The Evidence Lower Bound of Variational Autoencoders Converges to a Sum of Three Entropies

Jörg Lücke[1], Dennis Forster[1], Zhenwen Dai[2]
joerg.luecke@uol.de, dennis.forster@uol.de, zhenwend@spotify.com

[1] Machine Learning Lab, University of Oldenburg, Germany
[2] Spotify

Abstract

The central objective function of a variational autoencoder (VAE) is its variational lower bound. Here we show that for standard VAEs the variational bound is at convergence equal to the sum of three entropies: the (negative) entropy of the latent distribution, the expected (negative) entropy of the observable distribution, and the average entropy of the variational distributions. Our derived analytical results are exact and apply for small as well as complex neural networks for decoder and encoder. Furthermore, they apply for finite and infinitely many data points and at any stationary point (including local and global maxima). As a consequence, we show that the variance parameters of encoder and decoder play the key role in determining the values of variational bounds at convergence. Furthermore, the obtained results can allow for closed-form analytical expressions at convergence, which may be unexpected as neither variational bounds of VAEs nor log-likelihoods of VAEs are closed-form during learning. As our main contribution, we provide the proofs for convergence of standard VAEs to sums of entropies. Furthermore, we numerically verify our analytical results and discuss some potential applications. The obtained equality to entropy sums provides novel information on those points in parameter space that variational learning converges to. As such, we believe they can potentially significantly contribute to our understanding of established as well as novel VAE approaches.

1 Introduction

Variational autoencoders (VAEs; e.g. Kingma and Welling, 2014; Rezende et al., 2014) have emerged as very popular models for unsupervised learning. They combine elements of probabilistic generative models and deep neural networks (DNNs). Probabilistic generative models describe a given data distribution using parametrized probability density functions. A standard VAE is a model with one set of latent variables and one set of observable variables, where the two sets are coupled by deep neural networks (DNNs) for encoder and decoder. Let us denote the vector of latent variables by $\vec{z} \in \mathbb{R}^H$ and the vector of observed variables by $\vec{x} \in \mathbb{R}^D$. The decoder of a standard VAE is then given by:

Model VAE–1

$$p_\Theta(\vec{z}) = \mathcal{N}(\vec{z}; \vec{0}, \mathbb{1}) \tag{1}$$

$$p_\Theta(\vec{x} \,|\, \vec{z}) = \mathcal{N}(\vec{x}; \vec{\mu}_\Theta(\vec{z}), \sigma^2 \mathbb{1}) \tag{2}$$

$$\vec{\mu}_\Theta(\vec{z}) = \mathrm{DNN}_\mu(\vec{z}; W), \tag{3}$$

where $\vec{\mu}$ is a non-linear function parametrized by a DNN with parameters W (we take W to include all weight matrices and biases). $p_\Theta(\vec{z})$ denotes the prior distribution, and $p_\Theta(\vec{x} \,|\, \vec{z})$ is the observable distribution (both are Gaussian). We denote by $\Theta = (\sigma^2, W)$ the set of all parameters for the decoder.

The encoder of a standard VAE is for a given data point $\vec{x}^{(n)}$ defined by the following distribution:

$$q_\Phi^{(n)}(\vec{z}) = \mathcal{N}\big(\vec{z}; \vec{\nu}_\Phi(\vec{x}^{(n)}), \mathcal{T}_\Phi(\vec{x}^{(n)})\big), \quad \text{where} \tag{4}$$

$$\mathcal{T}_\Phi(\vec{x}^{(n)}) = \begin{pmatrix} \tau_1^2(\vec{x}^{(n)}; \Phi) & & \\ & \ddots & \\ & & \tau_H^2(\vec{x}^{(n)}; \Phi) \end{pmatrix} \tag{5}$$

is a diagonal covariance matrix with H positive entries $\tau_h^2(\vec{x}^{(n)})$ on the diagonal. The vectorial non-linearity $\vec{\tau}_\Phi^2(\vec{x}) = (\tau_1^2(\vec{x}; \Phi), \ldots, \tau_H^2(\vec{x}; \Phi))^\mathrm{T}$ for the covariance and the non-linearity $\vec{\nu}_\Phi(\vec{x})$ for the mean are both given by DNNs:

$$\vec{\nu}_\Phi(\vec{x}) = \mathrm{DNN}_\nu(\vec{x}; V), \tag{6}$$

$$\vec{\tau}_\Phi^2(\vec{x}) = \mathrm{DNN}_\tau(\vec{x}; T), \tag{7}$$

where V and T both include all weights and biases of their DNN, respectively. We denote by $\Phi = (V, T)$ the set of all encoder parameters.

Given a set of N data points $\vec{x}^{(n)}$, standard VAE learning algorithms seek decoder and encoder parameters (i.e., Θ and Φ) that maximize the central learning objective of VAEs: the variational lower bound. Two common (and mathematically equivalent) forms of the lower bound are:

$$\begin{aligned}\mathcal{F}(\Phi,\Theta) &= \tfrac{1}{N}\textstyle\sum_n \int q_\Phi^{(n)}(\vec{z})\log\big(p_\Theta(\vec{x}^{(n)}\,|\,\vec{z})\,p_\Theta(\vec{z})\big)\mathrm{d}\vec{z} \\ &\quad - \tfrac{1}{N}\textstyle\sum_n \int q_\Phi^{(n)}(\vec{z})\log\big(q_\Phi^{(n)}(\vec{z})\big)\mathrm{d}\vec{z} \qquad (8)\\ &= \tfrac{1}{N}\textstyle\sum_n \int q_\Phi^{(n)}(\vec{z})\log\big(p_\Theta(\vec{x}^{(n)}\,|\,\vec{z})\big)\mathrm{d}\vec{z} \\ &\quad - \tfrac{1}{N}\textstyle\sum_n D_{\mathrm{KL}}\big(q_\Phi^{(n)}(\vec{z}), p_\Theta(\vec{z})\big), \qquad (9)\end{aligned}$$

where $D_{\mathrm{KL}}\big(q(\vec{z}), p(\vec{z})\big)$ denotes the Kullback-Leibler divergence between two distributions.

The variational lower bound is for all encoder parameters Φ smaller or equal to the log-likelihood $\mathcal{L}(\Theta)$ which is defined by the decoder. As a tribute to its importance, the variational lower bound is known under different names, including *evidence lower bound* (ELBO; e.g. Blei et al., 2017) or variational *free energy* (e.g. Neal and Hinton, 1998). Variational bounds, which are defined through their *variational distributions* $q_\Phi^{(n)}(\vec{z})$, make optimization of model parameters much more convenient than a direct optimization of the likelihood. Only for elementary generative models such as probabilistic PCA (Tipping and Bishop, 1999) and few other examples (e.g. Monk et al., 2016) analytical solutions for variational lower bounds are known. The analytical intractability of the bound stems from the first summand (in Eqn. 8 or Eqn. 9): Because of (potentially very intricate) DNN non-linearities of standard VAEs, the integrals can not be solved analytically. The central research challenge for training VAEs is therefore the development of efficient methods to approximate intractable integrals of variational bounds. Indeed, the suggestion of efficient methods to estimate gradients of Eqn. (8) using sampling and reparametrization (Kingma and Welling, 2014; Rezende et al., 2014) has played the key role in the establishment of the whole field of VAE research.

Because of the potentially complex DNNs of VAEs few exact theoretical results may be expected especially in realistic settings, i.e., for real and finite data sets and convergence to local optima. Therefore, maybe surprisingly, we here provide an exact analytical result for all standard VAEs (VAE–1 as well as later discussed generalizations): we show that the variational bound is at convergence given by a sum of the entropies of their constituting distributions.

The derived results are based on exploiting properties of the variational bound (8) at stationary points, i.e., at points in parameter space that learning converges to. Proofs of the convergence results as well as the results themselves are our main contribution.

For VAE–1, the final result is concise enough to be stated here initially (before we discuss its derivation and related work later on): at all stationary points of the variational lower bound (8) of VAE–1 it applies:

$$\begin{aligned}&\mathcal{F}(\Phi,\Theta) \qquad (10)\\ &= -\mathcal{H}[p_\Theta(\vec{z})] - \mathcal{H}[p_\Theta(\vec{x}\,|\,\vec{z})] + \frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[q_\Phi(\vec{z};\vec{x}^{(n)})]\\ &= -\frac{D}{2}\log(2\pi e\sigma^2) + \frac{1}{2N}\sum_{n=1}^{N}\sum_{h=1}^{H}\log\big(\tau_h^2(\vec{x}^{(n)};\Phi)\big)\end{aligned}$$

where $\mathcal{H}[p(\cdot)]$ denotes the entropy of a distribution $p(\cdot)$. The bound is thus simply given by the entropies of prior, decoder and encoder distribution. As such, it only depends on the variance parameters of the encoder and decoder and not on any parameters of the distribution means. We also remark that in virtue of Eqn. (10) there are no approximations of any integrals required after convergence. Using just the data points and the learned parameters the bound can be computed exactly. The closed-form variational bound (10) does *not* replace the original bound as a learning objective, of course (i.e., it can not be used analogously to Eqn. 8 or 9 for learning). As an example of practical relevance, Eqn. (10) can, however, very easily be used to determine the value of the bound after learning has converged. Most importantly, however, is the theoretical finding, i.e., the fact that during learning the bound *has to* convergence to the sum of entropies (10).

Related work. There is a long-standing interest in stationary points of variational lower bounds. For instance, likelihoods, lower bounds and their stationary points have been studied in detail for (restricted) Boltzmann machines (e.g. Welling and Teh, 2003), elementary generative models (e.g. Frey, 1999; Tipping and Bishop, 1999) as well as general graphical models (Freeman and Weiss, 2000; Weiss and Freeman, 2000; Yedidia et al., 2001; Opper and Saad, 2001). For VAEs, stationary points are likewise of high interest. For instance, Lucas et al. (2019) linked phenomena like mode collapse in VAEs to the stability of stationary points of the variational lower bound.

Our derivations for VAEs will show that variational bounds are at convergence directly related to the entropies of those distributions that define VAEs. The relation between the integrals of the variational lower bound and entropies has been of interest previously (Lücke and Henniges, 2012). However, the main results of the previous work have used (A) the limit case of infinitely many data points, (B) assumed a perfect match of variational and full posterior distri-

butions, and (C) required convergence to global optima. The work also discussed relaxations of these rather rigid assumptions (Lücke and Henniges, 2012, Sec. 6). But those relaxations made use of properties of sparse coding like models trained using EM. While being an important inspiration for our study, the previous results are not applicable to the training of VAEs.

Finally, a related but different line of research discusses how other learning objectives for deep models can be defined that are closed-form by definition. Examples are work by Oord et al. (2016) and Kingma and Dhariwal (2018). In contrast, we here investigate the standard (in general intractable) learning objective of VAEs and show its analytical tractability at convergence.

2 Variational Bounds at Convergence

For the derivation of the introductory VAE (VAE–1, defined by Eqns. 1 to 4) we will need a more explicit form of the decoder DNN (3). We assume the DNN to be of standard form, i.e., the non-linearity $\vec{\mu}(\vec{z};\Theta)$ is a series of nested linear matrix multiplications followed by point-wise non-linearities:

$$\vec{\mu}(\vec{z};\Theta) = W^{(L)}\mathcal{S}_{L-1}\big(\cdots\mathcal{S}_0(W^{(0)}\vec{z}+\vec{b}_0)\cdots\big)+\vec{b}_L, \tag{11}$$

where $W^{(l)}$ denotes the weight matrix of layer l and $\vec{b}_l$ denotes its bias terms. The pointwise non-linearities $\mathcal{S}_l(\cdot)$ can differ from layer to layer. No further explicit forms are required from the encoder non-linearities.

2.1 A Reparametrized VAE

Before we derive Eqn. (10) for VAE–1, let us consider a different VAE which can be regarded as a reparametrization of VAE–1. It takes the following form:

Model VAE–2

$$p_\Theta(\tilde{\vec{z}}) = \mathcal{N}(\tilde{\vec{z}};\vec{0},A) \tag{12}$$

$$p_\Theta(\vec{x}\,|\,\tilde{\vec{z}}) = \mathcal{N}(\vec{x};\tilde{\vec{\mu}}_\Theta(\tilde{\vec{z}}),\sigma^2\mathbb{1}) \tag{13}$$

$$q_\Phi^{(n)}(\tilde{\vec{z}}) = \mathcal{N}\big(\tilde{\vec{z}};\vec{\nu}_\Phi(\vec{x}^{(n)}),\tilde{\mathcal{T}}_\Phi(\vec{x}^{(n)})\big), \tag{14}$$

$$\text{with } A = \begin{pmatrix}\alpha_1^2 & & \\ & \ddots & \\ & & \alpha_H^2\end{pmatrix}. \tag{15}$$

The matrix $\tilde{\mathcal{T}}_\Phi(\vec{x}^{(n)})$ is defined as in Eqn. (5) but with $\tilde{\tau}_h^2(\vec{x}^{(n)};\Phi)$ instead of $\tau_h^2(\vec{x}^{(n)};\Phi)$ denoting the positive diagonal elements. We will assume the same decoder DNN for VAE–2 as for VAE–1 except of constraining the matrix of the first linear mapping to have columns of unit length. Concretely, we use

$$\tilde{\vec{\mu}}_\Theta(\tilde{\vec{z}}) = W^{(L)}\mathcal{S}_{L-1}\big(\cdots\mathcal{S}_0(\tilde{W}^{(0)}\tilde{\vec{z}}+\vec{b}_0)\cdots\big)+\vec{b}_L, \tag{16}$$

$$\text{where}\quad \tilde{W}^{(0)} = (\tilde{\vec{W}}_1^{(0)},\ldots,\tilde{\vec{W}}_H^{(0)}) \tag{17}$$

$$\text{with } (\tilde{\vec{W}}_h^{(0)})^{\mathrm{T}}\tilde{\vec{W}}_h^{(0)} = 1 \;\text{ for all } h. \tag{18}$$

We will later see that VAE–2 can parametrize the same distributions as VAE–1. The advantage of VAE–2 compared to VAE–1 is that it is of a form for which the variance parameters of the prior distribution can be learned (which will be exploited below).

The stationary points of VAE–2 are clearly defined. They are those points in parameter space for which the derivatives w.r.t. all parameters (parameters Θ and Φ) vanish. For VAE–2 this implies:

$$\frac{d}{d\sigma^2}\mathcal{F}(\Phi,\Theta) = 0 \quad\text{and}\quad \frac{d}{d\alpha_h^2}\mathcal{F}(\Phi,\Theta) = 0\;\forall h. \tag{19}$$

For VAE–2 the derivatives w.r.t. the DNN parameters $W^{(l)}$ and $\vec{b}_l$ are also zero at stationary points as well as all derivatives w.r.t. the encoder parameters Φ (for the parameters $\tilde{W}^{(0)}$ a derivative with Lagrange multipliers is zero). However, we will below only use (19).

We can now proceed to proof the following proposition:

Proposition 1
Consider VAE–2 defined by Eqns. (12) to (18). Then at all stationary points of the variational lower bound of VAE–2 the following applies:

$$\begin{aligned}\mathcal{F}(\Phi,\Theta)&\\ &= -\mathcal{H}[p_\Theta(\tilde{\vec{z}})] - \mathcal{H}[p_\Theta(\vec{x}\,|\,\tilde{\vec{z}})] + \frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[q_\Phi(\tilde{\vec{z}};\vec{x}^{(n)})]\\ &= -\frac{1}{2}\log\big(\det(2\pi e A)\big) - \frac{D}{2}\log\big(2\pi e\sigma^2\big)\\ &\quad + \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\log\big(\det(2\pi e\tilde{\mathcal{T}}_\Phi(\vec{x}^{(n)}))\big).\end{aligned} \tag{20}$$

Proof
We can rewrite the standard formulation of the variational lower bound (8) to consist of three terms:

$$\begin{aligned}\mathcal{F}(\Phi,\Theta) &= \mathcal{F}_1(\Phi,\Theta)+\mathcal{F}_2(\Phi,\Theta)+\mathcal{F}_3(\Phi),\;\text{where}\\ \mathcal{F}_1(\Phi,\Theta) &= \mathcal{F}_1 = \tfrac{1}{N}\textstyle\sum_n\int q_\Phi^{(n)}(\vec{z})\log\big(p_\Theta(\vec{z})\big)\,\mathrm{d}\vec{z}\\ \mathcal{F}_2(\Phi,\Theta) &= \mathcal{F}_2 = \tfrac{1}{N}\textstyle\sum_n\int q_\Phi^{(n)}(\vec{z})\log\big(p_\Theta(\vec{x}^{(n)}\,|\,\vec{z})\big)\mathrm{d}\vec{z}\\ \mathcal{F}_3(\Phi) &= \mathcal{F}_3 = -\tfrac{1}{N}\textstyle\sum_n\int q_\Phi^{(n)}(\vec{z})\log\big(q_\Phi^{(n)}(\vec{z})\big)\mathrm{d}\vec{z},\end{aligned}$$

where we dropped the 'tilde' for $\vec{z}$ for the proof.

First consider $\mathcal{F}_2$ and observe that the logarithm of the prefactor $-\frac{D}{2}\log(2\pi\sigma^2)$ of $p_\Theta(\vec{x}\,|\,\vec{z})$ resembles the entropy of the Gaussian distribution (up to a constant factor). Hence, we can reexpress $\mathcal{F}_2$ as follows:

$$\begin{aligned}\mathcal{F}_2 &= \frac{1}{N}\sum_n \Big(-\frac{1}{2\sigma^2}\int q^{(n)}_\Phi(\vec{z})\,\|\vec{x}^{(n)}-\vec{\mu}_\Theta(\vec{z})\|^2\mathrm{d}\vec{z} \\ &\qquad -\frac{D}{2}\log\big(2\pi\sigma^2\big)-\frac{D}{2}+\frac{D}{2}\Big) \\ &= \frac{D}{2}\Big(1-\frac{1}{ND\sigma^2}\sum_n\int q^{(n)}_\Phi(\vec{z})\,\|\vec{x}^{(n)}-\vec{\mu}_\Theta(\vec{z})\|^2\mathrm{d}\vec{z}\Big) \\ &\quad -\frac{D}{2}\log\big(2\pi e\sigma^2\big), \qquad (21)\end{aligned}$$

where the last term is now the negative entropy of a Gaussian (where 'e' is Euler's number).

At stationary points it applies that $\frac{\mathrm{d}}{\mathrm{d}\sigma^2}\mathcal{F}(\Phi,\Theta)=0$. Only $\mathcal{F}_2(\Phi,\Theta)$ depends on σ^2, which implies $\frac{\mathrm{d}}{\mathrm{d}\sigma^2}\mathcal{F}_2(\Phi,\Theta)=0$. In virtue of Eqn. (21), the derivative has a specific structure given by:

$$\begin{aligned}0 &= \frac{\mathrm{d}}{\mathrm{d}\sigma^2}\mathcal{F}_2 \\ &= \frac{D}{2}\left(\frac{1}{ND\sigma^4}\sum_n\int q^{(n)}_\Phi(\vec{z})\|\vec{x}^{(n)}-\vec{\mu}_\Theta(\vec{z})\|^2\mathrm{d}\vec{z}\right)-\frac{D}{2\sigma^2} \\ &= -\frac{D}{2\sigma^2}\left(1-\frac{1}{ND\sigma^2}\sum_n\int q^{(n)}_\Phi(\vec{z})\|\vec{x}^{(n)}-\vec{\mu}_\Theta(\vec{z})\|^2\mathrm{d}\vec{z}\right).\end{aligned}$$

As $\frac{D}{2\sigma^2}$ is greater zero, it follows that

$$1-\frac{1}{ND\sigma^2}\sum_n\int q^{(n)}_\Phi(\vec{z})\|\vec{x}^{(n)}-\vec{\mu}_\Theta(\vec{z})\|^2\mathrm{d}\vec{z}=0.$$

We recognize the integral to be the first term of Eqn. (21), i.e., the term not depending on the Gaussian entropy. We can thus conclude that at stationary points of $\mathcal{F}(\Phi,\Theta)$ it applies that

$$\mathcal{F}_2 = -\frac{D}{2}\log\big(2\pi e\sigma^2\big) = -\mathcal{H}[p_\Theta(\vec{x}\,|\,\vec{z})]. \qquad (22)$$

Next we consider the term $\mathcal{F}_1$ of the variational lower bound. Analogous to $\mathcal{F}_2$, we observe that the logarithm of the prefactor is similar to the entropy of a Gaussian (this time with diagonal covariance) and we rewrite:

$$\begin{aligned}\mathcal{F}_1 &= \frac{1}{N}\sum_n\Big(-\sum_h\frac{1}{2\alpha_h^2}\int q^{(n)}_\Phi(\vec{z})z_h^2\mathrm{d}\vec{z}-\frac{1}{2}\sum_h\log\big(2\pi\alpha_h^2\big)\Big) \\ &= \frac{1}{2}\sum_h\Big(1-\frac{1}{N\alpha_h^2}\sum_n\int q^{(n)}_\Phi(\vec{z})z_h^2\mathrm{d}\vec{z}\Big)-\frac{1}{2}\sum_h\log\big(2\pi e\alpha_h^2\big), \qquad (23)\end{aligned}$$

where the last term is the negative entropy of the prior. At stationary points Eqn. (19) applies and we obtain:

$$\begin{aligned}0 = \frac{\mathrm{d}}{\mathrm{d}\alpha_h^2}\mathcal{F}_2 &= \frac{1}{2}\sum_{h'}\frac{\delta_{hh'}}{N\alpha_h^4}\sum_n\int q^{(n)}_\Phi(\vec{z})z_{h'}^2\mathrm{d}\vec{z}-\frac{1}{2}\sum_{h'}\frac{\delta_{hh'}}{\alpha_h^2} \\ &= -\frac{1}{2\alpha_h^2}\Big(1-\frac{1}{N\alpha_h^2}\sum_n\int q^{(n)}_\Phi(\vec{z})z_h^2\mathrm{d}\vec{z}\Big).\end{aligned}$$

As $\frac{1}{2\alpha_h^2}$ is greater zero, it follows that for each h

$$1-\frac{1}{N\alpha_h^2}\sum_n\int q^{(n)}_\Phi(\vec{z})(z_h)^2\mathrm{d}\vec{z}=0.$$

The first sum over h in Eqn. (23) is consequently zero, and we obtain at convergence:

$$\mathcal{F}_1 = -\frac{1}{2}\log\big(\det(2\pi eA)\big) = -\mathcal{H}[p_\Theta(\vec{z})]. \qquad (24)$$

The term $\mathcal{F}_3$ is equal to the average entropy of the variational distribution. Taken together, we thus obtain the claim (20). □

2.2 Convergence to Sums of Entropies

We can now provide a result for the variational bounds of VAE–1 by translating the result of Prop. 1 back to the original parametrization.

Proposition 2
Consider VAE–1 defined by Eqns. (1) to (5). At all stationary points of the variational lower bound of VAE–1 it applies that:

$$\begin{aligned}&\mathcal{F}(\Phi,\Theta) \\ &\quad= -\mathcal{H}[p_\Theta(\vec{z})]-\mathcal{H}[p_\Theta(\vec{x}\,|\,\vec{z})]+\frac{1}{N}\sum_{n=1}^N\mathcal{H}[q_\Phi(\vec{z};\vec{x}^{(n)})] \\ &\quad= -\frac{D}{2}\log\big(2\pi e\sigma^2\big)+\frac{1}{2N}\sum_{n=1}^N\sum_{h=1}^H\log\big(\tau_h^2(\vec{x}^{(n)};\Phi)\big), \qquad (25)\end{aligned}$$

Proof
Given VAE–1, we replace the weights $W^{(0)}$ of the decoder DNN (11) as follows:

$$W^{(0)} = \tilde{W}^{(0)}A^{\frac{1}{2}} = \big(\alpha_1\tilde{\vec{W}}_1^{(0)},\ldots,\alpha_H\tilde{\vec{W}}_H^{(0)}\big), \qquad (26)$$

with $\tilde{W}^{(0)} = \big(\tilde{\vec{W}}_1^{(0)},\ldots,\tilde{\vec{W}}_H^{(0)}\big)$ and A as in Eqn. (15). We take the column vectors of $\tilde{W}^{(0)}$ to be constrained to unit length as in Eqn. (17). A and $\tilde{W}^{(0)}$ parametrize the same space of matrices as W. The first linear operation of the DNN (11) now becomes:

$$W^{(0)}\vec{z}+\vec{b}_0 = \textstyle\sum_h\tilde{\vec{W}}_h^{(0)}\alpha_h z_h+\vec{b}_0\,. \qquad (27)$$

Considering the term $\alpha_h z_h$, we can instead of generating $z_h \sim \mathcal{N}(z_h; 0, 1)$ generate $\tilde{z}_h \sim \mathcal{N}(\tilde{z}_h; 0, \alpha_h^2)$. So we recognize that VAE–1 takes on the form of VAE–2 (12) to (18). Hence, when the parameters of VAE–1 represent a stationary point, the parameters with $W^{(0)}$ replaced by A and $\tilde{W}^{(0)}$ also represent a stationary point. Therefore, we can use Eqn. (20) of Prop. 1. The σ^2 remains unchanged and A could be obtained from the column vectors of $W^{(0)}$. However, it is left to express $\tilde{\mathcal{T}}_\Phi(\vec{x})$ in terms of $\mathcal{T}_\Phi(\vec{x})$. $\tilde{\mathcal{T}}$ is the covariance matrix of a Gaussian distribution of the space of $\tilde{\vec{z}}$. The random variable $\vec{z}$ is because of Eqn. (27) given by $\vec{z} = A^{-\frac{1}{2}}\tilde{\vec{z}}$. Consequently, if $\tilde{\vec{z}}$ is Gaussian distributed with covariance $\tilde{\mathcal{T}}$, then $\vec{z}$ is Gaussian distributed with covariance $\mathcal{T} = A^{-\frac{1}{2}}\tilde{\mathcal{T}}\big(A^{-\frac{1}{2}}\big)^{\mathrm{T}}$. As all matrices are diagonal, we get $\tilde{\mathcal{T}} = A\mathcal{T}$. Inserting into Eqn. (20) we observe that the first term cancels with part of the last term:

$$\begin{aligned}
\mathcal{F}(\Phi,\Theta) &= -\frac{1}{2}\log\big(\det(2\pi e A)\big) - \frac{D}{2}\log\big(2\pi e\sigma^2\big) \\
&\quad + \frac{1}{N}\sum_n \frac{1}{2}\log\big(\det(2\pi e A\mathcal{T}_\Phi(\vec{x}^{(n)})\big) \\
&= -\frac{H}{2}\log(2\pi e) - \frac{D}{2}\log\big(2\pi e\sigma^2\big) \\
&\quad + \frac{1}{N}\sum_n \frac{1}{2}\log\big(\det(2\pi e\mathcal{T}_\Phi(\vec{x}^{(n)})\big) \\
&= -\frac{D}{2}\log\big(2\pi e\sigma^2\big) + \frac{1}{2N}\sum_n \log\big(\det(\mathcal{T}_\Phi(\vec{x}^{(n)})\big).
\end{aligned} \tag{28}$$

The middle equation we recognize as the sum of three entropies in Eqn. (20), while the last equation is a further simplification. □

There are a number of implications and remarks if considering Prop. 2: As already pointed out in the introduction, no approximations of any integrals (nor any other approximations) are required. The computation of the bound is consequently very easy and efficient in practice.[1] More importantly, however, is the observation that learning of VAEs (as given by VAE–1) necessarily converges to sums of entropies. As a consequence, the bound only depends on a subset of VAE parameters at convergence: the variances of encoder and decoder (the entropies do not depend on the DNNs for Gaussian means). An interpretation of the result is that the goal of VAE training is to make the final variance of the decoder as small as possible while trying to keep the variances of the encoder as large as possible. Any other parameters of the VAE only affect the bound through these variances.

[1] We remark in this context that the DNN for the covariance of the variational distributions is in practice anyway often defined using the $\log\big(\tau_h^2(\vec{x};\Phi)\big)$.

2.3 Linear VAEs

Standard VAEs as studied above exhibit complex learning behavior such that theoretical insights are notoriously difficult to obtain. To better understand salient challenges of VAE training such as mode collapse, a natural approach is to first try to gain insights using as elementary as possible models. For VAEs, the most elementary such model is presumably represented by a linear VAE (also compare Rumelhart et al., 1985; Baldi and Hornik, 1989; Kunin et al., 2019) which was used by Lucas et al. (2019) to study mode collapse:

$$p_\Theta(\vec{z}) = \mathcal{N}(\vec{z}; \vec{0}, \mathbb{1}) \tag{29}$$

$$p_\Theta(\vec{x}\,|\,\vec{z}) = \mathcal{N}(\vec{x}; W\vec{z} + \vec{\mu}_0, \sigma^2\mathbb{1}) \tag{30}$$

with the encoder given by

$$q_\Phi^{(n)}(\vec{z}) = \mathcal{N}(\vec{z}; V(\vec{x}^{(n)} - \vec{\mu}_0), \mathcal{T}), \tag{31}$$

$$\text{where} \quad \mathcal{T} = \begin{pmatrix} \tau_1^2 & & \\ & \ddots & \\ & & \tau_H^2 \end{pmatrix}. \tag{32}$$

The linear VAE is a special case of VAE–1, with linear DNNs instead of the usual deep non-linear versions. Therefore, we can conclude the following:

Corollary 1
Consider the linear VAE defined by Eqns. (29) to (32). Then at all stationary points of its variational lower bound (9) it applies that:

$$\mathcal{F}(\Phi,\Theta) = -\tfrac{D}{2}\log\big(2\pi e\sigma^2\big) + \tfrac{1}{2}\textstyle\sum_{h=1}^{H}\log\big(\tau_h^2\big). \tag{33}$$

Proof
For the proof of Prop. 2 we only required the reparametrization of the first linear mapping of the decoder DNN (11). Prop. 2 thus also applies for the linear VAE as a special case of VAE–2 (we elaborate in Appendix A). Inserting the matrix $\mathcal{T}$ of Eqn. (32) into Eqn. (25) directly results in the claim (33) as $\mathcal{T}$ does not depend on n. □

Eqn. (33) further highlights that the variance parameters determine the bound at convergence. As Lucas et al. (2019) showed that the linear VAE can recover the exact likelihood, we can even conclude that the bound (33) is tight at convergence. We use this result in Sec. 3 and elaborate in Appendix A.

2.4 General Standard VAEs

VAE–1 represents a very common form of VAEs. However, generalizations which use a DNN for decoder variances alongside a DNN for decoder means are widespread as well (Rezende et al., 2014; Dorta

et al., 2018, etc). To cover all VAEs with standard Gaussian distributions, we hence have to extend our analysis. Consider therefore VAEs of the following form:

Model VAE–3

$$p_\Theta(\vec{z}) = \mathcal{N}(\vec{z}; \vec{0}, \mathbb{1}) \tag{34}$$

$$p_\Theta(\vec{x} \,|\, \vec{z}) = \mathcal{N}\big(\vec{x}; \vec{\mu}_\Theta(\vec{z}), \Sigma_\Theta(\vec{z})\big). \tag{35}$$

The decoder covariance Σ is diagonal with diagonal elements depending on $\vec{z}$:

$$\Sigma_\Theta(\vec{z}) = \begin{pmatrix} \sigma_1^2(\vec{z};\Theta) & & \\ & \ddots & \\ & & \sigma_D^2(\vec{z};\Theta) \end{pmatrix}, \tag{36}$$

where we will denote the vector of diagonal elements by $\vec{\sigma}^2_\Theta(\vec{z}) = (\sigma_1^2(\vec{z};\Theta), \ldots, \sigma_D^2(\vec{z};\Theta))^{\mathrm{T}}$. This vector we will now take to be parametrized by a DNN, i.e., the decoder of VAE–3 uses two DNNs:

$$\vec{\mu}_\Theta(\vec{z}) = \mathrm{DNN}_\mu(\vec{z}; W), \tag{37}$$

$$\vec{\sigma}^2_\Theta(\vec{z}) = \mathrm{DNN}_\sigma(\vec{z}; M). \tag{38}$$

We take these two DNNs to be parametrized by two distinct sets of parameters, W and M, and both sets of parameters we take to contain all weights and biases of the respective DNN. We require both DNNs to have at least one hidden layer. The encoder we take to be the same as for VAE–1.

Because of the $\vec{z}$-dependence, it is obvious that Prop. 2 can not apply since a naive replacement of σ^2 in Eqn. (25), even assuming a scalar $\sigma^2(\vec{z})$, does not make sense. The proof of Prop. 2 explicitly used that σ^2 does not depend on $\vec{z}$, so no straight-forward generalization offers itself. However, at convergence it is still possible to derive expressions for the bound in terms of entropies:

Proposition 3
Consider VAE–3 defined by Eqns. (34) to (38) with encoder (4) and (5). At all stationary points of the variational lower bound of VAE–3 it holds that:

$$\begin{aligned} &\mathcal{F}(\Phi,\Theta) \qquad\qquad (39)\\ &= \frac{1}{N}\sum_{n=1}^{N} \mathcal{H}[q_\Phi(\vec{z};\vec{x}^{(n)})] - \mathcal{H}[p_\Theta(\vec{z})] \\ &\quad -\frac{1}{N}\sum_{n=1}^{N} \mathrm{E}_{q_\Phi^{(n)}}\big\{\mathcal{H}[p_\Theta(\vec{x}\,|\,\vec{z})]\big\} \\ &= \frac{1}{2N}\sum_{n=1}^{N}\sum_{h=1}^{H} \log\big(\tau_h^2(\vec{x}^{(n)};\Phi)\big) \\ &\quad -\frac{1}{2N}\sum_{n=1}^{N}\sum_{d=1}^{D} \mathrm{E}_{q_\Phi^{(n)}}\big\{\log\big(\sigma_d^2(\vec{z};\Theta)\big)\big\} - \frac{D}{2}\log\big(2\pi e\big). \end{aligned}$$

Proof Sketch
The proof of Prop. 3 is considerably more intricate than those for Prop. 1 and 2. The main additional challenge is the integral over $\log(p_\Theta(\vec{x}^{(n)}\,|\,\vec{z}))$ (compare term $\mathcal{F}_2(\Theta,\Phi)$ in the proof of Prop. 1). A generalization is, however, possible using a reformulation of the integral in terms of the entropy of $p_\Theta(\vec{x}^{(n)}\,|\,\vec{z})$. Appendix B gives the full (long) derivation. The generalization can be thought of as similar to the proof of Prop. 1, which was presented in a form that makes it similar to the general case of Prop. 3. Derivations for the other terms ($\mathcal{F}_1(\Theta,\Phi)$ and $\mathcal{F}_2(\Theta,\Phi)$) as well as the reparametrization are as for Prop. 1 and Prop. 2. □

Considering Prop. 3, observe that the final result is concise and its application to a given VAE is straightforward (while the proof is long and technical). Also observe that Eqn. (39) is indeed a generalization of Eqn. (25): if we replace $\sigma_d^2(\vec{z};\Theta)$ by a scalar σ^2, then we drop back to Eqn. (25). As Prop. 2, the generalization of Prop. 3 applies for commonly encountered conditions. For *idealized* conditions, convergence to sums of entropies as in Prop. 3 can be shown relatively easily (see, e.g., Lücke and Henniges, 2012). However, idealized would in this context mean that four unrealistic conditions have to be fulfilled: (1) the data have to be distributed according to the used generative model; (2) the data set has to be infinitely large; (3) the variational distributions have to be equal to the posterior; and (4) learning has to converge to a global optimum. In contrast, Prop. 3 states that convergence to sums of entropies also follows in realistic conditions for all standard VAEs: for any (reasonable) finite or infinite data sets, for any stationary point, and for any variational distributions. We remark that Eqn. (39) can (like Prop. 2) also be used to more conveniently estimate the variational bound than the original bound (9). The expression (39) is in general not closed-form but the intractable integral only involves the DNN for the decoder variances. The dependence on entropies at stationary does again point to the key role played by the variance parameters: the entropies themselves do not depend on the DNNs for encoder or decoder means. However, the result of Prop. 3 differs from Prop. 2. Because of the expectation value w.r.t. the encoder $q_\Phi^{(n)}$ the bound at convergence does (in contrast to Prop. 2) also depend on the encoder DNN for the mean. We can shed more light on this dependence by observing that the average expectation w.r.t. $q_\Phi^{(n)}$ is actually an expectation w.r.t. one average distribution, i.e., the middle term of Eqn. (39) is equal to:

$$-\frac{1}{2}\sum_{d=1}^{D} \mathrm{E}_{\bar{q}_\Phi}\big\{\log\big(\sigma_d^2(\vec{z};\Theta)\big)\big\}, \text{ with } \bar{q}_\Phi = \frac{1}{N}\sum_n q_\Phi^{(n)}. \tag{40}$$

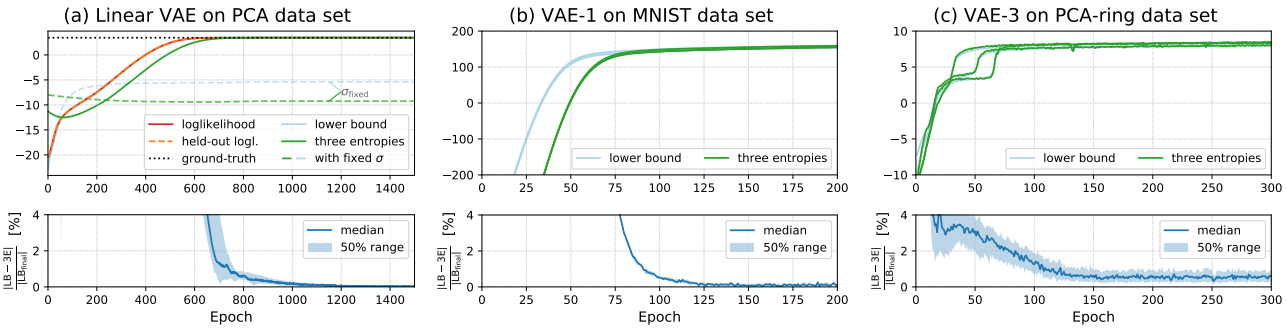


Figure 1: VAE models of increasing complexity (left to right) on different data sets. **Top plots:** Absolute values of the given bounds per data point of single runs. In (a) the lower bound is barely visible as it is covered by the log-likelihood (zoom in to see). **Bottom plots:** Median and interquartile range of the relative difference between the lower bound and the three entropy formulas for the respective VAEs over multiple runs.

If $q_\Phi^{(n)}$ models the true posteriors of the model well, the average posterior converges (for many data points) to the prior $p_\Theta(\vec{z})$. Consequently, the dependence on the DNN for the encoder *mean* disappears asymptotically for optimal $q_\Phi^{(n)}$ (and many data points). In that case, we recover (as for Prop. 2) that the value of the variational bound at convergence only depends on parameters for encoder and decoder variances.

Finally, note that Prop. 3 reflects the encoder and decoder parts of the lower bound more symmetrically than was apparent from Prop. 2 (only the expectation value of the middle term seems to break the symmetry). We may therefore be motivated to derive still more symmetric expressions. To do so, let us interpret the arguments of the logarithms of (39) as volumes covered by encoder and decoder distributions:

$$\text{z-Vol}(\vec{x};\Phi) = \prod_{h=1}^{H} \big(2\,\tau_h(\vec{x};\Phi)\big), \qquad (41)$$

$$\text{x-Vol}(\vec{z};\Theta) = \prod_{d=1}^{D} \big(2\,\sigma_d(\vec{z};\Theta)\big). \qquad (42)$$

$\text{z-Vol}(\vec{x};\Phi)$ is a measure of the volume covered by the encoder Gaussian $q_\Phi^{(n)}(\vec{z})$ in latent space ($\vec{z}$-space). Here we used a hyper-rectangle with two standard deviations edge length to define the volume. Analogously $\text{x-Vol}(\vec{z};\Theta)$ is the volume covered by the decoder distribution. If we now consider the limit of many data points, we arrive at a very symmetrical expression:

Corollary 2
Consider VAE–3 and let us denote by $p(\vec{x})$ the true data distribution. Then in the limit of many data points $N \to \infty$, it applies at convergence that

$$\mathcal{F}(\Phi,\Theta) = \text{E}_{p(\vec{x})}\,\big\{\log\big(\text{z-Vol}(\vec{x};\Phi)\big)\big\} - \text{E}_{\bar{q}_\Phi(\vec{z})}\big\{\log\big(\text{x-Vol}(\vec{z};\Theta)\big)\big\} + \text{ const.} \qquad (43)$$

Proof
We use $\frac{1}{N}\sum_n f(\vec{x}^{(n)}) \to \int p(\vec{x}) f(\vec{x}) d\vec{x}$ in the limit. Then elementary rearrangements of terms. The constant in (43) for the volume definitions (41) and (42) is given by const. $= (D-H)\log(2) - (D/2)\log(2\pi e)$. □

The definition of volumes covered by encoder and decoder distributions may seem a bit arbitrary. Instead of Eqns. (41) and (42) one could, e.g., have defined the volumes as those rectangles containing say 50% or 90% of a distribution's mass. Eqn. (43) does not change when changing the volume definitions, though. It is only important that $\text{z-Vol}(\vec{x};\Phi)$ and $\text{x-Vol}(\vec{z};\Theta)$ are measures of the covered volume. Any proportionality factors are simply changing the constant in Eqn. (43).

Considering Corollary 2, a VAE can thus be regarded as trying to balance two terms to achieve a high lower bound: it tries to make the log-volumes covered by its encoder distribution as large as possible while trying to make the log-volumes covered by its decoder distributions as small as possible.

3 Numerical Verification

Let us now consider some numerical experiments. They will serve as a verification of our analytical results and will point to some potential applications.

3.1 Linear VAE

The linear VAE given by Eqns. (29) to (32) allows for the most direct investigation of the result of Prop. 2. It has the advantage that we know the optimal solution in this case: it is given by the well-known maximum likelihood solution of p-PCA (Tipping and Bishop, 1999; Roweis, 1998). For the experiments we therefore first generate data according to the p-PCA generative model (details in Appendix C). In Fig. 1(a), top plot, we then compare the three entropies of Eqn. (33) with the standard lower

bound (9), estimated by sampling, and the known exact log-likelihood solution (Appendix A, Eqn. 44) for the same set of data points. For verification purposes, we additionally show the ground-truth log-likelihood for the generative parameters, as well as the model log-likelihood on held-out data.

Following the proof of Prop. 1 and thus Prop. 2, the lower bound (9) and the three entropies (33) only have to be identical at stationary points of the variance parameters. This suggests that e.g. for fixed σ^2 we can not expect the lower bound to converge to the three entropies. This is shown by the dashed lines in Fig. 1(a), top plot, where we fixed σ^2 to a suboptimal value.

3.2 Standard VAEs

To investigate more common use cases, we show in Fig. 1(b) experiments of a standard non-linear VAE (model VAE–1, Eqns. 1 to 5) on MNIST data. For verification of Prop. 3, we also show experiments with VAE–3 in Fig. 1(c), where we used the same PCA data as for the linear VAE but introduced an additional non-linear transformation on the data, projecting it onto a ring-like structure in the D-dimensional space (compare Doersch, 2016). Since no ground-truth log-likelihood is available for these models, we only show the standard lower bound (9), estimated by sampling, and compare it to the three entropies (25) and (39), respectively. More details and visualizations are provided in Appendix C.

3.3 Experimental Results

As can be observed, the sampled lower bound converges towards the three entropies when the variance parameters are converging towards stationary points. In Fig. 1(a), top, we also see that both the lower bound and the three entropies converge tight to the log-likelihood (with the lower bound mostly invisible under the log-likelihood) and recover the ground-truth well. The gap between lower bound and three entropies might become small even before all parameters converged, as seen, e.g., in Fig. 1(c), since only zero-gradients of the variances are required, including, e.g., saddle points or saddle surfaces, which commonly occur in DNNs. When σ is fixed to a suboptimal value (Fig. 1(a), top), the assumption for Props. 1 to 3 is not fulfilled, leading to a finite gap between lower-bound and the three entropies result at convergence.

The bottom plots show the absolute difference between the sampled lower bound and the three entropies relative to the final sampled lower bound (in percent). Shown is the median over 10 runs for Fig. 1(a) and (b) and over 100 runs for Fig. 1(c) with the interquartile range as shaded area, i.e., 50% of the deviations fall within this area. At convergence, the difference between lower bound and three entropies consistently approaches zero. Small gaps we can account primarily to fluctuations caused by finite learning rates, batch sizes and numbers of samples. E.g., for VAE–3, $\sigma_d(\vec{z};\Theta)$ has to be approximated via sampling and we confirmed that the small gap of $<0.5\%$ in Fig. 1(c), bottom, is in the order of magnitude of the sampling noise of $\sigma_d(\vec{z};\Theta)$, which makes the three entropies fluctuate around the lower bound (see also Appendix C, Fig. 3).

4 Discussion

Stationary points of learning objectives are of central importance for our understanding of learning algorithms. Their properties are consequently frequently studied in the literature (e.g. Frey, 1999; Tipping and Bishop, 1999; Yedidia et al., 2001; Opper and Saad, 2001; Lucas et al., 2019). Here we focused on variational autoencoders (VAEs) and the stationary points of their variational lower bounds. Although VAEs can involve complex DNNs, we have derived an exact analytical result: at stationary points the variational lower bound is identical to a sum of three entropies. The result does notably not require any idealized and unrealistic conditions. It applies precisely also under conditions met when standard VAEs are applied in practice: finite data sets, imperfect matches of generative models to data, and locally optimal parameters. All derived results notably also apply for encoder distributions which imperfectly match the true posteriors. The proofs for Props. 2 and 3 are technical, and it may be possible to find easier and/or more elegant proofs. However, we are not aware of any results similar to Props. 2 and 3 for VAEs or related models, nor about any discussions that VAEs may converge to sums of entropies under commonly encountered conditions.

We believe that based on Props. 2 and 3 further results can be derived. Corollary 2 may serve as an example. Furthermore, applications of the propositions to study learning in VAEs are conceivable which may include studying overfitting, mode collapse, analyzing the effect of annealing, etc. For instance, Props. 2 and 3 may be useful to generalize results of Lucas et al. (2019) to non-linear VAEs. We remark in this context that a main result of Lucas et al. (2019) is that additional local optima emerge if the decoder variance is fixed during learning – which is precisely when a VAE does not converge to a sum of entropies (compare Fig. 1(a), top). Future work may, furthermore, focus on generalizing Prop. 3 still fur-

ther. VAEs defined by Gaussian distributions are by far the most wide-spread, which makes the here derived results cover very large parts of VAE research. However, other, non-Gaussian VAEs have also been of interest. For some (e.g., VAEs with discrete latents Rolfe, 2016; Van Den Oord et al., 2017; Khoshaman et al., 2018; Sadeghi et al., 2019; Vahdat et al., 2018; Lorberbom et al., 2019) the generalization of our derivations is (considering the proofs of Props. 1 and 2) straight-forward. For others (e.g. VAEs with non-Gaussian observables) derivations will require non-trivial future generalizations.

References

P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

C. Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

G. Dorta, S. Vicente, L. Agapito, N. D. Campbell, and I. Simpson. Training VAEs under structured residuals. *arXiv preprint arXiv:1804.01050*, 2018.

W. Freeman and Y. Weiss. On the fixed points of the max-product algorithm. *IEEE Trans. on Info. Theory*, 2000.

B. Frey. Turbo factor analysis. *Adv. Neural Information Processing*, 1999.

A. Khoshaman, W. Vinci, B. Denis, E. Andriyash, H. Sadeghi, and M. H. Amin. Quantum variational autoencoder. *Quantum Science and Technology*, 4(1): 014001, 2018.

D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

D. Kunin, J. M. Bloom, A. Goeva, and C. Seed. Loss landscapes of regularized linear autoencoders. *arXiv preprint arXiv:1901.08168*, 2019.

G. Lorberbom, A. Gane, T. Jaakkola, and T. Hazan. Direct optimization through arg max for discrete variational auto-encoder. In *Advances in Neural Information Processing Systems*, pages 6203–6214, 2019.

J. Lucas, G. Tucker, R. B. Grosse, and M. Norouzi. Don't blame the ELBO! A linear VAE perspective on posterior collapse. In *Advances in Neural Information Processing Systems*, pages 9408–9418, 2019.

J. Lücke and M. Henniges. Closed-form entropy limits - A tool to monitor likelihood optimization of probabilistic generative models. In *Artificial Intelligence and Statistics*, pages 731–740. PMLR, 2012.

T. Monk, C. Savin, and J. Lücke. Neurons equipped with intrinsic plasticity learn stimulus intensity statistics. In *Advances in Neural Information Processing Systems*, volume 29, pages 4278–4286, 2016.

R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*. Kluwer, 1998.

A. V. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 1747–1756, 2016.

M. Opper and D. Saad. *Advanced mean field methods: Theory and practice*. MIT press, 2001.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

J. T. Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

S. T. Roweis. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems*, pages 626–632, 1998.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

H. Sadeghi, E. Andriyash, W. Vinci, L. Buffoni, and M. H. Amin. PixelVAE++: Improved PixelVAE with discrete prior. *arXiv preprint arXiv:1908.09948*, 2019.

M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B*, 61, 1999.

A. Vahdat, W. G. Macready, Z. Bian, A. Khoshaman, and E. Andriyash. DVAE++: Discrete variational autoencoders with overlapping transformations. *arXiv preprint arXiv:1802.04920*, 2018.

A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, pages 6306–6315, 2017.

Y. Weiss and W. T. Freeman. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. In *Advances in Neural Information Processing Systems*, pages 673–679, 2000.

M. Welling and Y. W. Teh. Approximate inference in Boltzmann machines. *Artificial Intelligence*, 143(1):19–50, 2003.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems*, pages 689–695, 2001.

Supplementary Materials

Appendix A: Linear VAEs

For the proof of Corollary 1 not that for the linear DNN (30) the same reparametrization is possible as we used it for VAE–1 (i.e., using $\tilde{W}$ with constraint columns instead of W in Eqn. 30). Furthermore, no conditions were imposed on the encoder DNN for Prop. 2, and the used properties of the stationary points (19) are the same for the linear VAE as for the VAEs above. Prop. 2 thus applies for the linear VAE as a special case of VAE–1.

Regarding the result, Eqn. (33), note that it highlights some properties of the variational bound at convergence. First, note that the variational bound at the stationary points can be computed efficiently and solely based on the variance parameters σ^2 and τ_h^2. For linear VAEs, the bound is even independent of the data points, i.e., just the $H+1$ variance parameters determine its value. In addition to this simplification, the linear VAE has a further property that makes it interesting from a theoretical perspective. As recently shown (Lucas et al., 2019), linear VAEs can be used to recover the maximum likelihood solution of probabilistic PCA (p-PCA). First, Lucas et al. (2019) showed that no stable stationary points of the variational lower bound exist other then the global maximum. Training of the linear VAE will thus always converge to the global maximum of the lower bound. Second, they showed that the linear encoder is flexible enough to finally recover full posteriors exactly, which makes the variational lower bound tight at convergence. As the decoder is identical to the generative model of p-PCA (Tipping and Bishop, 1999), the linear VAE thus converges to recover the optimal p-PCA likelihood. According to Corollary 1 it thus applies that after convergence of the linear VAE, Eqn. (33) is equal to the p-PCA log-likelihood.

We can combine the earlier results (Tipping and Bishop, 1999; Lucas et al., 2019) with Corollary 1 and obtain the following.

Corollary A1
Consider the linear VAE defined by Eqns. (29) to (32) with decoder parameters $\Theta = (\sigma^2, W, \vec{\mu}_0)$. Then after convergence, the parameters $(\sigma^2, W, \vec{\mu}_0)$ represent the maximum likelihood solution for p-PCA, and the value of the log-likelihood $\mathcal{L}(\Theta)$ is given by:

$$\mathcal{L}(\Theta) = -\frac{D}{2}\log\big(2\pi e\sigma^2\big) + \frac{1}{2}\sum_h \log\big(\tau_h^2\big), \tag{44}$$

where τ_h^2 are the learned variances of the VAE encoder.

Proof
As Eqn. (33) applies for all stationary points, it also applies for the global maximum of the variational lower bound. At the global maximum, the variational bound is equal to the p-PCA log-likelihood (result by Lucas et al., 2019), which proves the claim. □

Other than for standard non-linear VAEs, the existence of a closed-form result is itself not surprising for the linear VAE. The linearities make analytic solutions of the integrals of the variational bound possible. Indeed, we can instead of Eqn. (33) simply use the well-known closed-form solution of the p-PCA likelihood (Tipping and Bishop, 1999):

$$\mathcal{L}(\Theta) = -\frac{D}{2}\log(2\,\pi) - \frac{1}{2}\log\big(\det(C)\big) - \frac{1}{2}\,\mathrm{Tr}\,(C^{-1}S), \tag{45}$$

where $C = WW^T + \sigma^2\mathbb{1}$ and where S is the data covariance matrix $S = \frac{1}{N}\sum_n (\vec{x}^{(n)} - \vec{\mu})(\vec{x}^{(n)} - \vec{\mu})^T$.

At convergence, we thus have two alternatives to compute the log-likelihood: Eqn. (45) and Eqn. (44). The two expressions are very different, however. While both are closed-form, the well-known p-PCA likelihood (45) requires the data in the form of the data covariance matrix S. And even neglecting the computational effort to compute S, the computation of (45) is much more expensive than computing (44): For $\mathcal{L}(\Theta)$ the inverse and the determinant of the $D \times D$ matrix C have to be computed. In contrast, for the computation of $\mathcal{F}(\Phi, \Theta)$ in (44) only computations of $\mathcal{O}((H+1)D)$ are required (and no data). The derived result of Eqn. (44) can therefore be of theoretical and practical relevance especially considering the exceptionally widespread use of PCA in Machine Learning, Statistics and beyond. For instance, PCA is increasingly frequently applied in streaming setting especially in recent years. A linear VAE may provide a promising alternative to other PCA algorithms as it can easily be applied to streaming data. The value of the likelihood itself is then frequently of importance, e.g., for monitoring the fit of the data or for the selection of PCA dimensions. In this context, Eqn. (44) provides

an exceedingly easy way to compute the log-likelihood value. Also if the streamed data changes, the parameters of the linear VAE will change and Eqn. (44) can be used to track these changes without any effort. No data has to be kept in memory and no costly computations are required.

For the purposes of this paper, our first numerical experiments will use the linear VAE. The result can then directly be compared to the well known closed-form log-likelihood (45) as well as to sampling based approximations of the variational lower bound.

Appendix B: Proof of Proposition 3

We have deliberately stated the proof of Props. 1 and 2 in a form that, we hope, makes the relatively intricate and long proof of Prop. 3 easier to follow. As a prelude, observe that the reformulation of $\mathcal{F}_2$ in Eqn. (21) in the proof of Prop. 1 is a key step: we formulate $\mathcal{F}_2$ in terms of an entropy plus an additional term which converges to zero at stationary points. Derivatives w.r.t. σ^2 of entropy and the other term could be moved out as a factor which allowed for deriving the final result. While proofs for scalar σ^2 other than the one in Prop. 1 are possible, a generalization as required for Prop. 3 will best make use of a generalization of the expression Eqn. (21) used for the proof of Prop. 1. As for the proof of Prop. 1, we consider separately the three terms of the variational bound $\mathcal{F}(\Phi, \Theta)$:

$$\mathcal{F}(\Phi, \Theta) = \mathcal{F}_1(\Phi, \Theta) + \mathcal{F}_2(\Phi, \Theta) + \mathcal{F}_3(\Phi). \tag{46}$$

For $\vec{z}$-dependent decoder variances, the main difference to Prop. 1 lies in the term $\mathcal{F}_2(\Phi, \Theta)$, which is given for VAE–3 by:

$$\mathcal{F}_2(\Phi, \Theta) = \frac{1}{N} \sum_{n=1}^{N} \sum_{d=1}^{D} \int q_\Phi^{(n)}(\vec{z}) \log\big(\mathcal{N}\big(x_d^{(n)}; \mu_d(\vec{z}; \Theta), \sigma_d^2(\vec{z}; \Theta)\big)\big) \,\mathrm{d}\vec{z}$$

We now use the parametrization of the Gaussian in terms of its natural parameters $\vec{\eta}$ (of the exponential family), which will turn out to be convenient:

$$\mathcal{F}_2(\Phi, \Theta) = \frac{1}{N} \sum_{n=1}^{N} \sum_{d=1}^{D} \int q_\Phi^{(n)}(\vec{z}) \Big(\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z}; \Theta\big) \vec{T}\big(x_d^{(n)}\big) - A\big(\vec{\eta}_d(\vec{z}; \Theta)\big) - \frac{1}{2}\log(2\pi)\Big) \mathrm{d}\vec{z} \tag{47}$$

where

$$\vec{\eta}_d(\vec{z}; \Theta) \;=\; \begin{pmatrix} \frac{\mu_d(\vec{z};\Theta)}{\sigma_d^2(\vec{z};\Theta)} \\ -\frac{1}{2\sigma_d^2(\vec{z};\Theta)} \end{pmatrix}, \qquad \vec{T}(x_d^{(n)}) \;=\; \begin{pmatrix} x_d^{(n)} \\ (x_d^{(n)})^2 \end{pmatrix}, \tag{48}$$

$$A\big(\vec{\eta}_d(\vec{z}; \Theta)\big) \;=\; -\frac{\big(\eta_d^{(1)}(\vec{z}; \Theta)\big)^2}{4\eta_d^{(2)}(\vec{z}; \Theta)} - \frac{1}{2}\log\big(-2\eta_d^{(2)}(\vec{z}; \Theta)\big), \tag{49}$$

and where $-\frac{1}{2}\log(2\pi)$ is the (log) base measure.

Now we rewrite $A\big(\vec{\eta}_d(\vec{z}; \Theta)\big)$ in terms of the entropy of $p_\Theta(x_d \,|\, \vec{z})$ and other terms. For this we use the definition of the entropy and obtain using the parametrization with natural parameters:

$$\begin{aligned}
\mathcal{H}[p_\Theta(x_d \,|\, \vec{z})] \;&=\; -\int p_\Theta(x_d \,|\, \vec{z}) \log\big(p_\Theta(x_d \,|\, \vec{z})\big) \mathrm{d}x_d \\
&=\; -\int p_\Theta(x_d \,|\, \vec{z}) \Big(-\frac{1}{2}\log(2\pi) + \vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z}; \Theta\big) \vec{T}\big(x_d^{(n)}\big) - A\big(\vec{\eta}_d(\vec{z}; \Theta)\big)\Big) \mathrm{d}x_d \\
&=\; -\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z}; \Theta\big)\; \mathrm{E}_{p_\Theta(x_d \,|\, \vec{z})}\big\{\vec{T}\big(x_d^{(n)}\big)\big\} + A\big(\vec{\eta}_d(\vec{z}; \Theta)\big) + \frac{1}{2}\log(2\pi) \\
&=\; -\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z}; \Theta\big)\; \frac{\mathrm{d}}{\mathrm{d}\vec{\eta}} A(\vec{\eta})\big|_{\vec{\eta}=\vec{\eta}_d(\vec{z};\Theta)} + A\big(\vec{\eta}_d(\vec{z}; \Theta)\big) + \frac{1}{2}\log(2\pi)
\end{aligned} \tag{50}$$

where we have used the standard equality $\mathrm{E}_{p(x|\vec{\eta})}\big\{\vec{T}(x)\big\} = \frac{\mathrm{d}}{\mathrm{d}\vec{\eta}} A(\vec{\eta})$ which applies for the Gaussian as exponential family distribution. We further abbreviate using $\vec{A}' = \frac{\mathrm{d}}{\mathrm{d}\vec{\eta}} A(\vec{\eta})$ and obtain:

$$-A\big(\vec{\eta}_d(\vec{z}; \Theta)\big) - \frac{1}{2}\log(2\pi) \;=\; -\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z}; \Theta\big)\; \vec{A}'\big(\vec{\eta}_d(\vec{z}; \Theta)\big) - \mathcal{H}[p_\Theta(x_d \,|\, \vec{z})]$$

By inserting this expression into (47) we can consequently express $\mathcal{F}_2(\Phi,\Theta)$ as follows:

$$\mathcal{F}_2(\Phi,\Theta) = \frac{1}{N}\sum_{n=1}^{N}\sum_{d=1}^{D}\int q_\Phi^{(n)}(\vec{z})\Big(\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\Theta\big)\ \vec{T}\big(x_d^{(n)}\big) - \ \vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\Theta\big)\ \vec{A}'\big(\vec{\eta}_d\big(\vec{z};\Theta\big)\big) - \mathcal{H}[p_\Theta(x_d\,|\,\vec{z})]\Big)\mathrm{d}\vec{z}. \tag{51}$$

Expression (51) is the generalization of (21) for general $\sigma_d^2(\vec{z};\Theta)$. As for the proof of Prop. 1, we now seek to show that all terms of (51) except the one depending on $\mathcal{H}[p_\Theta(x_d\,|\,\vec{z})]$ are zero at stationary points.

Let us first abbreviate the networks DNN_μ and DNN_σ of the decoder (Eqns. 37 and 38) as follows:

$$\mu_d(\vec{z};\Theta) \ = \ \big(\mathrm{DNN}_\mu(\vec{z};W)\big)_d \ = \ \sum_i W_{di}^{(L)}\,\varphi_i^\mu(\vec{z};\Theta) \tag{52}$$

$$\sigma_d^2(\vec{z};\Theta) \ = \ \big(\mathrm{DNN}_\sigma(\vec{z};M)\big)_d \ = \ \sum_j M_{dj}^{(L)}\,\varphi_j^\sigma(\vec{z};\Theta), \tag{53}$$

where $\varphi_i^\mu(\vec{z};\Theta)$ and $\varphi_j^\sigma(\vec{z};\Theta)$ simply stand for the remainder of the DNN after the lowest weights are removed. We demanded that the decoder networks both have at least one hidden layer such that $L \geq 1$, i.e., the weights $W^{(L)}$ and $M^{(L)}$ are different from the weight matrices $W^{(0)}$ and $M^{(0)}$. Hence, we do not have to worry about any constraints that are potentially introduced for $W^{(0)}$ and $M^{(0)}$. At all stationary points of the variational lower bound $\mathcal{F}(\Phi,\Theta)$ it thus applies:

$$\frac{\mathrm{d}}{\mathrm{d}W^{(L)}}\mathcal{F}(\Phi,\Theta) \ = \ \frac{\mathrm{d}}{\mathrm{d}W^{(L)}}\mathcal{F}_2(\Phi,\Theta) \ = \ 0, \quad \text{and} \quad \frac{\mathrm{d}}{\mathrm{d}M^{(L)}}\mathcal{F}(\Phi,\Theta) \ = \ \frac{\mathrm{d}}{\mathrm{d}M^{(L)}}\mathcal{F}_2(\Phi,\Theta) \ = \ 0. \tag{54}$$

The derivative w.r.t. $\mathcal{F}_2(\Phi,\Theta)$ is zero as the other terms of $\mathcal{F}(\Phi,\Theta)$ (i.e., $\mathcal{F}_1(\Phi,\Theta)$ and $\mathcal{F}_3(\Phi)$) do not depend on $W^{(L)}$ and $M^{(L)}$. Let us now arrange all entries of $W^{(L)}$ and $M^{(L)}$ into one long vector which we shall call $\vec{R}$. Let us denote by $\vec{R}^*$ a fixed stationary point that we consider. All the following derivations can be considered to apply for one such stationary point $\vec{R}^*$. But as any stationary point $\vec{R}^*$ can be considered, all the following derivations will apply for all possible stationary points $\vec{R}^*$.

Let us now borrow methodology from variational calculus, i.e., we use that for arbitrary parameters $\tilde{\vec{R}}$ the following applies at the stationary points $\vec{R}^*$:

$$\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{F}_2(\Phi,\vec{R}^*+\epsilon\,\tilde{\vec{R}})\big|_{\epsilon=0} \ = \ 0, \tag{55}$$

where we neglected all parameters Θ other than W and M in the argument of $\mathcal{F}_2$ for readability. We will also do so in the following, and furthermore will write $p(x_d\,|\,\vec{z};\vec{R})$ instead of $p_\Theta(x_d\,|\,\vec{z})$.

Eqn. (55) combines the above conditions (54), and we can proceed by taking derivatives w.r.t. the scalar ϵ. Inserting the expression (51) for $\mathcal{F}_2$ into (55), we obtain:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{F}_2(\Phi,\vec{R}^*+\epsilon\,\tilde{\vec{R}})\\
&= \frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\frac{\mathrm{d}}{\mathrm{d}\epsilon}\Big[\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\,\vec{T}\big(x_d^{(n)}\big) - \vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\,\vec{A}'\big(\vec{\eta}_d\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\big)\\
&\qquad\qquad - \mathcal{H}[p(x_d\,|\,\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}})]\Big]\mathrm{d}\vec{z}\\
&= \frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\Big[\Big(\frac{\mathrm{d}}{\mathrm{d}\epsilon}\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\Big)\,\vec{T}\big(x_d^{(n)}\big) - \Big(\frac{\mathrm{d}}{\mathrm{d}\epsilon}\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\Big)\,\vec{A}'\big(\vec{\eta}_d\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\big)\\
&\qquad\qquad - \vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\,\Big(\frac{\mathrm{d}}{\mathrm{d}\epsilon}\vec{A}'\big(\vec{\eta}_d\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\big)\Big) - \frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{H}[p(x_d\,|\,\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}})]\Big]\ \mathrm{d}\vec{z}.
\end{aligned} \tag{56}$$

The derivative of the entropy can now be determined using the expression (50) that we already used above:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{H}[p(x_d\,|\,\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}})] \ &= \ -\frac{\mathrm{d}}{\mathrm{d}\epsilon}\Big(\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\ \vec{A}'(\vec{\eta}_d(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}))\Big) + \frac{\mathrm{d}}{\mathrm{d}\epsilon}A\big(\vec{\eta}_d(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}})\big)\\
&= \ -\Big(\frac{\mathrm{d}}{\mathrm{d}\epsilon}\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\Big)\ \vec{A}'(\vec{\eta}_d(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}))\\
&\qquad - \vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\ \frac{\mathrm{d}}{\mathrm{d}\epsilon}\vec{A}'\big(\vec{\eta}_d(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}})\big) + \frac{\mathrm{d}}{\mathrm{d}\epsilon}A\big(\vec{\eta}_d(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}})\big).
\end{aligned} \tag{57}$$

By inserting this expression back into (56), the two first terms (importantly) cancel with two terms of (56) and we obtain:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{F}_2(\Phi,\vec{R}^*+\epsilon\,\tilde{\vec{R}})\\
&\quad=\frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\Big[\Big(\frac{\mathrm{d}}{\mathrm{d}\epsilon}\vec{\eta}_d^{\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\Big)\vec{T}\big(x_d^{(n)}\big)-\frac{\mathrm{d}}{\mathrm{d}\epsilon}A\big(\vec{\eta}_d(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}})\big)\Big]\,\mathrm{d}\vec{z}\\
&\quad=\frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\Big[\Big(\frac{\mathrm{d}}{\mathrm{d}\epsilon}\vec{\eta}_d^{\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\Big)\vec{T}\big(x_d^{(n)}\big)-\Big(\frac{\mathrm{d}}{\mathrm{d}\epsilon}\vec{\eta}_d^{\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\Big)\vec{A}'(\vec{\eta}(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}))\Big]\,\mathrm{d}\vec{z}\\
&\quad=\frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\Big(\frac{\mathrm{d}}{\mathrm{d}\epsilon}\vec{\eta}_d^{\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\Big)\Big[\vec{T}\big(x_d^{(n)}\big)-\vec{A}'(\vec{\eta}_d(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}))\Big]\,\mathrm{d}\vec{z}
\end{aligned}$$

The derivative w.r.t. ϵ is in the above expression now restricted to $\vec{\eta}_d$. If we evaluate at stationary points, i.e. for $\epsilon=0$, we obtain:

$$0\;=\;\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{F}_2(\Phi,\vec{R}^*+\epsilon\,\tilde{\vec{R}})\big|_{\epsilon=0}\;=\;\frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\,\Big(\frac{\mathrm{d}}{\mathrm{d}\epsilon}\vec{\eta}_d^{\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\big|_{\epsilon=0}\Big)\Big[\vec{T}\big(x_d^{(n)}\big)-\vec{A}'(\vec{\eta}_d(\vec{z};\vec{R}^*))\Big]\,\mathrm{d}\vec{z}.\quad(58)$$

Expression (58) can be compared with expression (51) which expresses the lower bound (for any parameters) in terms of the entropy. The part of (51) which is not depending on the entropy is given by

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{d=1}^{D}\int q_\Phi^{(n)}(\vec{z})\;\vec{\eta}_d^{\mathrm{T}}\big(\vec{z};\vec{R}\big)\;\Big[\vec{T}\big(x_d^{(n)}\big)-\;\vec{A}'\big(\vec{\eta}_d\big(\vec{z};\vec{R}\big)\big)\Big]\mathrm{d}\vec{z},\qquad(59)$$

If we can show that (59) is zero at stationary points $\vec{R}^*$, then we have proven that $\mathcal{F}_2$ converges to an expected entropy (as desired). However, we can not directly conclude from (58) that (51) is zero – the middle factor differs. On the other hand, (58) applies for any vector $\tilde{\vec{R}}$. In the following, we will therefore try to choose $\tilde{\vec{R}}$ in a way that allows to conclude that (59) is zero. As preparation, let us reformulate (58) as follows:

$$\begin{aligned}
0\;&=\;\frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{F}_2(\Phi,\vec{R}^*+\epsilon\,\tilde{\vec{R}})\big|_{\epsilon=0}\\
&=\frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\Big(\frac{\mathrm{d}}{\mathrm{d}\epsilon}\vec{\eta}_d^{\mathrm{T}}\big(\vec{z};\vec{R}^*+\epsilon\,\tilde{\vec{R}}\big)\big|_{\epsilon=0}\Big)\Big[\vec{T}\big(x_d^{(n)}\big)-\vec{A}'(\vec{\eta}_d(\vec{z};\vec{R}^*))\Big]\,\mathrm{d}\vec{z}\\
&=\frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\begin{pmatrix}\big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_d^{(1)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\big)\tilde{\vec{R}}\\ \big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_d^{(2)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\big)\tilde{\vec{R}}\end{pmatrix}^{\mathrm{T}}\Big[\vec{T}\big(x_d^{(n)}\big)-\vec{A}'(\vec{\eta}_d(\vec{z};\vec{R}^*))\Big]\,\mathrm{d}\vec{z}\\
&=\quad\frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\,\Big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_d^{(1)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\Big)\,\tilde{\vec{R}}\,\Big(T^{(1)}\big(x_d^{(n)}\big)-\frac{\mathrm{d}}{\mathrm{d}\eta^{(1)}}A(\vec{\eta})\big|_{\vec{\eta}=\vec{\eta}_d(\vec{z};\vec{R}^*)}\Big)\,\mathrm{d}\vec{z}\\
&\quad+\frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\,\Big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_d^{(2)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\Big)\,\tilde{\vec{R}}\,\Big(T^{(2)}\big(x_d^{(n)}\big)-\frac{\mathrm{d}}{\mathrm{d}\eta^{(2)}}A(\vec{\eta})\big|_{\vec{\eta}=\vec{\eta}_d(\vec{z};\vec{R}^*)}\Big)\,\mathrm{d}\vec{z}\qquad(60)
\end{aligned}$$

We now defined the vector $\tilde{\vec{R}}$ to contain all elements of W and M in one (long) row vector. To further evaluate (60), we have to evaluate the derivatives of $\eta_d^{(1)}(\vec{z};\vec{R})$ and $\eta_d^{(2)}(\vec{z};\vec{R})$ w.r.t. the all entries. Eqn. (48) describes the relation between the natural parameters $\eta_d^{(1)}(\vec{z};\vec{R})$ and $\eta_d^{(2)}(\vec{z};\vec{R})$ and the decoder DNNs $\mu_d(\vec{z};\vec{R})$ and $\sigma_d^2(\vec{z};\vec{R})$. If we use for the DNNs abbreviations (52) and (53), we can evaluate the derivatives. We demanded

two separate sets of parameters for the DNNs, i.e., $\mu_d(\vec{z};\vec{R})$ only depends on W and $\sigma_d^2(\vec{z};\vec{R})$ only depends on M. We therefore obtain:

$$\begin{aligned}
&\tfrac{\mathrm{d}}{\mathrm{d}W_{di}^{(L)}}\eta_{d'}^{(1)}(\vec{z};\vec{R}) \\
&= \tfrac{1}{\sigma_{d'}^2(\vec{z};\vec{R})}\,\tfrac{\mathrm{d}}{\mathrm{d}W_{di}^{(L)}}\mu_{d'}(\vec{z};\vec{R}) \\
&= \tfrac{1}{\sigma_{d'}^2(\vec{z};\vec{R})}\,\tfrac{\mathrm{d}}{\mathrm{d}W_{di}^{(L)}}\textstyle\sum_{i'} W_{d'i'}^{(L)}\,\varphi_{i'}^{\mu}(\vec{z};\vec{R}) \\
&= \delta_{dd'}\,\tfrac{1}{\sigma_d^2(\vec{z};\vec{R})}\,\varphi_i^{\mu}(\vec{z};\vec{R}),
\end{aligned}
\qquad
\begin{aligned}
&\tfrac{\mathrm{d}}{\mathrm{d}M_{dj}^{(L)}}\eta_{d'}^{(1)}(\vec{z};\vec{R}) \\
&= \mu_d(\vec{z};\vec{R})\,\tfrac{\mathrm{d}}{\mathrm{d}M_{dj}^{(L)}}\tfrac{1}{\sigma_{d'}^2(\vec{z};\vec{R})} \\
&= -\tfrac{\mu_d(\vec{z};\vec{R})}{\sigma_{d'}^4(\vec{z};\vec{R})}\,\tfrac{\mathrm{d}}{\mathrm{d}M_{dj}^{(L)}}\textstyle\sum_{j'} M_{d'j'}^{(L)}\,\varphi_{j'}^{\sigma}(\vec{z};\vec{R}) \\
&= -\delta_{dd'}\,\tfrac{\mu_d(\vec{z};\vec{R})}{\sigma_d^4(\vec{z};\vec{R})}\,\varphi_j^{\sigma}(\vec{z};\vec{R}),
\end{aligned}$$

$$\begin{aligned}
&\tfrac{\mathrm{d}}{\mathrm{d}W_{di}^{(L)}}\eta_{d'}^{(2)}(\vec{z};\vec{R}) \\
&= -\tfrac{1}{2}\,\tfrac{\mathrm{d}}{\mathrm{d}W_{di}^{(L)}}\tfrac{1}{\sigma_{d'}^2(\vec{z};\vec{R})} \\
&= 0
\end{aligned}
\qquad
\begin{aligned}
&\tfrac{\mathrm{d}}{\mathrm{d}M_{dj}^{(L)}}\eta_{d'}^{(2)}(\vec{z};\vec{R}) \\
&= -\tfrac{1}{2}\,\tfrac{\mathrm{d}}{\mathrm{d}M_{dj}^{(L)}}\tfrac{1}{\sigma_{d'}^2(\vec{z};\vec{R})} \\
&= \tfrac{1}{2}\tfrac{1}{\sigma_{d'}^4(\vec{z};\vec{R})}\,\tfrac{\mathrm{d}}{\mathrm{d}M_{dj}^{(L)}}\textstyle\sum_{j'} M_{d'j'}^{(L)}\,\varphi_{j'}^{\sigma}(\vec{z};\vec{R}) \\
&= \tfrac{1}{2}\delta_{dd'}\tfrac{1}{\sigma_d^4(\vec{z};\vec{R})}\,\varphi_j^{\sigma}(\vec{z};\vec{R})
\end{aligned}$$

Using these expressions, we can (considering Eqn. 60) evaluate the terms

$$\Big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_d^{(1)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\Big)\,\tilde{\vec{R}} \quad\text{and}\quad \Big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_d^{(2)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\Big)\,\tilde{\vec{R}} \tag{61}$$

for specifically chosen $\tilde{\vec{R}}$ (keep in mind that Eqn. 60 holds for any $\tilde{\vec{R}}$). Let us choose $\tilde{\vec{R}}$ to be zero almost everywhere, only those entries for $W_{di}^{(L)}$ we keep non-zero. We then obtain:

$$\Big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_{d'}^{(1)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\Big)\,\tilde{\vec{R}} \;=\; \sum_d\sum_i\Big(\frac{\mathrm{d}}{\mathrm{d}W_{di}^{(L)}}\,\eta_{d'}^{(1)}(\vec{z};\vec{R})\Big)\big|_{\vec{R}=\vec{R}^*}\,W_{di}^{(L)} \tag{62}$$

$$= \sum_d\sum_i\Big(\delta_{dd'}\,\frac{1}{\sigma_d^2(\vec{z};\vec{R}^*)}\,\varphi_i^{\mu}(\vec{z};\vec{R}^*)\Big)\,W_{di}^{(L)} \;=\; \frac{1}{\sigma_{d'}^2(\vec{z};\vec{R}^*)}\sum_i W_{d'i}^{(L)}\,\varphi_i^{\mu}(\vec{z};\vec{R}^*) \tag{63}$$

$$= \frac{1}{\sigma_{d'}^2(\vec{z};\vec{R}^*)}\mu_{d'}(\vec{z};\vec{R}^*) \;=\; \eta_{d'}^{(1)}(\vec{z};\vec{R}^*), \quad\text{and further} \tag{64}$$

$$\Big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_{d'}^{(2)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\Big)\,\tilde{\vec{R}} \;=\; \sum_d\sum_i\Big(\frac{\mathrm{d}}{\mathrm{d}W_{di}^{(L)}}\,\eta_{d'}^{(2)}(\vec{z};\vec{R})\Big)\big|_{\vec{R}=\vec{R}^*}\,W_{di}^{(L)} \;=\; 0, \tag{65}$$

where for the line (63) we have chosen $W_{di}^{(L)}$ not to be arbitrary non-zero entries of $\tilde{\vec{R}}$ but those at the stationary points (we are free to choose any value). The reason why we can relate the derivative of $\eta_{d'}^{(1)}$ to $\eta_{d'}^{(1)}$ itself is that the lowest stage of both DNNs is a linear operation. Consequently, the relation can not necessarily be derived for general functions $\mu_d(\vec{z};\vec{R})$ and $\sigma_d^2(\vec{z};\vec{R})$ but it can be fulfilled for the standard DNNs we consider. By inserting (64) and (65) into (60) we obtain that at stationary points $\vec{R}^*$ holds:

$$0 \;=\; \frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{F}_2(\Phi,\vec{R}^*+\epsilon\,\tilde{\vec{R}})\big|_{\epsilon=0} = \frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\;\eta_d^{(1)}(\vec{z};\vec{R}^*)\,\Big(T^{(1)}\big(x_d^{(n)}\big) - \frac{\mathrm{d}}{\mathrm{d}\eta^{(1)}}A(\vec{\eta})\big|_{\vec{\eta}=\vec{\eta}_d(\vec{z};\vec{R}^*)}\Big)\,\mathrm{d}\vec{z} \tag{66}$$

By comparing with (59) we observe that the first summand of the integrand is zero at $\vec{R}^*$. However, we also have to show that the second summand (with $\eta_d^{(2)}(\vec{z};\vec{R}^*)$) is zero. We still have the liberty to choose other $\tilde{\vec{R}}$

for (60). A complementary choice to the $\tilde{\vec{R}}$ chosen above would be to choose all entries of $\tilde{\vec{R}}$ to be zero except of those corresponding to the entries of $M_{dj}^{(L)}$. If we do so, we obtain:

$$\Big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_{d'}^{(1)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\Big)\,\tilde{\vec{R}} \;=\; \sum_d\sum_j\Big(\frac{\mathrm{d}}{\mathrm{d}M_{dj}^{(L)}}\,\eta_{d'}^{(1)}(\vec{z};\vec{R})\Big)\big|_{\vec{R}=\vec{R}^*}\;M_{dj}^{(L)} \tag{67}$$

$$=\; \sum_d\sum_j\Big(-\delta_{dd'}\,\frac{\mu_d(\vec{z};\vec{R})}{\sigma_d^4(\vec{z};\vec{R})}\,\varphi_j^\sigma(\vec{z};\vec{R})\Big)\;M_{dj}^{(L)} \;=\; -\frac{\mu_{d'}(\vec{z};\vec{R})}{\sigma_{d'}^4(\vec{z};\vec{R})}\sum_j M_{d'j}^{(L)}\,\varphi_j^\sigma(\vec{z};\vec{R}) \tag{68}$$

$$=\; -\frac{\mu_{d'}(\vec{z};\vec{R})}{\sigma_{d'}^4(\vec{z};\vec{R})}\sigma_{d'}^2(\vec{z};\vec{R}) \;=\; -\frac{\mu_{d'}(\vec{z};\vec{R})}{\sigma_{d'}^2(\vec{z};\vec{R})} \;=\; -\eta_{d'}^{(1)}(\vec{z};\vec{R}^*),\quad\text{and further} \tag{69}$$

$$\Big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_{d'}^{(2)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\Big)\,\tilde{\vec{R}} \;=\; \sum_d\sum_j\Big(\frac{\mathrm{d}}{\mathrm{d}M_{dj}^{(L)}}\,\eta_{d'}^{(2)}(\vec{z};\vec{R})\Big)\big|_{\vec{R}=\vec{R}^*}\;M_{dj}^{(L)} \tag{70}$$

$$=\; \sum_d\sum_j\Big(\frac{1}{2}\delta_{dd'}\,\frac{1}{\sigma_d^4(\vec{z};\vec{R}^*)}\,\varphi_j^\sigma(\vec{z};\vec{R}^*)\Big)\;M_{dj}^{(L)} \;=\; \frac{1}{2}\frac{1}{\sigma_{d'}^4(\vec{z};\vec{R}^*)}\sum_j M_{d'j}^{(L)}\,\varphi_j^\sigma(\vec{z};\vec{R}^*) \tag{71}$$

$$=\; \frac{1}{2}\frac{1}{\sigma_{d'}^4(\vec{z};\vec{R}^*)}\;\sigma_{d'}^2(\vec{z};\vec{R}^*) \;=\; \frac{1}{2}\frac{1}{\sigma_{d'}^2(\vec{z};\vec{R}^*)} \;=\; -\eta_{d'}^{(2)}(\vec{z};\vec{R}^*) \tag{72}$$

By inserting (69) and (72) into (60) we obtain:

$$\begin{aligned}
0 \;&=\; \frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{F}_2(\Phi,\vec{R}^*+\epsilon\,\tilde{\vec{R}})\big|_{\epsilon=0}\\
&=\; \frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\;\Big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_d^{(1)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\Big)\,\tilde{\vec{R}}\;\Big(T^{(1)}\big(x_d^{(n)}\big)-\frac{\mathrm{d}}{\mathrm{d}\eta^{(1)}}\vec{A}(\vec{\eta})\big|_{\vec{\eta}=\vec{\eta}_d(\vec{z};\vec{R}^*)}\Big)\,\mathrm{d}\vec{z}\\
&\quad+\frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\;\Big(\frac{\mathrm{d}}{\mathrm{d}\vec{R}}\eta_d^{(2)}(\vec{z};\vec{R})\big|_{\vec{R}=\vec{R}^*}\Big)\,\tilde{\vec{R}}\;\Big(T^{(2)}\big(x_d^{(n)}\big)-\frac{\mathrm{d}}{\mathrm{d}\eta^{(2)}}\vec{A}(\vec{\eta})\big|_{\vec{\eta}=\vec{\eta}_d(\vec{z};\vec{R}^*)}\Big)\,\mathrm{d}\vec{z} \qquad (73)\\
&=\; \frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\;\big(-\eta_d^{(1)}(\vec{z};\vec{R}^*)\big)\;\Big(T^{(1)}\big(x_d^{(n)}\big)-\frac{\mathrm{d}}{\mathrm{d}\eta^{(1)}}\vec{A}(\vec{\eta})\big|_{\vec{\eta}=\vec{\eta}_d(\vec{z};\vec{R}^*)}\Big)\,\mathrm{d}\vec{z}\\
&\quad+\frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\;\big(-\eta_d^{(2)}(\vec{z};\vec{R}^*)\big)\;\Big(T^{(2)}\big(x_d^{(n)}\big)-\frac{\mathrm{d}}{\mathrm{d}\eta^{(2)}}\vec{A}(\vec{\eta})\big|_{\vec{\eta}=\vec{\eta}_d(\vec{z};\vec{R}^*)}\Big)\,\mathrm{d}\vec{z} \qquad (74)
\end{aligned}$$

We have already shown above (66) that the first term with $\eta_{d'}^{(1)}(\vec{z};\vec{R}^*)$ is zero, which means that we can conclude:

$$0 \;=\; \frac{1}{N}\sum_{n,d}\int q_\Phi^{(n)}(\vec{z})\;\eta_d^{(2)}(\vec{z};\vec{R}^*)\;\Big(T^{(2)}\big(x_d^{(n)}\big)-\frac{\mathrm{d}}{\mathrm{d}\eta^{(2)}}\vec{A}(\vec{\eta})\big|_{\vec{\eta}=\vec{\eta}_d(\vec{z};\vec{R}^*)}\Big)\,\mathrm{d}\vec{z}. \tag{75}$$

Taking the intermediate results (66) and (75) together, we have shown that both summands that make up (59) are zero, i.e., expression (59) is zero at stationary point $\vec{R}^*$.

Therefore, we can finally go back to (51) and obtain at stationary point Θ^* (we go back again to full parameters):

$$\begin{aligned}
\mathcal{F}_2(\Phi,\Theta^*) &= \frac{1}{N}\sum_{n=1}^N\sum_{d=1}^D\int q_\Phi^{(n)}(\vec{z})\Big(\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\Theta^*\big)\;\vec{T}\big(x_d^{(n)}\big)-\;\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\Theta^*\big)\,\vec{A}'\big(\vec{\eta}_d\big(\vec{z};\Theta^*\big)\big)-\mathcal{H}[p_{\Theta^*}(x_d\,|\,\vec{z})]\Big)\mathrm{d}\vec{z}\\
&= \frac{1}{N}\sum_{n=1}^N\sum_{d=1}^D\int q_\Phi^{(n)}(\vec{z})\,\vec{\eta}_d^{\,\mathrm{T}}\big(\vec{z};\Theta^*\big)\,\Big(\vec{T}\big(x_d^{(n)}\big)-\vec{A}'\big(\vec{\eta}_d\big(\vec{z};\Theta^*\big)\Big)\mathrm{d}\vec{z}-\frac{1}{N}\sum_{n=1}^N\sum_{d=1}^D\int q_\Phi^{(n)}(\vec{z})\,\mathcal{H}[p_{\Theta^*}(x_d\,|\,\vec{z})]\mathrm{d}\vec{z}\\
&= -\frac{1}{N}\sum_{n=1}^N\int q_\Phi^{(n)}(\vec{z})\,\mathcal{H}[p_{\Theta^*}(\vec{x}\,|\,\vec{z})]\mathrm{d}\vec{z} \;=\; -\frac{1}{N}\sum_{n=1}^N\mathrm{E}_{q_\Phi^{(n)}}\big\{\mathcal{H}[p_{\Theta^*}(\vec{x}\,|\,\vec{z})]\big\}.
\end{aligned} \tag{76}$$

With (76) we have thus arrived at the generalization of the term $\mathcal{F}_2(\Phi,\Theta)$ for $\vec{z}$-dependent decoder variances. The derivation for the terms $\mathcal{F}_1(\Phi,\Theta)$ and $\mathcal{F}_3(\Phi)$ follow along the same lines as for the proofs of Props. 1 and 2 with no changes to the final results for $\mathcal{F}_1(\Phi,\Theta)$ and $\mathcal{F}_3(\Phi)$, which proofs the claim of Prop. 3. □

Appendix C: Details of Numerical Experiments

We here provide further details on the numerical experiments of Sec. 3.

C.1 Generation of Artificial Data Sets and Learning Visualizations

We generated the PCA data set according to the following generative model of probabilistic PCA:

$$p(\vec{z}) = \mathcal{N}(\vec{z}; \vec{0}, \mathbb{1}) \tag{77}$$

$$p(\vec{x} \,|\, \vec{z}) = \mathcal{N}(\vec{x}; W_{\text{gen.}}\vec{z} + \vec{\mu}_{\text{gen.}}; \sigma^2_{\text{gen.}}\mathbb{1}). \tag{78}$$

The generative parameters $W_{\text{gen.}}$ and $\vec{\mu}_{\text{gen.}}$ were drawn randomly from uniform distributions between 0 and 1 in each dimension for each new run. $\sigma_{\text{gen.}}$ was always set to 0.1. We used $H = 2$ as latent dimension and $D = 10$ as output dimension and generated 10.000 new training and testing data points for each experiment.

For the PCA-ring data sets, we introduced an additional non-linear transformation $\vec{x}' = g(\vec{x})$ with

$$p(\vec{z}) = \mathcal{N}(\vec{z}; \vec{0}, \mathbb{1}) \tag{79}$$

$$p(\vec{x} \,|\, \vec{z}) = \mathcal{N}(\vec{x}; W_{\text{gen.}}\vec{z}; \sigma^2_{\text{gen.}}\mathbb{1}) \tag{80}$$

$$\vec{x}' = \vec{\mu}_{\text{gen.}} + \frac{\vec{x}}{10} + \frac{\vec{x}}{|\vec{x}|} \tag{81}$$

projecting the data onto a ring-like structure in D-dimensional space (see, e.g., Doersch, 2016).

Fig. 3(c), on the last page, shows a PCA projection of the training data set, visualizations of the z- and x-space during training of VAE–3 as well as plots of the lower bound and the three entropies. Figs. 3(a) and (b) show the same plots for the linear VAE and VAE–1 on the PCA and MNIST data set, respectively. As expected, we see a slight overfitting of the linear VAE to the PCA training data, with the training log-likelihood converging to a bit above of the ground-truth. However, even with early stopping at around 700 iterations (see third segment of Fig. 3a), we see that the three entropies already compute the lower bound very well.

Videos showing the whole training process are available as ancillary files.

C.2 Noise in Three Entropies and Lower Bound

The optimization of VAEs is stochastic due to finite learning rates, finite batch sizes and approximations of integrals over $\vec{z}$ using sampling. A stationary point is consequently never fully converged to. Instead the parameters will finally stochastically fluctuate around a stationary point. In Fig. 1(c) we have (for VAE–3) numerically quantified the relative difference between the variational lower bound and Eqn. (39) of Prop. 3. As can be observed, and as stated in the main text, the values for the standard variational bound (9) and Eqn. (39) match very well in the region close to the stationary point (i.e. the region to which stochastic learning finally converges to). By evaluating over 100 runs, we observed an average *absolute* error of below 0.5% (i.e., 0.005) between the variational bound and Eqn. (39).

We can also further study the effect of stochasticity by using smaller batch sizes (Fig. 2b) or higher learning rates (Fig. 2c). In both cases, we obtain a higher stochasticity of the final fluctuations around the stationary point. As a consequence, the average absolute error becomes larger (Fig. 2, middle row) but is still smaller-equal than 1% (i.e., 0.01). The change of the error with changing stochasticity is better observed for the absolute relative error than for the relative deviation (i.e., if we define the relative deviation as the relative absolute error but without taking magnitudes in the numerator, see Fig. 2, bottom row). However, for completeness, we also provide the relative deviation which is (as expected) smaller than the error (Fig. 2, bottom row). This means that the difference between variational lower bound and Eqn. (39) can be positive as well as negative close to the stationary point. Averaging cancels out the differences in large parts, which is the reason for the relative deviation being significantly smaller than the relative absolute error.

In summary, the numerical experiments provide (A) consistency with the theoretical result of Prop. 3 (and the other propositions), and (B) they show that the three entropies results of Props. 2 and 3 can provide very accurate estimations of the variational bound in practice. Note in this respect that the computation of final values of lower bounds always provides valuable information. Final lower bounds can be used as approximations

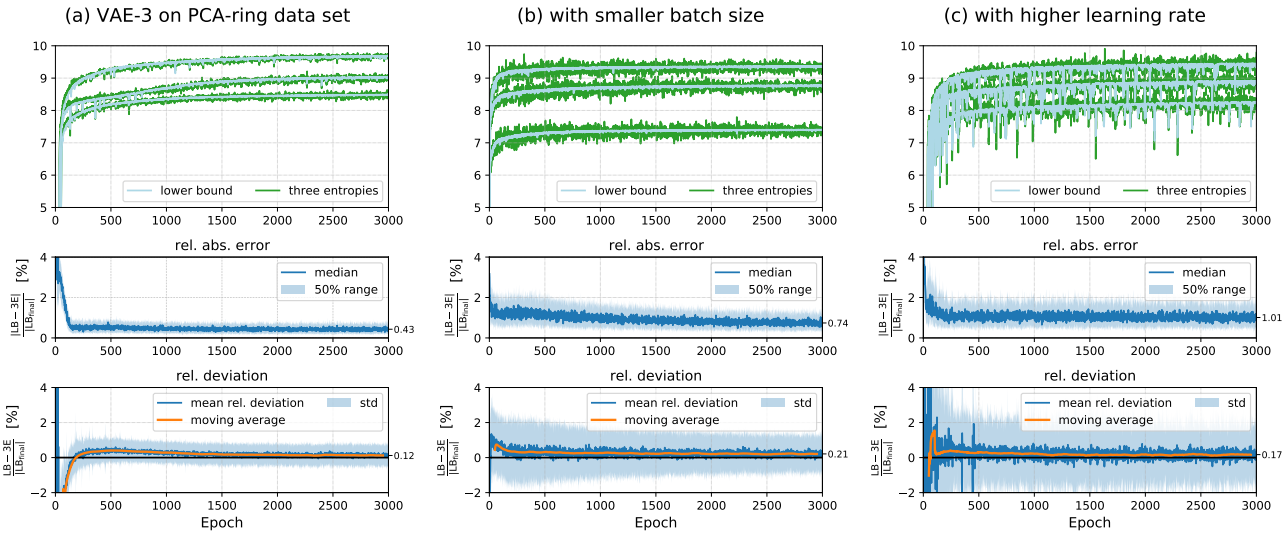


Figure 2: VAE–3 on PCA-ring data sets. The top two plots show experiments in the same way as Fig. 1: The top plot shows three independent runs on new randomly generated PCA-ring data each, while the middle plot shows the median of the relative absolute error between lower bound and three entropies over 100 such independent runs. Additionally we show the mean relative deviation (i.e., without taking absolute values) between lower bound an three entropies of these runs as bottom plots, together with the moving averages over 100 epochs. (a) shows experiment in the same setting as Fig. 1(c) over longer training time. (b) shows the same experiments with a batch size of 100 (compared to a batch size of 2000 in the other plots). (c) shows the same experiments with a learning rate of 0.005 (compared to a learning rate of 0.001 in the other plots).

to the true log-likelihood in many settings, and could be used for comparisons between different runs and/or different VAEs. Typical such comparisons can be applied for model selection, for instance. But also when the lower bounds are not good approximations of the log-likelihood, knowing their values at convergence is very useful to analyze learning: runs with high final values for variational lower bounds but low values for (held-out) log-likelihood indicate overfitting, for instance. The experiments of Figs. 1 and 2 show that the values of variational lower bounds can in practice easily be estimated with high accuracy. In the case of VAEs in the form of VAE–1, estimation with high accuracy is even possible using a closed-form expressions (see Prop. 2).

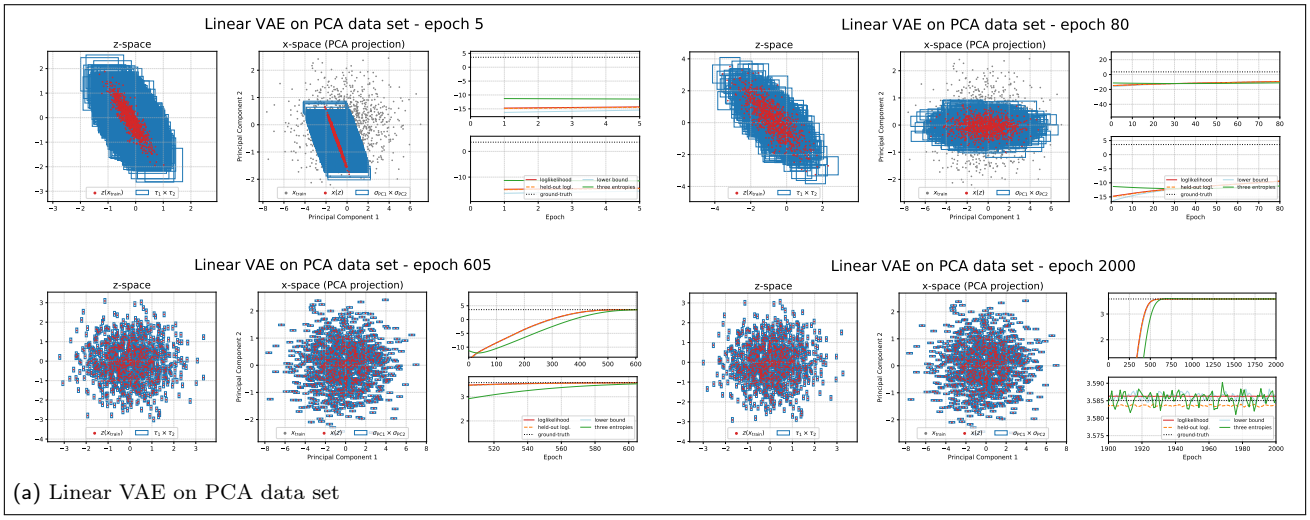
C.3 Implementation and Training Details

We used for all experiments a batch size of 2000, learning rates of $\epsilon = 10^{-3}$ and 100 samples (with the exception of Figs. 2b and 2c, as stated). These values were roughly chosen to give results with little fluctuations in the sampled lower bound and three entropies within reasonable time. The encoder and decoder DNNs for VAE–1 and VAE–3 were built with two hidden layers of 50 hidden units each and ReLU activations. Listing 1 shows an example `PyTorch` implementation of the linear VAE used in Fig. 1(a) as well as functions to compute the accumulated three entropies (i.e., summed over mini-batches) for the linear VAE, as well as VAE–1 and VAE–3.

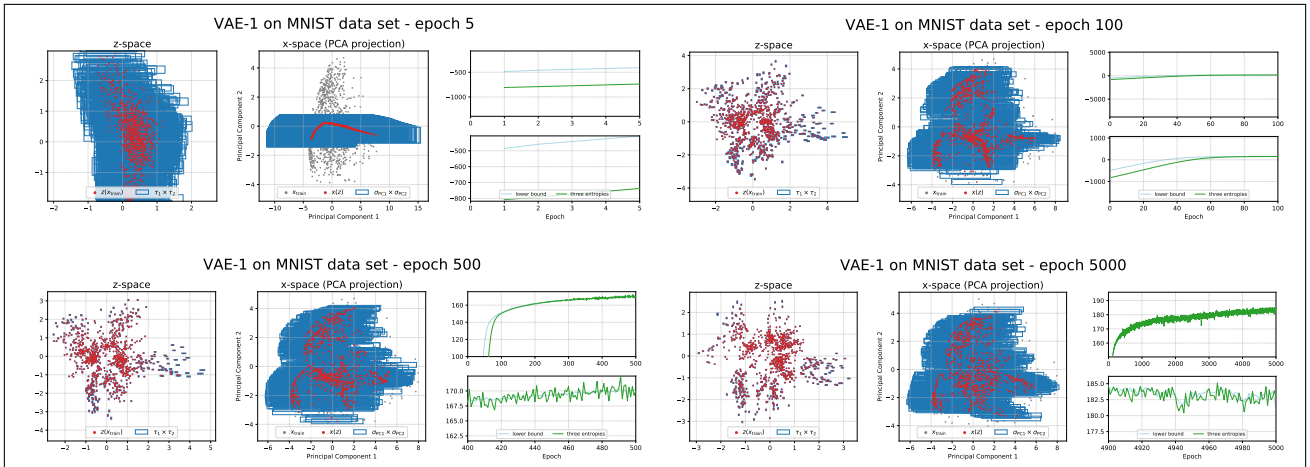
Listing 1: Example Implementations

```
import torch
from torch.nn import Linear, Module, Parameter
from torch.nn.functional import softplus
from torch.distributions import Normal, kl_divergence
import math
pi = torch.Tensor([math.pi])

class LinearVAE(Module):
    def __init__(self, H, D, num_samples=100):
        super(LinearVAE, self).__init__()
        self.H = H
        self.D = D
        self.num_samples = num_samples

        self.encoder = Linear(D, H)
```

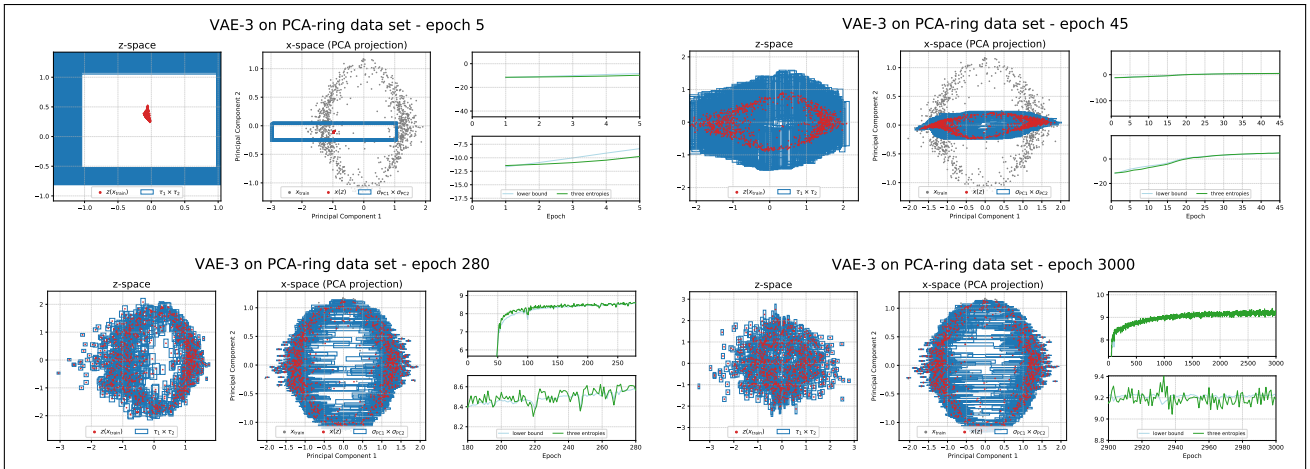
```
        self.decoder = Linear(H, D)
        self.noise_std = Parameter(torch.Tensor([0.]))
        self.q_z_std = Parameter(torch.zeros(H))
        self.p_z = Normal(torch.Tensor([0.]), torch.Tensor([1.]))

    def forward(self, x):
        sigma = softplus(self.noise_std)
        tau = softplus(self.q_z_std)
        N, D = x.shape

        z_params = self.encoder(x)
        q_z = Normal(z_params, tau)

        z_samples = q_z.rsample([self.num_samples])
        p_x_z_mean = self.decoder(z_samples)
        p_x_z = Normal(p_x_z_mean, sigma)

        lower_bound = p_x_z.log_prob(x).mean(0).sum() \
                      - kl_divergence(q_z, self.p_z).sum()
        three_entropies_linear = self.three_entropies_linear(N, D, sigma, tau)
        return lower_bound, three_entropies_linear

    @staticmethod
    def three_entropies_linear(N, D, sigma, tau):
        """Three Entropies (accumulated) for linear VAE, Eq.(33).
        Args:
            N (int) : batch size
            D (int) : output dimensionality
            sigma (torch.tensor, size=(1,)) : decoder standard deviation
            tau (torch.tensor, size=(H,)    : encoder standard deviation
        """
        return N*(-D/2*(torch.log(2*pi)+1)-D*torch.log(sigma)+torch.log(tau).sum())

class VAE1(Module):

    [...]

    @staticmethod
    def three_entropies_nonlinear(N, D, sigma, tau):
        """Three Entropies (accumulated) for non-linear VAE (VAE-1), Eq.(25).
        Args:
            N (int) : batch size
            D (int) : output dimensionality
            sigma (torch.tensor, size=(1,)) : decoder standard deviation
            tau (torch.tensor, size=(N, H)) : encoder standard deviation
        """
        return N*D*(-1/2*(torch.log(2*pi)+1)-torch.log(sigma)) + torch.log(tau).sum()

class VAE3(Module):

    [...]

    @staticmethod
    def three_entropies_sigmaz(N, D, sigma, tau):
        """Three Entropies (accumulated) for sigma(z)-VAE (VAE-3), Eq.(39).
        Args:
            N (int) : batch size
            D (int) : output dimensionality
            sigma (torch.tensor, size=(num_samples, N, D)) : decoder std
            tau (torch.tensor, size=(N, H)) : encoder standard deviation
        """
        return -N*D/2*(torch.log(2*pi)+1) - torch.log(sigma).mean(0).sum() \
               + torch.log(tau).sum()
```



(a) Linear VAE on PCA data set



(b) VAE–1 on MNIST data set



(c) VAE–3 on PCA-ring data set

Figure 3: The training state of the VAEs of Fig. 1 is shown at four different points during training for each VAE on their respective data sets. The first plot of each segment shows 1000 $\vec{z}$-samples from the encoder in the 2-dimensional z-space with the encoder standard deviation τ_h displayed as rectangle of width $2\tau_1$ and height $2\tau_2$. The second plot shows projections of the training data and of the 1000 $\vec{x}(\vec{z})$-reconstructions of the decoder to the first two PCA components of the training data, as well as the PCA projections of the decoder standard deviation as rectangle of width $2\sigma_{\mathrm{PC1}}$ and height $2\sigma_{\mathrm{PC2}}$. The last two plots show the sampled lower bound and the three entropies per data point, with the lower plots showing zoomed-in versions of the upper plots. For (a) the training, held-out and ground-truth log-likelihoods are shown additionally.