LVAE with 5 layers as well. We also compare with a VAE with two stochastic layers, as we were unable to either improve VAEs with more stochastic layers and we could not find in the literature a reference of a vanilla VAE deeper than 2 layers with better performance on this dataset. We show the results for static MNIST in table 2 and for dynamic MNIST in table 3. We observe that OU-smoothed VAEs outperform other methods in both settings while relying on relatively simple architectures.

Residual ConvNets. Last, we experiment with a more complex ResNet architecture on MNIST and OMNIGLOT and the same architecture structure as the MLP VAEs, with up to 4 stochastic layers and 5 ResNet convolutional blocks between stochastic layers (14 × 14 features maps). We do not employ stochastic skip connections between blocks. We report results in table 5 in the appendix. In both experiments we obtain competitive scores despite the simple architecture, reaching -96.08 validation ELBO on OMNIGLOT, compared to -97.65 and -97.56 for VAE and VampPrior.

5.2.2. CIFAR-10

Next, we experiment with ResNets of up to 6 stochastic layers, intertwined with deterministic layers comprising 6 ResNet blocks on CIFAR-10. We used 100 feature maps for all deterministic layers. The stochastic layers have 8 feature maps of width 16 × 16. We also experiment with skip connections between stochastic layers. We report results in table 4 comparing with other deep VAE models.

We obtain an ELBO of 3.5 bpd without skip connections. With skip connections we improve to 3.42 bpd on the 3-layer architecture, or 3.39 when evaluated with 100 importance samples. Compared to other deep VAE methods, we are on

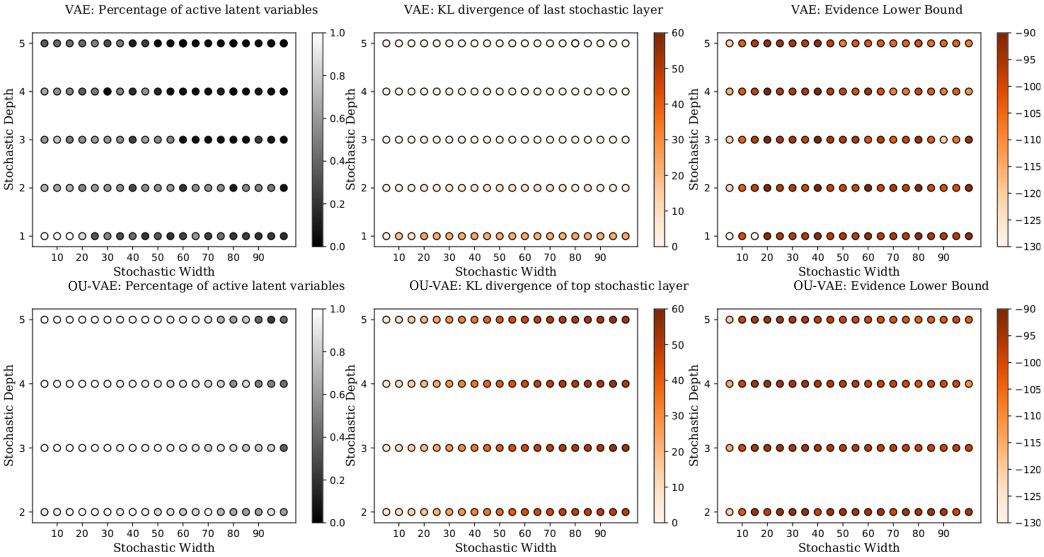

Figure 3. VAEs (top) exhibit suddenly many inactive units when increasing depth to 2 layers and width beyond 20 neurons, while collapsing completely after 3 layers and 60 neurons. OU-smoothed VAEs (bottom) retain for nearly all cases full latent activity, while maintaining similar ELBOs (not shown). Trained on MNIST for 100K steps for practical reasons (normal convergence if left training), with 2 deterministic, 200-neuron $\tanh$ layers before every stochastic layer.

Table 4. Comparing bits per dimension, parameters and depths on CIFAR-10. 'OU-VAE+' include stochastic skip connections.

Model	BPD	Layers	Parameters
Vanilla feedforward networks, residual connections			
OU-VAE, $\mathcal{L}_1$	3.5	3	9.95M
OU-VAE, $\mathcal{L}_1$	3.46	4	12.4M
OU-VAE, $\mathcal{L}_1$	3.43	6	16.8M
OU-VAE+, $\mathcal{L}_1$	3.42	3	9.95M
OU-VAE+, $\mathcal{L}_{100}$	3.39	3	9.95M
Feedforward networks, residual connections, shared weights between encoder and decoder			
LVAE (Maaløe et al., 2019)	3.60	15	72.36M
LVAE+ (Maaløe et al., 2019)	3.41	15	73.35M
LVAE+ (Maaløe et al., 2019)	3.45	29	119.71M
BIVA (Maaløe et al., 2019)	3.12	15	102.95M
VAE+IAF (Kingma et al., 2016)	3.11	–	–
NVAE (Vahdat & Kautz)	2.91	–	–
Feedforward networks, residual connections, normalizing flow prior/autoregressive			
Disc. VAE++ (Vahdat et al., 2018)	3.38	–	–
NICE (Dinh et al., 2014)	4.48	–	–
RealNVP (Dinh et al., 2017)	3.49	–	–

par with the 15-layer Ladder VAE (LVAE) (Maaløe et al., 2019), and comparable to the 15-layer LVAE+ (Maaløe et al., 2019) architecture that adds skip connections to LVAE. Note that OU-smoothed VAEs rely on much simpler architectures with 5-10X fewer layers. The 3-layer OU-smoothed VAE has about 7X fewer parameters than Ladder VAE. Considering further architectural innovations, as shown by NVAE and VAE+IAF, can further boost performance and we leave to future work.

We conclude that OU smoothing gives an efficient and accurate option for training deep VAEs even with vanilla architectural choices.

5.3. Complexity

In our experiments, 5-10 OU samples suffice, similar to IWAE, which also has a multi-sample objective. We do not OU sample in the final stochastic layer, which is especially important for deep VAEs as the last decoder layer is the largest and heaviest one to match image dimensions. *E.g.*, with 4 stochastic layers taking 5 samples we additionally require 5 times more memory for $p(z_2|z_1)$, $p(z_3|z_2)$, $p(z_4|z_3)$, *but not* $p(x|z_1)$. As in many architectures higher layers are typically smaller, the total memory is significantly less than 5 times the memory requirement of vanilla VAE.

We quantitatively support our justification in table 10 in the appendix with 4 and 6-layer models, using 5 deterministic layers per stochastic layer with 100 units per layer and 5 OU or IWAE samples. The memory usage is the maximum amount of used memory (batch size of 64). For LVAE and BIVA we use code from the BIVA repository. We see that all methods use a similar number of parameters (12-16M), except for LVAE and BIVA with 3-6X as many parameters. Save for regular VAE, OU-smoothed VAE requires the least memory, 50% less than IWAE and 30-50% less than LVAE and BIVA. We compare timings in table 9 in appendix. We bear a small only extra cost (4.6 sec) compared to VAE (4.2 sec) per epoch, on par with IWAE (4.4 sec) that has actually more parallel computations than us.

We conclude that the additional memory and computational cost by the OU sampling is small and OU-smoothed VAE models are considerably smaller than IWAE, LVAE and BIVA.

6. Conclusion

We present spectral smoothing for VAEs using the OU semigroup, based on analyzing intermediate VAE functions as Hermite expansions, using tools from the field of analysis of Gaussian functions. By damping the high frequency Hermite coefficients, we can now construct deep stochastic models with reduced variance. Furthermore, we show that casting the generative functions of VAEs with Gaussian latents on a Hermite basis yields a bias-variance decomposition to control the smoothing. We corroborate the theory by an extensive array of experiments. The deep stochastic models obtained with OU-smoothed VAEs show reduced variance, perform on par with much larger state-of-the-art models that rely on complex architectures to avoid collapse, and can eliminate (almost) completely posterior collapse even with very simple feedforward architectures. Importantly, we show the spectral smoothing in VAEs and variance reduction connects to phase transitions.

References

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Józefowicz, R., and Bengio, S. Generating Sentences from a Continuous Space. In *CoNLL*, 2016. doi: 10.18653/v1/K16-1002.

Burda, Y., Grosse, R. B., and Salakhutdinov, R. R. Importance Weighted Autoencoders. *CoRR*, abs/1509.00519, 2016. arXiv: 1509.00519.

Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., Sutskever, I., and Abbeel, P. Variational lossy autoencoder. *ArXiv*, abs/1611.02731, 2016.

Dinh, L., Krueger, D., and Bengio, Y. NICE: Non-linear Independent Components Estimation. *arXiv:1410.8516 [cs]*, October 2014. URL http://arxiv.org/abs/1410.8516. arXiv: 1410.8516.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. *arXiv:1605.08803 [cs, stat]*, February 2017. URL http://arxiv.org/abs/1605.08803. arXiv: 1605.08803.

Gulrajani, I., Kumar, K., Ahmed, F., Taïga, A. A., Visin, F., Vázquez, D., and Courville, A. C. PixelVAE: A Latent Variable Model for Natural Images. *ArXiv*, abs/1611.05013, 2017.

He, J., Spokoyny, D., Neubig, G., and Berg-Kirkpatrick, T. Lagging Inference Networks and Posterior Collapse in Variational Autoencoders. *arXiv:1901.05534 [cs, stat]*, January 2019. URL http://arxiv.org/abs/1901.05534. arXiv: 1901.05534.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*, 2017.

Janson, S. et al. *Gaussian hilbert spaces*, volume 129. Cambridge university press, 1997.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes, 2014.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved Variational Inference with Inverse Autoregressive Flow. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4743–4751. Curran Associates, Inc., 2016.

Lucas, J., Tucker, G., Grosse, R., and Norouzi, M. Don't Blame the ELBO! A Linear VAE Perspective on Posterior Collapse. *arXiv:1911.02469 [cs, stat]*, November 2019a. URL http://arxiv.org/abs/1911.02469. arXiv: 1911.02469.

Lucas, J., Tucker, G., Grosse, R., and Norouzi, M. Understanding Posterior Collapse in Generative Latent Variable Models. pp. 16, 2019b.

Maaløe, L., Sønderby, C. K., Sønderby, S. K., and Winther, O. Auxiliary Deep Generative Models. *arXiv:1602.05473 [cs, stat]*, February 2016. URL http://arxiv.org/abs/1602.05473. arXiv: 1602.05473.

Maaløe, L., Fraccaro, M., Liévin, V., and Winther, O. BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling. *arXiv:1902.02102 [cs, stat]*, November 2019. URL http://arxiv.org/abs/1902.02102. arXiv: 1902.02102.

Miller, A. C., Foti, N. J., D'Amour, A., and Adams, R. P. Reducing reparameterization gradient variance. 2017. URL http://arxiv.org/abs/1705.07880.

Oord, A. v. d., Kalchbrenner, N., and Kavukcuoglu, K. Pixel Recurrent Neural Networks. In *ICML*, 2016.

Pervez, A., Cohen, T., and Gavves, E. Low bias low variance gradient estimates for boolean stochastic networks. ICML, 2020.

Razavi, A., Oord, A. v. d., Poole, B., and Vinyals, O. Preventing Posterior Collapse with delta-VAEs. *arXiv:1901.03416 [cs, stat]*, January 2019. URL http://arxiv.org/abs/1901.03416. arXiv: 1901.03416.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. In *ICML*, 2015.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *ICML*, 2014.

Roeder, G., Wu, Y., and Duvenaud, D. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. 2017. URL http://arxiv.org/abs/1703.09194.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. Ladder Variational Autoencoders. In *NIPS*, 2016.

Tomczak, J. M. and Welling, M. Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

Tomczak, J. M. and Welling, M. VAE with a VampPrior. *arXiv preprint arXiv:1705.07120*, 2017.

Tucker, G., Lawson, D., Gu, S., and Maddison, C. J. Doubly reparameterized gradient estimators for monte carlo objectives. 2018. URL https://arxiv.org/abs/1810.04152v2.

Vahdat, A. and Kautz, J. NVAE: A deep hierarchical variational autoencoder. URL http://arxiv.org/abs/2007.03898.

Vahdat, A., Macready, W. G., Bian, Z., Khoshaman, A., and Andriyash, E. DVAE++: Discrete Variational Autoencoders with Overlapping Transformations. *arXiv:1802.04920 [cs, stat]*, May 2018. URL http://arxiv.org/abs/1802.04920. arXiv: 1802.04920.

Yang, Z., Hu, Z., Salakhutdinov, R., and Berg-Kirkpatrick, T. Improved variational autoencoders for text modeling using dilated convolutions. In *ICML*, 2017.