

Alma Mater Studiorum Università di Bologna  
Archivio istituzionale della ricerca

Variance Loss in Variational Autoencoders

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

*Published Version:*

Asperti, A. (2020). Variance Loss in Variational Autoencoders. Springer [10.1007/978-3-030-64583-0\_28].

*Availability:*

This version is available at: <https://hdl.handle.net/11585/796185> since: 2021-02-08

*Published:*

DOI: [http://doi.org/10.1007/978-3-030-64583-0\\_28](http://doi.org/10.1007/978-3-030-64583-0_28)

*Terms of use:*

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).  
When citing, please refer to the published version.

(Article begins on next page)

This is the final peer-reviewed accepted manuscript of:

**Asperti, A. (2020). Variance Loss in Variational Autoencoders. In: , et al. Machine Learning, Optimization, and Data Science. LOD 2020. Lecture Notes in Computer Science(), vol 12565. Springer, Cham**

The final published version is available online at  
[https://dx.doi.org/10.1007/978-3-030-64583-0\\_28](https://dx.doi.org/10.1007/978-3-030-64583-0_28)

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

*This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>)*

***When citing, please refer to the published version.***

# Variance Loss in Variational Autoencoders

Andrea Asperti

University of Bologna

Department of Informatics: Science and Engineering (DISI)

`andrea.asperti@unibo.it`

**Abstract.** In this article, we highlight what appears to be major issue of Variational Autoencoders (VAEs), evinced from an extensive experimentation with different networks architectures and datasets: the variance of generated data is significantly lower than that of training data. Since generative models are usually evaluated with metrics such as the Fréchet Inception Distance (FID) that compare the distributions of (features of) real versus generated images, the variance loss typically results in degraded scores. This problem is particularly relevant in a two stage setting [8], where a second VAE is used to sample in the latent space of the first VAE. The minor variance creates a mismatch between the actual distribution of latent variables and those generated by the second VAE, that hinders the beneficial effects of the second stage. Renormalizing the output of the second VAE towards the expected normal spherical distribution, we obtain a sudden burst in the quality of generated samples, as also testified in terms of FID.

## 1 Introduction

Since their introduction ([19,21]), Variational Autoencoders (VAEs) have rapidly become one of the most popular frameworks for generative modeling. Their appeal mostly derives from the strong probabilistic foundation; moreover, they are traditionally reputed for granting more stable training than Generative Adversarial Networks (GANs) ([12]).

However, the behaviour of Variational Autoencoders is still far from satisfactory, and there are a lot of well known theoretical and practical challenges that still hinder this generative paradigm. We may roughly identify four main (interrelated) topics that have been addressed so far:

**balancing issue** [5,18,15,7,8,3] a major problem of VAE is the difficulty to find a good compromise between sampling quality and reconstruction quality. The VAE loss function is a combination of two terms with somehow contrasting effects: the log-likelihood, aimed to reduce the reconstruction error, and the Kullback-Leibler divergence, acting as a regularizer of the latent space with the final purpose to improve generative sampling (see Section 2 for details). Finding a good balance between these components during training is a complex and delicate issue;

**variable collapse phenomenon** [6,27,2,25,8]. The KL-divergence component of the VAE loss function typically induces a parsimonious use of latent variables, some of which may be altogether neglected by the decoder, possibly resulting in a under-exploitation of the network capacity; if this is a beneficial side effect or regularization (sparsity), or an issue to be solved (overpruning), it is still debated;

**training issues** VAE approximate expectations through sampling during training that could cause an increased variance in gradients ([6,26]); this and other issues require some attention in the initialization, validation, and annealing of hyperparameters ([5,15,4])

**aggregate posterior vs. expected prior mismatch** [18,8,1,11] even after a satisfactory convergence of training, there is no guarantee that the learned aggregated posterior distribution will match the latent prior. This may be due to the choice of an overly simplistic prior distribution; alternatively, the issue can e.g. be addressed by learning the actual distribution, either via a second VAE or by ex-post estimation by means of different techniques.

The main contribution of this article is to highlight an additional issue that, at the best of our knowledge, has never been pointed out so far: the variance of generated data is significantly lower than that of training data.

This resulted from a long series of experiments we did with a large variety of different architectures and datasets. The variance loss is systematic, although its extent may vary, and looks roughly proportional to the reconstruction loss.

The problem is relevant because generative models are traditionally evaluated with metrics such as the popular Fréchet Inception Distance (FID) that compare the distributions of (features of) real versus generated images: any bias in generated data usually results in a severe penalty in terms of FID score.

The variance loss is particularly serious in a two stage setting [8], where we use a second VAE to sample in the latent space of the first VAE. The reduced variance induces a mismatch between the actual distribution of latent variables and those generated by the second VAE, substantially hindering the beneficial effects of the second stage.

We address the issue by a simple renormalization of the generated data to match the expected variance (that should be 1, in case of a two stage VAE). This simple expedient, in combination with a new balancing technique for the VAE loss function discussed in a different article [3], are the basic ingredients that permitted us to get the *best FID scores* ever achieved with variational techniques over traditional datasets such as CIFAR-10 and CelebA.

The cause of the reduced variance is not easy to identify. A plausible explanation is the following. It is well known that, in presence of multimodal output, the mean square error objective typically results in blurriness, due to averaging (see [14]).

Variational Autoencoders are intrinsically multimodal, due to the sampling process during training, comporting averaging around the input data  $X$  in the data manifold, and finally resulting in the blurriness so typical of Variational Autoencoders [10]. The reduced variance is just a different facet of the same

phenomenon: averaging on the data manifold eventually reduces the variance of data, due to Jensen’s inequality.

The structure of the article is the following. Section 2 contains a short introduction to Variational Autoencoders from an operational perspective, focusing on the regularization effect of the Kullback-Leibler component of the loss function. In Section 3, we discuss the variance loss issue, relating it to a similar problem of Principal Component Analysis, and providing experimental evidence of the phenomenon. Section 4 is devoted to our approach to the variance loss, with experimental results on CIFAR-10 and CelebA, two of the most common datasets in the field of generative modeling. A summary of the content of the article and concluding remarks are given in Section 5.

## 2 Variational Autoencoders

A Variational Autoencoder is composed by an encoder computing an *inference* distribution  $Q(z|X)$ , and a decoder, computing the posterior probability  $P(X|z)$ . Supposing that  $Q(z|X)$  has a Gaussian distribution  $N(\mu_z(X), \sigma_z(X))$  (different for each data  $X$ ), computing it amounts to compute its two first moments: so we expect the encoder to return the standard deviation  $\sigma_z(X)$  in addition to the mean value  $\mu_z(X)$ .

During decoding, instead of starting the reconstruction from  $\mu_z(X)$ , we sample around this point with the computed standard deviation:

$$\hat{z} = \mu_z(X) + \sigma_z(X) * \delta$$

where  $\delta$  is a random normal noise (see Figure 1). This may be naively understood as a way to inject noise in the latent representation, with the aim to improve the robustness of the autoencoder; in fact, it has a much stronger theoretical foundation, well addressed in the literature (see e.g. [9]). Observe that sampling is outside the backpropagation flow; backpropagating the reconstruction error (typically, mean squared error), we correct the current estimation of  $\sigma_z(X)$ , along with the estimation of  $\mu(X)$ .

Without further constraints,  $\sigma_z(X)$  would naturally collapse to 0: as a matter of fact,  $\mu_z(X)$  is the expected encoding, and the autoencoder would have no reason to sample away from this value. The variational autoencoder adds an additional component to the loss function, preventing  $Q(z|X)$  from collapsing to a dirac distribution: specifically, we try to bring each  $Q(z|X)$  close

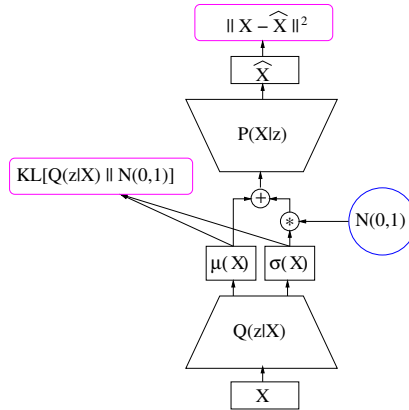


Fig. 1: VAE architecture

to the prior  $P(z)$  distribution by minimizing their Kullback-Leibler divergence  $KL(Q(z|X)||P(z))$ .

If we average this quantity on all input data, and expand KL-divergence in terms of entropy, we get:

$$\begin{aligned}
& \mathbb{E}_X KL(Q(z|X)||P(z)) \\
&= -\mathbb{E}_X \mathcal{H}(Q(z|X)) + \mathbb{E}_X \mathcal{H}(Q(z|X), P(z)) \\
&= -\mathbb{E}_X \mathcal{H}(Q(z|X)) + \mathbb{E}_X \mathbb{E}_{z \sim Q(z|X)} \log P(z) \\
&= -\mathbb{E}_X \mathcal{H}(Q(z|X)) + \mathbb{E}_{z \sim Q(z)} \log P(z) \\
&= -\underbrace{\mathbb{E}_X \mathcal{H}(Q(z|X))}_{\text{Avg. Entropy of } Q(z|X)} + \underbrace{\mathcal{H}(Q(z), P(z))}_{\text{Cross-entropy of } Q(X) \text{ vs } P(z)} \tag{1}
\end{aligned}$$

By minimizing the cross-entropy between the distributions we are pushing  $Q(z)$  towards  $P(z)$ . Simultaneously, we aim to augment the entropy of each  $Q(z|X)$ ; assuming  $Q(z|X)$  is Gaussian, this amounts to enlarge the variance, with the effect of improving the coverage of the latent space, essential for a good generative sampling. The price we have to pay is more overlapping, and hence more confusion, between the encoding of different datapoints, likely resulting in a worse reconstruction quality.

## 2.1 KL divergence in closed form

We already supposed that  $Q(X|z)$  has a Gaussian distribution  $N(\mu_z(X), \sigma_z(X))$ . Moreover, provided the decoder is sufficiently expressive, the shape of the prior distribution  $P(z)$  can be arbitrary, and for simplicity it is usually assumed to be a normal distribution  $P(z) = N(0, 1)$ . The term  $KL(Q(z|X)||P(z))$  is hence the KL-divergence between two Gaussian distributions  $N(\mu_z(X), \sigma_z(X))$  and  $N(1, 0)$  which can be computed in closed form:

$$KL(N(\mu_z(X), \sigma_z(X)), N(0, 1)) = \frac{1}{2}(\mu_z(X)^2 + \sigma_z^2(X) - \log(\sigma_z^2(X)) - 1) \tag{2}$$

The closed form helps to get some intuition on the way the regularizing effect of the KL-divergence is supposed to work. The quadratic penalty  $\mu_z(X)^2$  is centering the latent space around the origin; moreover, under the assumption to fix the ratio between  $\mu_z(X)$  and  $\sigma_z(X)$  (rescaling is an easy operation for a neural network) it is easy to prove [1] that expression 2 has a minimum when  $\mu_z(X)^2 + \sigma_z(X)^2 = 1$ . So, we expect

$$\mathbb{E}_X \mu(X) = 0 \tag{3}$$

and also, assuming 3, and some further approximation (see [1] for details),

$$\mathbb{E}_X \mu_z(X)^2 + \mathbb{E}_X \sigma_z^2(X) = 1 \tag{4}$$

If we look at  $Q(z) = \mathbb{E}_X Q(z|X)$  as a Gaussian Mixture Model (GMM) composed by a different Gaussian  $Q(z|X)$  for each  $X$ , the two previous equations express the two moments of the GMM, confirming that they coincide with those of a normal prior. Equation 4, that we call *variance law*, provides a simple sanity check to ensure that the regularization effect of the KL-divergence is working as expected.

Of course, even if two first moments of the aggregated inference distribution  $Q(z)$  are 0 and 1, it could still be very far from a Normal distribution. The possible mismatching between  $Q(z)$  and the expected prior  $P(z)$  is likely the most problematic aspect of VAEs since, as observed by several authors [16,22,1], it could compromise the whole generative framework. Possible approaches consist in revising the VAE objective by encouraging the aggregated inference distribution to match  $P(z)$  [23] or by exploiting more complex priors [17,24,4].

An interesting alternative addressed in [8] is that of training a second VAE to learn an accurate approximation of  $Q(z)$ ; samples from a Normal distribution are first used to generate samples of  $Q(z)$ , that are then fed to the actual generator of data points. Similarly, in [11], the authors try to give an ex-post estimation of  $Q(z)$ , e.g. imposing a distribution with a sufficient complexity (they consider a combination of 10 Gaussians, reflecting the ten categories of MNIST and Cifar10).

These two works provide the current state of the art in generative frameworks based on variational techniques (hence, excluding models based on adversarial training), so we shall mostly compare with them.

### 3 The variance loss issue

Autoencoders, and especially variational ones, seems to suffer from a systematic loss of variance of reconstructed/generated data with respect to source data. Suppose to have a training set  $X$  of  $n$  data, each one with  $m$  features, and let  $\hat{X}$  be the corresponding set of reconstructed data. We measure the (mean) variance loss as the mean over data (that is over the the first axis) of the differences of the variances of the features (i.e. over the second, default, axis):

$$\text{mean}(\text{var}(X) - \text{var}(\hat{X}))$$

Not only this quantity is always positive, but it is also approximately equal to the mean squared error (mse) between  $X$  and  $\hat{X}$ :

$$\text{mse}(X, \hat{X}) = \text{mean}((X - \hat{X})^2)$$

where the mean is here computed over all axes.

We observed the variance loss issue over a large variety of neural architectures and datasets. In particular cases, we can also give a theoretical explanation of the phenomenon, that looks strictly related to averaging. This is for instance the case of Principal Component Analysis (PCA), where the variance loss is precisely equal to the reconstruction error (it is well known that a *shallow* Autoencoder implements PCA, see e.g [13]).

Let us discuss this simple case first, since it helps to clarify the issue.

Principal component analysis (PCA) is a well know statistical procedure for dimensionality reduction. The idea is to project data in a lower dimensional space via an orthogonal linear transformation, choosing the system of coordinates that maximize the variance of data (principal components). These are easily computed as the vectors with the largest eigenvalues relative to the covariance matrix of the given dataset (centered around its mean points). Since the distance

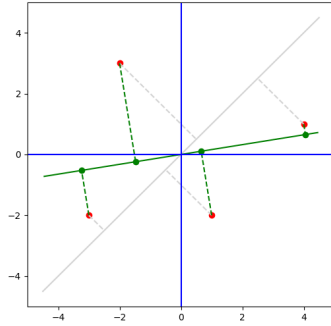


Fig.2: The principal component is the green line. Projecting the red points on it, we maximize their variance or equivalently we minimize their quadratic distance.

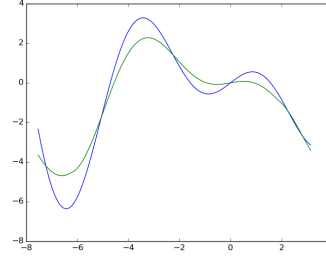


Fig.3: The green line is a smoother version of the blue line, obtained by averaging values in a suitable neighborhood of each point. The two lines have the same mean; the mean squared error between them is 0.546, the variance loss is 2.648.

of each point from the origin is fixed, by the Pythagorean theorem, maximizing its variance is equivalent to minimize its quadratic error from the hyper-plane defined by the principal components. For the same reason, *the quadratic error of the reconstruction is equal to the sum of the variance errors of the components which have been neglected.*

This is a typical example of variance loss due to averaging. Since we want to renounce some components, the best we can do along them is to take the mean value. We entirely lose the variance along these directions, that is going to be paid in terms of reconstruction error.

### 3.1 General case

We expect to have a similar phenomenon even with more expressive networks. The idea is expressed in Figure 3. Think of the blue line as the real data manifold; due to averaging, the network reconstructs a smoother version of the input data, resulting in a significant loss in terms of variance.



The need for averaging may have several motivations: it could be caused by a dimensionality reduction, as in the case of PCA, but also, in the case of variational autoencoders, it could derive from the Gaussian sampling performed before reconstruction. Since the noise injected during sampling is completely unpredictable, the best the network can do is to reconstruct an “average image” corresponding to a portion of the latent space around the mean value  $\mu_z(X)$ , spanning an area proportional to the variance  $\sigma_z(X)^2$ .

In Figure 4, we plot the relation between mean squared error (mse) and variance loss for *reconstructed images*, computed over a large variety of different neural architectures and datasets: the distribution is close to the diagonal. Typically, the variance loss for *generated images* is even greater. We must also account for a few pathological cases *not reported in the figure*, occurring with dense networks with very high capacity, and easily prone to overfitting. In this cases, mse is usually relatively high, while variance loss may drop to 0.

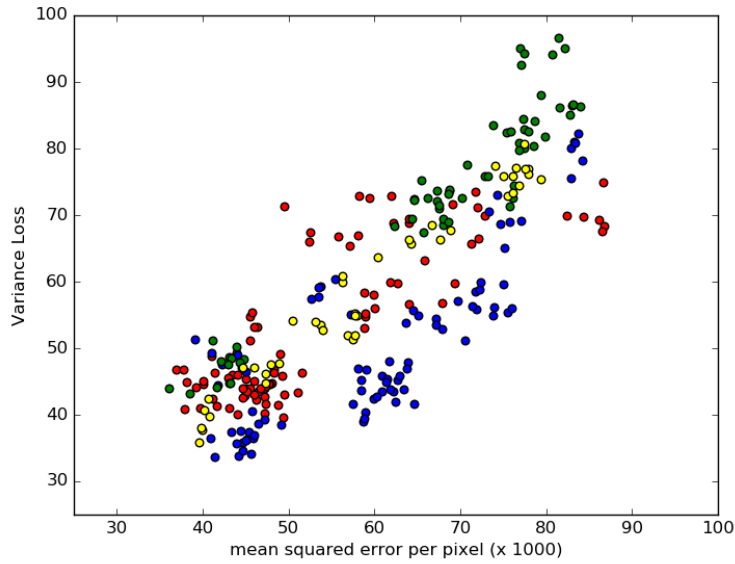


Fig. 4: Relation between mean squared error and variance loss. The different colors refer to different neural architectures: • (blue) Dense Networks; • (red) ResNet-like; • (green) Convolutional Networks; • (yellow) Iterative Networks (DRAW-GQN-like)

In the general, deep case, however, it is not easy to relate the variance loss to the mean squared error. We just discuss a few cases.

If for each data  $X$ , the reconstructed value  $\hat{X}$  is comprised between  $X$  and its mean value  $\mu$ , it is easy to prove that the mean squared error is a *lower bound*

to the variance loss (the worse case is when  $\hat{X} = \mu$ , where the variance loss is just equal to the mean squared error, as in the PCA case).

Similarly, let  $X_p$  be an arbitrary permutation of elements of  $X$  and let  $\hat{X} = (X + X_p)/2$ . Then, the mean square distance between  $X$  and  $\hat{X}$  is equal to the variance loss. However, the previous property does not generalize when we average over an arbitrary number of permutations; usually the mean squared error is lower than the quadratic distance between  $X$  and  $\hat{X}$ , but we can also get examples of the contrary.

We are still looking for a comfortable theoretical formulation of the property we are interested in.

## 4 Addressing the variance loss

As we explained in the introduction, the variance loss issue has a great practical relevance. Generative models are traditionally evaluated with metrics such as the popular Fréchet Inception Distance (FID) aimed to compare the distributions of real versus generated images through a comparison of extracted features. In the case of FID, the considered features are inception features; inception is usually preferred over other models due to the limited amount of preprocessing performed on input images. As a consequence, a bias in generated data may easily result in a severe penalty in terms of FID score (see [20] for an extensive analysis of FID in relation to the training set).

The variance loss is particularly dangerous in a two stage setting [8], where a second VAE is used to sample in the latent space of the first VAE, in order to fix the possible mismatch between the aggregate inference distribution  $Q(z)$  and the expected prior  $P(z)$ . The reduced variance induces a mismatch between the actual distribution of latent variables and those generated by the second VAE, hindering the beneficial effects of the second stage.

A simple way to address the variance loss issue consists in renormalizing generated data to match the actual variance of real data by applying a multiplicative scaling factor. We implemented this simple approach in a variant of ours of the two stage model of Dai and Wipf, based on a new balancing strategy between reconstruction loss and Kullback-Leibler described in [3]. We refer to this latter work for details about the structure of the network, hyperparameter configuration, and training settings, clearly outside the scope of this article. The code is available at <https://github.com/asperti/BalancingVAE>. In Figure 5 we provide examples of randomly generated faces. Note the particularly sharp quality of the images, so unusual for variational approaches.

Both for CIFAR-10 and CelebA, the renormalization operation results in an improvement in terms of FID scores, particularly significant in the case of CelebA, as reported in Tables 1 and 2. At the best of our knowledge, these are the best generative results ever obtained for these datasets without relying on adversarial training. In the Tables, we compare our generative model with the original two-stage model in [8] and with the recent deterministic model in [11]; as we mentioned above, these approaches represent the state of the art for



Fig. 5: Examples of **generated** faces. The resulting images do not show the blurred appearance so typical of variational approaches, significantly improving their perceptive quality.

Table 1: CIFAR-10: summary of results

model	REC	GEN-1	GEN-2
RAE-12 [11] (128 vars)	32.24±?	80.8±?	74.2±?
2S-VAE [8]		76.7 ± 0.8	72.9 ± 0.9
2S-VAE (ours)	53.8 ± 0.9	80.2 ± 1.3	69.8 ± 1.1
with normalization	53.5 ± 0.9	78.6 ± 1.2	<b>69.4 ± 1.0</b>

generative models not based on adversarial training. For our model, we provide scores with and without normalization. For each model, we give FID scores for reconstructed images (REC), images generated after the first stage (GEN-1), and

Table 2: CelebA: summary of results

model	REC	GEN-1	GEN-2
RAE-SN [11]	36.0 $\pm$ ?	44.7 $\pm$ ?	40.9 $\pm$ ?
2S-VAE [8]		60.5 $\pm$ 0.6	44.4 $\pm$ 0.7
2S-VAE (ours)	33.9 $\pm$ 0.8	43.6 $\pm$ 1.3	42.7 $\pm$ 1.0
with normalization	33.7 $\pm$ 0.8	42.7 $\pm$ 1.2	<b>38.6 <math>\pm</math> 1.0</b>

images generated after the second stage (GEN-2). In the case of the deterministic model [11], the “first stage” refers to sampling after fitting a Gaussian on the latent space, where the second stage refers to a more complex ex-post estimation of the latent space distribution via a GMM of ten Gaussians. The variance was computed over ten different trainings.



Fig. 6: Faces with and without *latent space* re-normalization (right and left respectively). Images on the right have better contrasts and more definite contours.

In Figure 6 we show the difference between faces generated from a same random seed with and without latent space re-normalization. We hope that the quality of images allows the reader to appreciate the improvement: renormalized images (on the right) have more precise contours, sharper contrasts and more definite details.

## 5 Conclusions

In this article, we stressed an interesting and important problem typical of autoencoders and especially of variational ones: the variance of generated data can be significantly lower than that of training data. We addressed the issue with a simple renormalization of generated data towards the expected moments of the data distribution, permitting us to obtain significant improvements in the quality of generated data, both in terms of perceptual assessment and FID score. On typical datasets such as CIFAR-10 and CelebA, this technique - in conjunction with a new balancing strategy between reconstruction error and Kullback-Leibler divergence - allowed us to get what seems to be the best generative results ever obtained without the use of adversarial training.

## References

1. Andrea Asperti. About generative aspects of variational autoencoders. In *Machine Learning, Optimization, and Data Science - 5th International Conference, LOD 2019, Siena, Italy, September 10-13, 2019, Proceedings*, pages 71–82, 2019.
2. Andrea Asperti. Sparsity in variational autoencoders. In *Proceedings of the First International Conference on Advances in Signal Processing and Artificial Intelligence, ASPAI, Barcelona, Spain, 20-22 March 2019*, 2019.
3. Andrea Asperti and Matteo Trentin. Balancing reconstruction error and Kullback-Leibler divergence in Variational Autoencoders. *CoRR*, abs/2002.07514, Feb 2020.
4. Matthias Bauer and Andriy Mnih. Resampled priors for variational autoencoders. *CoRR*, abs/1810.11428, 2018.
5. Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. Generating sentences from a continuous space. *CoRR*, abs/1511.06349, 2015.
6. Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *CoRR*, abs/1509.00519, 2015.
7. Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. 2018.
8. Bin Dai and David P. Wipf. Diagnosing and enhancing vae models. In *Seventh International Conference on Learning Representations (ICLR 2019), May 6-9, New Orleans*, 2019.
9. Carl Doersch. Tutorial on variational autoencoders. *CoRR*, abs/1606.05908, 2016.
10. Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 658–666, 2016.
11. Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael J. Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. *CoRR*, abs/1903.12436, 2019.
12. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *ArXiv e-prints*, June 2014.

13. Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
14. Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017.
15. Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2017.
16. Matthew D. Hoffman and Matthew J. Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, volume 1, 2016.
17. Diederik P. Kingma, Tim Salimans, Rafal Józefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational autoencoders with inverse autoregressive flow. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4736–4744, 2016.
18. Diederik P. Kingma, Tim Salimans, and Max Welling. Improving variational inference with inverse autoregressive flow. *CoRR*, abs/1606.04934, 2016.
19. Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
20. Daniele Ravaglia. Performance dei variational autoencoders in relazione al training set. Master’s thesis, University of Bologna, school of Science, Session II 2020.
21. Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic back-propagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org, 2014.
22. Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. Distribution matching in variational inference, 2018.
23. Ilya O. Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schölkopf. Wasserstein auto-encoders. *CoRR*, abs/1711.01558, 2017.
24. Jakub M. Tomczak and Max Welling. VAE with a vampprior. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2018, 9-11 April 2018, Playa Blanca, Lanzarote, Canary Islands, Spain*, pages 1214–1223, 2018.
25. Brian Trippe and Richard Turner. Overpruning in variational bayesian neural networks. In *Advances in Approximate Bayesian Inference workshop at NIPS 2017*, 2018.
26. George Tucker, Andriy Mnih, Chris J. Maddison, John Lawson, and Jascha Sohl-Dickstein. REBAR: low-variance, unbiased gradient estimates for discrete latent variable models. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 2627–2636, 2017.
27. Serena Yeung, Anitha Kannan, Yann Dauphin, and Li Fei-Fei. Tackling over-pruning in variational autoencoders. *CoRR*, abs/1706.03643, 2017.