# The ELBO of Variational Autoencoders Converges to a Sum of Entropies

**Simon Damm**[*]
Ruhr-University Bochum, Germany

**Dennis Forster**[†]
Frankfurt University of Applied Sciences, Germany

**Dmytro Velychko**
University of Oldenburg, Germany

**Zhenwen Dai**
Spotify, London, UK

**Asja Fischer**
Ruhr-University Bochum, Germany

**Jörg Lücke**[*]
University of Oldenburg, Germany

## Abstract

The central objective function of a variational autoencoder (VAE) is its variational lower bound (the ELBO). Here we show that for standard (i.e., Gaussian) VAEs the ELBO converges to a value given by the sum of three entropies: the (negative) entropy of the prior distribution, the expected (negative) entropy of the observable distribution, and the average entropy of the variational distributions (the latter is already part of the ELBO). Our derived analytical results are exact and apply for small as well as for intricate deep networks for encoder and decoder. Furthermore, they apply for finitely and infinitely many data points and at any stationary point (including local maxima and saddle points). The result implies that the ELBO can for standard VAEs often be computed in closed-form at stationary points while the original ELBO requires numerical approximations of integrals. As a main contribution, we provide the proof that the ELBO for VAEs is at stationary points equal to entropy sums. Numerical experiments then show that the obtained analytical results are sufficiently precise also in those vicinities of stationary points that are reached in practice. Furthermore, we discuss how the novel entropy form of the ELBO can be used to analyze and understand learning behavior. More generally, we believe that our contributions can be useful for future theoretical and practical studies on VAE learning as they provide novel information on those points in parameters space that optimization of VAEs converges to.

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

## 1 INTRODUCTION

Variational autoencoders (VAEs; Kingma and Welling, 2014; Rezende et al., 2014) have emerged as a popular choice for probabilistic generative modeling, a sub-field of unsupervised (deep) learning. VAEs are in their most common form defined as latent variable models that learn representations of the data $\mathbf{x} \sim p(\mathbf{x})$, with $\mathbf{x} \in \mathbb{R}^D$, in a usually low(er) dimensional latent space $\mathbf{z} \in \mathbb{R}^H$. In contrast to conventional autoencoders, the mappings between data and latent space are stochastic. The (by far) most common choice for prior distribution $p(\mathbf{z})$, encoder $q_\Phi(\mathbf{z}|\mathbf{x})$ and decoder $p_\Theta(\mathbf{x}|\mathbf{z})$ are Gaussian distributions, where deep neural networks (DNNs) are used to define Gaussian means and (optionally) covariances. Ideally, maximum likelihood estimation of model parameters $\Theta$ would be deployed with

$$p_\Theta(\mathbf{x}) = \int p(\mathbf{z}) p_\Theta(\mathbf{x}|\mathbf{z}) \mathrm{d}\mathbf{z} \tag{1}$$

approximating the target data distribution $p(\mathbf{x})$. However, since the likelihood in Eqn. (1) is not tractable for complex decoder distributions, the *evidence lower bound* (ELBO) is optimized instead. The ELBO is also known as *variational lower bound* or *variational free energy* (Neal and Hinton, 1998).

Given a set of $N$ data points $\mathbf{x}^{(n)}$, $n \in \{1, \ldots, N\}$, the ELBO in dependence of encoder and decoder parameters (i.e., $\Phi$ and $\Theta$, respectively) is given by

$$\mathcal{F}(\Phi,\Theta) = \overbrace{\frac{1}{N}\sum_n \int q_\Phi(\mathbf{z}\,|\,\mathbf{x}^{(n)}) \log\big(p_\Theta(\mathbf{x}^{(n)}\,|\,\mathbf{z})\big)\,\mathrm{d}\mathbf{z}}^{S_{\mathrm{rec}}(\Phi,\Theta)} \underbrace{-\frac{1}{N}\sum_n D_{\mathrm{KL}}\big(q_\Phi(\mathbf{z}\,|\,\mathbf{x}^{(n)})\,\|\,p(\mathbf{z})\big)}_{S_{\mathrm{reg}}(\Phi,\Theta)}\ , \tag{2}$$

[*]joint main contributions
[†]Large parts of the research were conducted while the author was affiliated with the University of Oldenburg, Germany.

where $D_{\mathrm{KL}}\big(q(\mathbf{z}) \,\|\, p(\mathbf{z})\big)$ denotes the Kullback-Leibler divergence between two distributions $q$ and $p$. The first ELBO term, $S_{\mathrm{rec}}(\Phi, \Theta)$, is usually referred to as reconstruction score and the second $S_{\mathrm{reg}}(\Phi, \Theta)$ as regularization score (Kingma et al., 2019). For all encoder parameters $\Phi$ and decoder parameters $\Theta$ the ELBO is smaller or equal to the log-likelihood $\mathcal{L}(\Theta) = \frac{1}{N}\sum_n \log\big(p_\Theta(\mathbf{x}^{(n)})\big)$.

While the ELBO has proven to be an exceptionally successful learning objective, it is, like the log-likelihood itself, usually not analytically tractable for VAEs (and neither for many other models). The intractability of the bound for VAEs stems from $S_{\mathrm{rec}}(\Phi, \Theta)$ in Eqn. (2): Because of (potentially very intricate) DNN non-linearities of standard VAEs, the integrals cannot be solved analytically. A central research challenge for training VAEs is therefore the development of efficient methods to approximate intractable integrals of the ELBO. Indeed, the suggestion of efficient methods to estimate gradients of Eqn. (2) using sampling and reparametrization (Kingma and Welling, 2014; Rezende et al., 2014) has played the key role in the establishment of the field of VAE research.

Because of the potentially complex DNNs deployed in 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 or saddle points. Therefore maybe unexpectedly, we here provide an exact analytical result for all standard VAEs. Concretely, we show that at convergence the ELBO is given by a sum of the entropies of those distributions defining a VAE. In order to make our contribution more precise let us properly define a vanilla Gaussian VAE, which we here term **VAE–1**.

**Definition 1** (VAE–1; VAE with component-wise equivalent decoder variances)**.** *Consider a VAE with standard normal prior*

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbb{I}) \ ,$$

*and Gaussian encoder and decoder given by*

$$\begin{aligned} q_\Phi(\mathbf{z}|\mathbf{x}) &= \mathcal{N}\big(\mathbf{z}; \boldsymbol{\nu}_\Phi(\mathbf{x}), \mathcal{T}_\Phi(\mathbf{x})\big) \ , \\ p_\Theta(\mathbf{x}|\mathbf{z}) &= \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\Theta(\mathbf{z}), \sigma^2\mathbb{I}) \ . \end{aligned}$$

*Let the encoder covariance $\mathcal{T}_\Phi(\mathbf{x})$ be diagonal and let encoder mean $\boldsymbol{\nu}_\Phi(\mathbf{x})$, covariance $\mathcal{T}_\Phi(\mathbf{x})$, and decoder mean $\boldsymbol{\mu}_\Theta(\mathbf{z})$ be parameterized by DNNs, i.e.,*

$$\begin{aligned} \boldsymbol{\nu}_\Phi(\mathbf{x}) &= \mathrm{DNN}_\nu(\mathbf{x}; V) \ , \\ \mathcal{T}_\Phi(\mathbf{x}) &= \mathrm{diag}\left(\tau_1^2(\mathbf{x}), \ldots, \tau_H^2(\mathbf{x})\right) \ , \\ \boldsymbol{\mu}_\Theta(\mathbf{z}) &= \mathrm{DNN}_\mu(\mathbf{z}; W) \end{aligned} \tag{3}$$

*with $\big(\tau_1^2(\mathbf{x}), \ldots, \tau_H^2(\mathbf{x})\big)^{\mathrm{T}} = \mathrm{DNN}_\tau(\mathbf{x}; T)$. We term a VAE of this form VAE–1. By $\Theta = (W, \sigma^2)$ we denote the set of all parameters of the decoder, and by $\Phi = (V, T)$ all parameters of the encoder. We hereby assume that the respective parameter sets $W, V,$ and $T$ include all weight matrices and biases of the DNNs.*

The results we will derive are exploiting properties of the ELBO (Eqn. (2)) at stationary points, i.e., of those points in parameter space where the ELBO reaches an extremum (local or global optima) or a saddle point. As a consequence, the presented results are applicable, e.g., for gradient-based optimization techniques. Throughout the paper, we refer with ‘at convergence’ to the stationary points but we remark that VAE parameters will in practice only reach the vicinity of these points (due to finite learning rates and stochasticity; we elaborate on this in Appendix D).

For VAE–1 as defined above, the final result is sufficiently concise to be stated here initially (before we discuss its derivation, related work, and more general VAEs later on): At all stationary points of the ELBO (Eqn. (2)) we have

$$\begin{aligned} \mathcal{F}(\Phi, \Theta) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{H}[q_\Phi(\mathbf{z}|\mathbf{x}^{(n)})] \\ - \mathcal{H}[p(\mathbf{z})] - \mathcal{H}[p_\Theta(\mathbf{x} \,|\, \mathbf{z})] \end{aligned} \tag{4}$$

where $\mathcal{H}[p(\cdot)]$ denotes the entropy of a distribution $p(\cdot)$. That is, the ELBO is, at convergence, given by the entropies of encoder, prior, and decoder distribution. Using the closed-form expressions for Gaussian entropies, the ELBO is consequently closed-form and solely depends on the variance parameters of encoder and decoder:

$$\begin{aligned} \mathcal{F}(\Phi, \Theta) = \frac{1}{2N}\sum_{n=1}^{N}\sum_{h=1}^{H} \log\big(2\pi e \tau_h^2(\mathbf{x}^{(n)}; \Phi)\big) \\ - \frac{H}{2}\log(2\pi e) - \frac{D}{2}\log(2\pi e \sigma^2) \ . \end{aligned} \tag{5}$$

No other parameters are required to compute the bound, and in particular no knowledge about (or passes through) the decoder DNN is needed. Considering Eqn. (5), we also remark that the expression does not contain any approximations of any integrals. Using just the data and the learned parameters, the ELBO can at convergence be computed exactly. We stress that the closed-form expression (Eqn. (5)) does *not* replace the original bound as a learning objective (i.e., it can not be used analogously to Eqn. (2) for parameter optimization). Importantly, however, Eqn. (4) (and later discussed generalizations) implies that during learning the ELBO converges to a sum of three entropies.

## 2 RELATED WORK

In order to understand and improve learning algorithms, points in parameter space representing (potentially locally) optimal solutions are of high interest. For VAEs there are several lines of work in this respect which investigate ELBO optimization (Hoffman and Johnson, 2016; Mescheder et al., 2017; Dai et al., 2018; Lucas et al., 2019; Shekhovtsov et al., 2022). Work by Dai et al. (2018) and Lucas et al. (2019), for instance, highlight the connections of linear (Gaussian) VAEs to (robust) principal component analysis (PCA; Tipping and Bishop, 1999; Xu et al., 2010; Candès et al., 2011; Holtzman et al., 2020) and the crucial role of covariances of encoder and decoder, respectively, that help to circumvent undesirable maxima in the optimization landscape of VAEs. Dai et al. (2018) derived meaningful insights on optima of the ELBO based on tractable special cases. In addition, Lucas et al. (2019) link spurious local maxima to posterior collapse and derive a measure to quantify this phenomenon. Shekhovtsov et al. (2022) study the approximation gap between likelihood and ELBO for exponential family VAEs, and demonstrate that ELBO-based learning is subject to an inductive bias towards the (rather restricted) consistent set.

The here reported results, in contrast, closely relate the ELBO objective of VAEs to entropies. The relation between the integrals of the ELBO and entropies has been of interest previously (Lücke and Henniges, 2012). However, the main results of that previous work have used (A) the limit case of infinitely many data points, (B) assumed a perfect match of variational and full posterior distributions, and (C) required convergence to global optima. The work also discussed relaxations of these relatively unrealistic assumptions (Lücke and Henniges, 2012, Sec. 6). But those relaxations made use of properties of sparse coding like models trained using expectation maximization. While being related, the previous results are, therefore, not applicable to the training of VAEs. Furthermore, work in parallel to this study considers elementary generative models with distributions in the exponential family (Lücke, 2022) but no deep models such as VAEs are treated. We will later make use of that work for more complex VAEs in which DNNs also parameterize decoder variances. Results of (Lücke, 2022) in principle also allow treatments of still less conventional VAEs, e.g., VAEs defined using distributions such as Gamma, Bernoulli, continuous Bernoulli, Beta or Categorical distributions. However, for each distribution or combination of distributions, a parameterization condition has to be verified analytically. Appendix B shows the verification for the most complex Gaussian VAEs treated here. In this context the Appendix also provides a brief discussion of the conditions for non-Gaussian distributions. While the result of Eqn. (5) is generalizable, VAEs with fixed decoder variances are straight-forward examples for VAEs not converging to entropy sums. But we remark that learning $\sigma^2$ is usually beneficial (Lucas et al., 2019; Rybkin et al., 2021).

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 contributions by Kingma and Dhariwal (2018) or Oord et al. (2016). In contrast, we here investigate the standard (in general intractable) learning objective of VAEs and show its analytical tractability at convergence.

## 3 THE ELBO AT CONVERGENCE

For the derivation of our main results we require a more explicit, yet standard form of the decoder DNN:

**Assumption 1.** *The network $\mu_\Theta(\mathbf{z}) = \mathrm{DNN}_\mu(\mathbf{z}, W)$ is a composition of linear mappings followed by point-wise non-linear functions. Concretely, $\boldsymbol{\mu}_\Theta(\mathbf{z})$ is given by*

$$W^L \mathcal{S}\big(W^{L-1}\mathcal{S}(\cdots\mathcal{S}(W^0\mathbf{z}+\mathbf{b}^0)\cdots) + \mathbf{b}^{L-1}\big) + \mathbf{b}^L,$$

*where $W^l$ denotes the weight matrix of layer $l$ and $\mathbf{b}^l$ denotes its bias terms. The point-wise non-linearities $\mathcal{S}(\cdot)$ can (but do not have to) differ from layer to layer.*

Throughout this paper we presume Assumption 1 to be fulfilled. We do not assume any specific architecture for the encoder networks $\mathrm{DNN}_\nu$ and $\mathrm{DNN}_\tau$.

### 3.1 A Reparameterized VAE

Before we derive the result for VAE–1 presented in Eqn. (4), let us consider a different VAE which can be regarded as a reparametrization of VAE–1 and which we refer to as **VAE–2**.

**Definition 2** (VAE–2, VAE with component-wise equivalent decoder variances *and* learnable prior covariance)**.** *Consider a VAE with a parameterized prior*

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

*and Gaussian encoder and decoder given by*

$$q_\Phi(\tilde{\mathbf{z}}|\mathbf{x}) = \mathcal{N}\big(\tilde{\mathbf{z}}; \tilde{\boldsymbol{\nu}}_\Phi(\mathbf{x}), \tilde{\mathcal{T}}_\Phi(\mathbf{x})\big)\ ,$$
$$p_\Theta(\mathbf{x}\,|\,\tilde{\mathbf{z}}) = \mathcal{N}(\mathbf{x}; \tilde{\boldsymbol{\mu}}_\Theta(\tilde{\mathbf{z}}), \sigma^2\mathbb{I})$$

*where $\tilde{\boldsymbol{\nu}}_\Phi(\mathbf{x})$ and $\tilde{\mathcal{T}}_\Phi(\mathbf{x})$ are parametrized analogously to VAE–1 (Def. 1). In contrast to VAE–1, the prior's covariance matrix is now given by the diagonal matrix $A = \mathrm{diag}\left(\alpha_1^2, \ldots, \alpha_H^2\right)$. Furthermore, we assume for the decoder DNN, $\tilde{\boldsymbol{\mu}}_\Theta(\tilde{\mathbf{z}})$, that the columns of the weight matrix $\tilde{W}^0$ are of unit length, i.e, for $\tilde{W}^0 = (\tilde{\mathbf{W}}_1^0, \ldots, \tilde{\mathbf{W}}_H^0)$ we demand*

$$\forall h \in \{1,\ldots,H\} : \big(\tilde{\mathbf{W}}_h^0\big)^{\mathrm{T}}\tilde{\mathbf{W}}_h^0 = 1\,. \qquad (6)$$

*We refer to such a VAE as VAE–2.*

We will later see that VAE–2 can indeed 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 its prior distribution can be learned (which will be exploited below).

The stationary points are those points in parameter space for which the derivatives w.r.t. all parameters (parameters $\Phi$ and $\Theta$) vanish. In particular, this implies for VAE–2:

$$\frac{\mathrm{d}}{\mathrm{d}\sigma^2}\mathcal{F}(\Phi,\Theta) = 0 \quad \text{and} \quad \frac{\mathrm{d}}{\mathrm{d}\alpha_h^2}\mathcal{F}(\Phi,\Theta) = 0\ \forall h\ . \qquad (7)$$

The derivatives w.r.t. the DNN parameters $W, V$, and $T$ are also zero at stationary points (for the parameters $\tilde{W}^0$ a derivative with Lagrange multipliers is zero). However, we will only use Eqn. (7) in the sequel. We can now proceed to proof the following theorem:

**Theorem 1.** *Given a VAE–2 as in Def. 2 that satisfies Assumption 1. At all stationary points the ELBO $\mathcal{F}(\Phi,\Theta)$ of VAE–2 is then equal to*

$$\begin{aligned}&\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[q_\Phi(\tilde{\mathbf{z}}|\mathbf{x}^{(n)})]-\mathcal{H}[p_\Theta(\tilde{\mathbf{z}})]-\mathcal{H}[p_\Theta(\mathbf{x}\,|\,\tilde{\mathbf{z}})]\\ &=\frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}\log\big(\det(2\pi e\tilde{\mathcal{T}}_\Phi(\mathbf{x}^{(n)}))\big)\\ &\quad-\frac{1}{2}\log\big(\det(2\pi eA)\big)-\frac{D}{2}\log\big(2\pi e\sigma^2\big).\end{aligned}\tag{8}$$

*Proof.* We can rewrite the standard formulation of the ELBO (Eqn. (2)) to consist of three terms:

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

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

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

$$\begin{aligned}&\frac{1}{N}\sum_n\Bigg(-\frac{1}{2\sigma^2}\int q_\Phi^{(n)}(\mathbf{z})\,\|\mathbf{x}^{(n)}-\boldsymbol{\mu}_\Theta(\mathbf{z})\|^2\mathrm{d}\mathbf{z}\\ &\qquad-\frac{D}{2}\log\big(2\pi\sigma^2\big)-\frac{D}{2}+\frac{D}{2}\Bigg)\\ &=\frac{D}{2}\Bigg(1-\frac{1}{ND\sigma^2}\sum_n\int q_\Phi^{(n)}(\mathbf{z})\,\|\mathbf{x}^{(n)}-\boldsymbol{\mu}_\Theta(\mathbf{z})\|^2\mathrm{d}\mathbf{z}\Bigg)\\ &\qquad-\frac{D}{2}\log(2\pi e\sigma^2)\ \ ,\end{aligned}\tag{9}$$

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 $\sigma^2$, which implies $\frac{\mathrm{d}}{\mathrm{d}\sigma^2}\mathcal{F}_2(\Phi,\Theta)=0$. In virtue of Eqn. (9), the derivative has a specific structure given by:

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

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

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

We recognize the integral to be the first term of Eqn. (9), 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

$$\mathcal{F}_2(\Phi,\Theta)=-\frac{D}{2}\log\big(2\pi e\sigma^2\big)=-\mathcal{H}[p_\Theta(\mathbf{x}\,|\,\mathbf{z})]\ \ .\tag{10}$$

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

$$\begin{aligned}&\frac{1}{N}\sum_n\Big(-\sum_h\frac{1}{2\alpha_h^2}\int q_\Phi^{(n)}(\mathbf{z})z_h^2\mathrm{d}\mathbf{z}-\frac{1}{2}\sum_h\log\big(2\pi\alpha_h^2\big)\Big)=\\ &\frac{1}{2}\sum_h\Big(1-\frac{1}{N\alpha_h^2}\sum_n\int q_\Phi^{(n)}(\mathbf{z})z_h^2\mathrm{d}\mathbf{z}\Big)-\frac{1}{2}\sum_h\log\big(2\pi e\alpha_h^2\big)\ ,\end{aligned}\tag{11}$$

where the last term is the negative entropy of the prior. At stationary points Eqn. (7) applies, and we obtain:

$$\begin{aligned}0&=\frac{\mathrm{d}}{\mathrm{d}\alpha_h^2}\mathcal{F}_1(\Phi,\Theta)\\ &=\frac{1}{2}\sum_{h'}\frac{\delta_{hh'}}{N\alpha_h^4}\sum_n\int q_\Phi^{(n)}(\mathbf{z})z_{h'}^2\mathrm{d}\mathbf{z}-\frac{1}{2}\sum_{h'}\frac{\delta_{hh'}}{\alpha_h^2}\\ &=-\frac{1}{2\alpha_h^2}\Big(1-\frac{1}{N\alpha_h^2}\sum_n\int q_\Phi^{(n)}(\mathbf{z})z_h^2\mathrm{d}\mathbf{z}\Big)\ \ .\end{aligned}$$

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

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

The first sum over $h$ in Eqn. (11) is consequently zero, and we obtain at convergence:

$$\mathcal{F}_1(\Phi,\Theta)=-\frac{1}{2}\log\big(\det(2\pi eA)\big)=-\mathcal{H}[p_\Theta(\mathbf{z})].\tag{12}$$

The term $\mathcal{F}_3(\Phi)$ is directly given as the average entropy of the variational distribution. Taken together, we thus obtain the claim. □

## 3.2 Convergence to Sums of Entropies

We can now provide a result for the ELBO of the standard VAE given by Def. 1 (VAE–1) by translating the result of Theorem 1 for VAE–2 back to the original parameterization.

**Theorem 2.** *Given a VAE–1 as in Def. 1 that satisfies Assumption 1. Then at all stationary points the ELBO, $\mathcal{F}(\Phi,\Theta)$, of VAE–1 is equal to*

$$\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[q_\Phi(\mathbf{z}|\mathbf{x}^{(n)})]-\mathcal{H}[p_\Theta(\mathbf{z})]-\mathcal{H}[p_\Theta(\mathbf{x}\,|\,\mathbf{z})] \quad (13)$$

$$=\frac{1}{2N}\sum_{n=1}^{N}\sum_{h=1}^{H}\log\big(\tau_h^2(\mathbf{x}^{(n)};\Phi)\big)-\frac{D}{2}\log\big(2\pi e\sigma^2\big). \quad (14)$$

*Proof.* Let us start by showing that VAE–1 and VAE–2 indeed parametrize the same distributions. Following Assumption 1 the initial mapping of $\mathrm{DNN}_\mu$ is linear with weight matrix $W^0$ for VAE–1 or $\tilde{W}^0$ for VAE–2. With $\tilde{W}^0=\big(\tilde{\mathbf{W}}_1^0,\ldots,\tilde{\mathbf{W}}_H^0\big)$ and $A$ as in Def. 2 we can now set

$$W^0 = \tilde{W}^0A^{\frac{1}{2}}=\big(\alpha_1\tilde{\mathbf{W}}_1^0,\ldots,\alpha_H\tilde{\mathbf{W}}_H^0\big)\ . \quad (15)$$

Note that the column vectors of $\tilde{W}^0$ are constrained to unit length (see Def. 2). Thus, $\big(\tilde{W}^0A^{\frac{1}{2}}\big)$ parameterize the same space of matrices as $W^0$. The first linear operation of $\mathrm{DNN}_\mu$ now becomes:

$$W^0\mathbf{z}+\mathbf{b}^0=\sum_h\tilde{\mathbf{W}}_h^0\alpha_h z_h+\mathbf{b}^0\ . \quad (16)$$

Considering the term $\alpha_h z_h$, we can now generate $\tilde{z}_h\sim\mathcal{N}(\tilde{z}_h;0,\alpha_h^2)$ instead of $z_h\sim\mathcal{N}(z_h;0,1)$. We recognize that VAE–1 in this way takes on the form of VAE–2. 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.

We thus conclude that Theorem 1 applies. The decoder variance $\sigma^2$ remains unchanged and $A$ could be obtained from the column vectors of $W^0$. However, it is left to express $\tilde{\mathcal{T}}_\Phi(\mathbf{x})$ in terms of $\mathcal{T}_\Phi(\mathbf{x})$. Let us drop subscript and argument of $\tilde{\mathcal{T}}$ for readability. Then $\tilde{\mathcal{T}}$ is the covariance matrix of a Gaussian distribution defined in the space of $\tilde{\mathbf{z}}$. In virtue of Eqn. (16) the random variable $\mathbf{z}$ is given by $\mathbf{z}=A^{-\frac{1}{2}}\tilde{\mathbf{z}}$. Consequently, if $\tilde{\mathbf{z}}$ is Gaussian distributed with covariance $\tilde{\mathcal{T}}$, then $\mathbf{z}$ 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. (8) we observe the first term to cancel with part of the last term:

$$\begin{aligned}\mathcal{F}(\Phi,\Theta) &= -\frac{1}{2}\log\big(\det(2\pi eA)\big)-\frac{D}{2}\log\big(2\pi e\sigma^2\big)\\ &\quad+\frac{1}{N}\sum_n\frac{1}{2}\log\big(\det(2\pi eA\mathcal{T}_\Phi(\mathbf{x}^{(n)})\big)\\ &= -\frac{H}{2}\log(2\pi e)-\frac{D}{2}\log\big(2\pi e\sigma^2\big)\\ &\quad+\frac{1}{N}\sum_n\frac{1}{2}\log\big(\det(2\pi e\mathcal{T}_\Phi(\mathbf{x}^{(n)})\big)\\ &= -\frac{D}{2}\log\big(2\pi e\sigma^2\big)+\frac{1}{2N}\sum_n\log\big(\det(\mathcal{T}_\Phi(\mathbf{x}^{(n)})\big).\end{aligned} \quad (17)$$

The middle equation we recognize as the sum of three entropies in Eqn. (13), which proofs the claim. The last equation explicitly expresses the entropies (after further simplification) using the variance parameters of VAE–1. For Eqn. (13) we moved the last term in Eqn. (17) to the front to match the order of terms to the order of processing in VAEs. □

There are a number of implications and remarks if considering Theorem 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 straight-forward and efficient in practice. 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 value of the bound only depends on a subset of VAE parameters at convergence: the variances of encoder and decoder. In particular, the entropies (and therefore the ELBO value at convergence) do not depend on the DNNs for Gaussian means.

**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; Dai et al., 2018; Kunin et al., 2019; Lucas et al., 2019), i.e., a VAE with decoder and encoder given, respectively, by:

$$p_\Theta(\mathbf{z})=\mathcal{N}(\mathbf{z};\mathbf{0},\mathbb{I}),\ p_\Theta(\mathbf{x}\,|\,\mathbf{z})=\mathcal{N}(\mathbf{x};W\mathbf{z}+\boldsymbol{\mu}_0,\sigma^2\mathbb{I}), \quad (18)$$

$$q_\Phi^{(n)}(\mathbf{z})=\mathcal{N}(\mathbf{z};V(\mathbf{x}^{(n)}-\boldsymbol{\mu}_0),\mathcal{T}), \quad (19)$$

with weight matrices $W$ and $V$, and covariance $\mathcal{T}=\mathrm{diag}\big(\tau_1^2,\ldots,\tau_H^2\big)$. 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.* (18) *and* (19)*. At all stationary points of its ELBO (Eqn.* (2)*) it applies that:*

$$\mathcal{F}(\Phi,\Theta)=\frac{1}{2}\sum_h\log\big(\tau_h^2\big)\ -\ \frac{D}{2}\log\big(2\pi e\sigma^2\big). \quad (20)$$

*Proof.* For the proof of Theorem 2 we only required the reparametrization of the first linear mapping of the decoder DNN (cf. Assumption 1). Theorem 2 thus also applies for the linear VAE as a special case of VAE–1 (we elaborate in Appendix A). Inserting the matrix $\mathcal{T}$ of Eqn. (19) into Eqn. (14) proves the claim as $\mathcal{T}$ is independent of $n$. □

Corollary 1 further highlights that the variance parameters determine the bound at convergence. As it is known that linear VAEs can recover the exact maximum likelihood (Dai et al., 2018; Lucas et al., 2019), we can even conclude that the bound given in Eqn. (20) is tight at convergence. We use this result in Section 4 and elaborate in Appendix A.

## 3.3 More General Gaussian VAEs

VAE–1 represents the presumably most common form of VAEs. However, generalizations which use a DNN to learn more complex decoder covariances alongside a DNN for decoder means represent a possible generalization (Rezende et al., 2014; Dorta et al., 2018, etc). To also include VAEs with DNNs for *decoder* variances, we have to extend our analysis as we, so far, considered covariance $\sigma^2\mathbb{I}$. As relatively straight-forward generalization, we therefore consider the following VAE:

**Definition 3** (VAE–3; VAE with latent dependent diagonal decoder covariance)**.** *Consider a VAE with distributions*

$$\begin{aligned} p(\mathbf{z}) &= \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbb{I})\ , \\ q_\Phi(\mathbf{z}|\mathbf{x}) &= \mathcal{N}\big(\mathbf{z}; \boldsymbol{\nu}_\Phi(\mathbf{x}), \mathcal{T}_\Phi(\mathbf{x})\big)\ , \\ p_\Theta(\mathbf{x}|\mathbf{z}) &= \mathcal{N}\big(\mathbf{x}; \boldsymbol{\mu}_\Theta(\mathbf{z}), \Sigma_\Theta(\mathbf{z})\big)\ , \end{aligned}$$

*where $\nu_\Phi$ and $\mathcal{T}_\Phi$ are defined and parametrized analogously to VAE–1 (Def. 1). Also $\boldsymbol{\mu}_\Theta(\mathbf{z}) = \mathrm{DNN}_\mu(\mathbf{z}; W)$ is defined as for VAE–1 (i.e., according to Assumption 1). However, in contrast to VAE–1, the decoder covariance is now a diagonal matrix $\Sigma_\Theta(\mathbf{z}) = \mathrm{diag}\big(\sigma_1^2(\mathbf{z};\Theta), \ldots, \sigma_D^2(\mathbf{z};\Theta)\big)$ with elements depending on the latent code $\mathbf{z}$, implemented as*

$$\big(\sigma_1^2(\mathbf{z};\Theta), \ldots, \sigma_D^2(\mathbf{z};\Theta)\big)^{\mathrm{T}} = \mathrm{DNN}_\sigma(\mathbf{z}; M)\ , \quad (21)$$

*where $\mathrm{DNN}_\sigma(\mathbf{z}; M)$ is a standard DNN of the form:*

$$M^{L'}\mathcal{S}\big(M^{L'-1}\mathcal{S}(\cdots\mathcal{S}(M^0\mathbf{z} + \mathbf{c}^0)\cdots) + \mathbf{c}^{L'-1}\big) + \mathbf{c}^{L'},$$

*where $M^l$ denotes the weight matrix of layer $l$ and $\mathbf{c}^l$ denotes its bias terms. $\mathrm{DNN}_\sigma(\mathbf{z}; M)$ is of the same form as $\mathrm{DNN}_\mu(\mathbf{z}; W)$ but can have a different architecture and different non-linearities $\mathcal{S}(\cdot)$. Furthermore, we demand $\mathrm{DNN}_\sigma(\mathbf{z}; M)$ to always output positive values to avoid singularities. Both decoder DNNs we require to have at least one hidden layer, and we require that they are parameterized by two different sets of parameters, $W$ and $M$, respectively. We refer to such a VAE as VAE–3.*

Because of the $\mathbf{z}$-depending variances, it is obvious that Theorem 2 can not apply. Furthermore, the proof of Theorem 2 explicitly used that $\sigma^2$ does not depend on $\mathbf{z}$, so the proof for $\mathbf{z}$-dependent variances can not be a straight-forward generalization. It is still possible, however, to derive expressions for the bound in terms of entropies:

**Theorem 3.** *Consider a VAE–3 as in Def. 3. At all stationary points the ELBO of VAE–3 is then given by*

$$\begin{aligned} \mathcal{F}(\Phi,\Theta) &= \frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[q_\Phi(\mathbf{z}|\mathbf{x}^{(n)})] - \mathcal{H}[p_\Theta(\mathbf{z})] \\ &\quad - \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{q_\Phi^{(n)}}\big\{\mathcal{H}[p_\Theta(\mathbf{x}\,|\,\mathbf{z})]\big\} \\ &= \frac{1}{2N}\sum_{n=1}^{N}\sum_{h=1}^{H}\log\big(\tau_h^2(\mathbf{x}^{(n)};\Phi)\big) - \frac{D}{2}\log\big(2\pi e\big) \\ &\quad - \frac{1}{2N}\sum_{n=1}^{N}\sum_{d=1}^{D}\mathbb{E}_{q_\Phi^{(n)}}\big\{\log\big(\sigma_d^2(\mathbf{z};\Theta)\big)\big\}. \end{aligned} \quad (22)$$

*Proof Sketch.* The proof of Theorem 3 is considerably more intricate than those of Theorem 1 and Theorem 2. The main challenge is the integral over $\log(p_\Theta(\mathbf{x}^{(n)}\,|\,\mathbf{z}))$ (compare term $\mathcal{F}_2(\Phi,\Theta)$ in the proof of Theorem 1). A generalization is, however, possible again by a reformulation of the integral in terms of the entropy of $p_\Theta(\mathbf{x}^{(n)}\,|\,\mathbf{z})$. We present the full proof in Appendix B which is itself based on parallel work (Lücke, 2022) that also considers non-Gaussian distributions for elementary (non-deep) generative models. While the proof for VAE–3 shares with the proofs of Theorems 1 and 2 the use of a reparameterized VAE and rewriting of ELBO terms using entropies, it requires significantly more elaborate derivations (including details especially of the decoder DNN for the variances). □

Considering Theorem 3, observe that the final result is concise and its application to a given VAE is straight-forward (while the proof is long and technical).

Also observe that Theorem 3 is indeed a generalization of Theorem 2: if we replace $\sigma_d^2(\mathbf{z};\Theta)$ by a scalar $\sigma^2$, then we drop back to Eqn. (14). As was the case for Theorem 2, the result of Theorem 3 applies for commonly encountered conditions. For *idealized* conditions, convergence to sums of entropies as in Theorem 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, Theorem 3 states the convergence to sums of entropies for realistic conditions: for any (reasonable) finite or infinite data sets, for any stationary point, and for any variational distributions.

## 4 VERIFICATION AND ENTROPY-BASED ANALYSIS

The results of Theorems 2 and 3 and Corollary 1 are the key theoretical contributions of this work. Still, we here

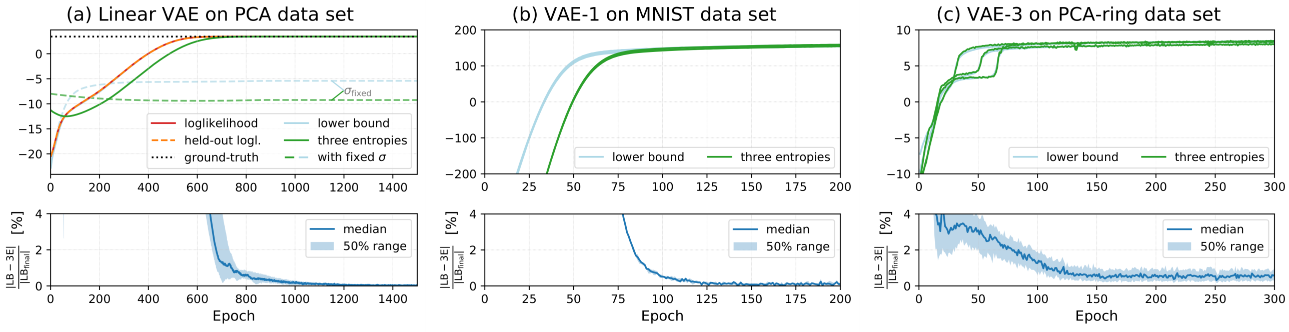


Figure 1: **Verification of the entropy results** on VAE models of increasing complexity on different data sets. *Top plots:* Absolute values of the given bounds per data point of single runs. In (a) the ELBO is essentially equal to the log-likelihood (zoom in to see). In (c) both quantities are displayed for three different seeds. *Bottom plots:* Median and interquartile range of the relative difference between the ELBO and the sum of three entropies over multiple runs (10 for (a) and (b), 100 for (c)). See Fig. 5 for further architectures and data sets, and Appendix D for details on the experiments.

study the results numerically[1], which will be instructive also about their potential practical relevance. First, we investigate how well the results apply in those vicinities of stationary points that are reached when VAEs are optimized in practice. Then, we discuss entropy-based perspectives on VAE learning: We investigate entropy-based forms of reconstruction and regularization terms and the analysis of posterior collapse with help of entropies. Moreover, we discuss improved ELBO estimation as well as fast model selection for linear VAEs based on the results presented in Section 3.

**Verification** Fig. 1 (and 5 in Appendix D) show numerical experiments for linear VAEs, and for the non-linear VAE–1 as well as VAE–3 applied to different data sets ranging from PCA data, over high-energy physics data (SUSY, Baldi et al., 2014) to the image data sets MNIST (LeCun et al., 1998) and CelebA (Liu et al., 2015). We used VAEs with simple as well as relatively complex network architectures. Details about the experimental setup are given in Appendix D. In all experiments, we observed a close correspondence of the original form of the ELBO and the sum of entropies also in vicinities of stationary points that are reached in practice. Deviations between original ELBO values and three entropy expressions were essentially all due to stochasticity in ELBO computations (we elaborate in Appendix D.6).

**Entropy-Based Analysis of VAE Learning** The results presented in Theorems 2 and 3 show that central figures in VAE optimization can be expressed solely based on entropies. For instance, reconstruction and regularization score (cf. Eqn. (2)) are at stationary points given by

$$\begin{aligned} S_{\text{reg}}(\Phi,\Theta) &= \frac{1}{N}\sum_n \mathcal{H}[q_\Phi(\mathbf{z}|\mathbf{x}^{(n)})] \;-\; \mathcal{H}[p_\Theta(\mathbf{z})]\;, \\ S_{\text{rec}}(\Theta) &= -\,\mathcal{H}[p_\Theta(\mathbf{x}^{(n)}\,|\,\mathbf{z})] \end{aligned} \tag{23}$$

[1]Code is available at github.com/Learning-with-Entropies.

where for VAE–3 the entropy $\mathcal{H}[p_\Theta(\mathbf{x}^{(n)}\,|\,\mathbf{z})]$ is replaced by the expected entropy (compare Theorem 3). VAE optimization with the ELBO can therefore be re-interpreted by recalling that differential entropies characterize the volume of the typical set, the effective volume of a distribution (Cover and Thomas, 2006): Maximizing the ELBO ultimately corresponds to minimizing the volume of the decoder's typical set, resembled by $\mathcal{H}[p_\Theta(\mathbf{x}^{(n)}\,|\,\mathbf{z})$, *and* the difference between the volume of the typical sets of prior and encoder distributions, captured in $\mathcal{H}[p_\Theta(\mathbf{z})] - \frac{1}{N}\sum_n \mathcal{H}[q_\Phi(\mathbf{z}|\mathbf{x}^{(n)})]$ (see Appendix C.1 for the full discussion). In Appendix C we also present an additional discussion of the optimization landscape based on the entropy results. In the following we make use of the entropy expressions to show how the decoder entropy $\mathcal{H}[p_\Theta(\mathbf{x}^{(n)}\,|\,\mathbf{z})]$ enables improved ELBO estimations and model selection, and how the encoder entropy $\frac{1}{N}\sum_n \mathcal{H}[q_\Phi(\mathbf{z}|\mathbf{x}^{(n)})]$ naturally provides an analysis tool for posterior collapse.

**ELBO Estimation** By using Theorem 2 it is possible to significantly simplify ELBO estimation for the wide-spread VAE–1 (Def. 1). To estimate the ELBO after convergence, we can by knowing Eqn. (14) simply use the values of the model parameters after training. Furthermore, merely the variance parameters are required.

No integrals have to be solved for ELBO estimation, while conventional estimation would require a numerical approximation of the integrals for the reconstruction term $S_{\text{rec}}(\Theta)$ in Eqn. (2). Estimations based on the original ELBO, e.g., by using Monte-Carlo approximation of integrals, is of course possible and can be sufficiently precise. However, such estimations are stochastic, require additional hyperparameters (e.g., number of samples/data points or some smoothing parameter) and can be computationally costly. Hence, one way of interpreting the result is that the problem of solving the integral of Eqn. (2) has already been solved by training the VAE, so it does not have to be solved again for the estimation of the ELBO value.[2] Also regarding the second term, $S_{\text{reg}}(\Phi, \Theta)$, which is already given in closed-form for the original ELBO, Eqn. (14) provides a simplified analytical expression (and no analytical solutions of the integral are required).

The accuracy of ELBO estimation using Theorem 2 can be quantified. In Fig. 2 (and Fig. 6 in Appendix D.2) we compare ELBO estimation using the model parameters for the entropies to the conventional and direct solution of the ELBO integrals using mini-batches.

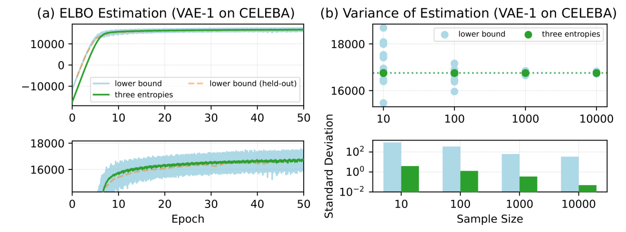


Figure 2: **ELBO Estimation** for the VAE–1 model on CelebA. (a) ELBO and the sum of the three entropies during training. The close-up reveals that the ELBO of mini-batches fluctuates around the sum of entropies. (b) Direct approximation of ELBO and three entropies for the trained model with different sample sizes, repeated 10 times. Note, that the standard deviation is depicted on logarithmic scale.

**Model Selection for Linear VAEs** In practice and for large data sets, PCA is often used as the first processing to reduce data dimensionality. Streaming PCA algorithms have found applications in 'big data' domains where storing the full dataset is prohibitively expensive, e.g., high energy physics (Guglielmo et al., 2021) and online neural data analysis (Wu et al., 2017, 2018; Migenda et al., 2021). Here we demonstrate a straight-forward application of the three entropies result to online model selection in the case of linear VAEs applied to streaming data. Linear VAEs effectively perform probabilistic PCA (e.g. Lucas et al., 2019).

For our experiments, we trained three linear VAEs of different complexity on non-stationary streaming data (see Appendix A for details). To perform model selection for such data, a very common approach is to compute the Bayesian Information Criterion (BIC) (Schwarz, 1978), which requires an estimate of the likelihood at current model parameters and for different models. For linear VAEs the ELBO can itself be used as likelihood estimation (also see Appendix A). Different ways to estimate the ELBO are conceivable. For instance, (1) the presumably most common one would use the standard stochastic estimation of the ELBO itself (using the reparameterization trick and data batches), (2) one could use the closed-form expression derived by Lucas et al. (2019, their App. C) and data batches, or (3) the ELBO could be estimated based on the closed-form analytical solution by Tipping and Bishop (1999). Notably all these three alternatives require the data and are computationally demanding.[3] Using the result derived in this contribution, in particular Corollary 1, we are provided with a novel alternative to estimate the likelihood. Importantly, and in contrast to all previous approaches, this alternative does not require the data nor does it involve any costly computations (see Corollary 1). Instead, we are merely using the variance parameters $\{\tau_h^2\}$ and $\sigma^2$ for Eqn. (20). Of course, the linear VAE still has to be trained using a method of choice, which in a streaming setting can simply be standard stochastic VAE training. But any additional cost to estimate the ELBO itself is negligible knowing Corollary 1.

In Fig. 3(a) we compare three conventionally trained linear VAEs based on BIC scores computed using stochastic ELBO estimation and using Corollary 1 for ELBO estimation, respectively. The BIC score based on stochastic ELBO estimation has a high variance due to the noise in the used data batches. While both estimations allow for model selection, the entropy-based BIC score is inherently less noisy while reliably tracking the dimensionality information in the data stream. The experiments provide evidence for the entropy-based approach to allow for reliable, low variance model selection of VAEs in streaming settings, with almost no additional computation cost. More details are given in Appendix A.

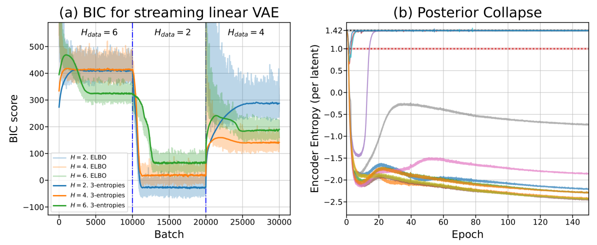


Figure 3: **(a) Streaming VAE application.** Three linear VAEs trained on streaming data with changing dimensionality. BIC score based on online ELBO estimate (transparent plots) is very noisy. The three entropies BIC score (smooth solid plots) is a estimator depending only on the model parameters. Notice, that it clearly allows for easier and more stable model selection (the lower the BIC score, the better the model). **(b) Posterior collapse** monitoring for VAE–1 on SUSY. Latent variables are collapsed if $\frac{1}{N}\sum_{n=1}^{N} \mathcal{H}[q_\Phi(z_h|\mathbf{x}^{(n)})] > 1$ (Eqn. (24)). See Appendixes C.2 and D for details on posterior collapse and experimental set-up.

[2]We remark that all our results apply for convergence based on the training data. The resulting ELBO values that can be estimated are instructive about the optimization process. The results do not directly transfer to validation or test sets (other data) unless the relevant parameters are retrained.

[3]Alternative (1) requires potentially many samples, alternative (2) involves a series of matrix multiplications, and alternative (3) the computation of the data covariance matrix and eigenvalue computations.

**Entropy-based Analysis of Posterior Collapse.** Posterior collapse is an empirically observed phenomenon: a subset of VAE latent variables ceases to participate in encoding and decoding and instead assume a high variance (He et al., 2018; Dieng et al., 2019). High encoder variance in turn means that the corresponding latents favor the optimization of the regularization score over the reconstruction score. We remark that posterior collapse is not an issue *per se*, but needs to be carefully monitored such that – for the data set with (unknown) intrinsic dimensionality at hand – a reasonable amount of latent distributions are pruned out. Lucas et al. (2019) point out that "despite a large volume of work studying posterior collapse, it has not been measured [or even defined] in a consistent way" and proposed a measurement for posterior collapse based on the KL-divergence. However, this requires two threshold values to be hand-set and is quite *ad hoc* (see Eqn. (71) and the accompanying discussion). Considering the above discussed entropy-based measures in Eqn. (23), posterior collapse can be defined much more naturally. Starting with the assumption of uncorrelated Gaussians for VAE encoder and prior, we can use the entropy-based regularization score $S_{\mathrm{reg}}(\Phi, \Theta)$ and elementary properties of entropies to derive a criterion for posterior collapse (we elaborate in Appendix C.2). Concretely, we will consider a latent distribution $q_\Phi(z_h|\mathbf{x}^{(n)})$ as collapsed if for $\delta > 0$

$$\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[q_\Phi(z_h|\mathbf{x}^{(n)})] > \mathcal{H}[p(z_h)] - \delta \ . \tag{24}$$

That is, a latent variable is collapsed if it is nearly as widely dispersed as its corresponding prior variable (and in turn provides no reliable information about the input). Note that we do not need to monitor the encoder means (as opposed to Lucas et al., 2019), the variance parameters suffice to detect posterior collapse. Fig. 3(b) visualize this definition in practice for VAE–1 throughout the optimization process with the threshold $\mathcal{H}[p(\mathbf{z})] - \delta$ set to one. The full experiments are presented in Appendix D.3 (Fig. 7).

## 5 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. Many such theoretical results are obtained for elementary models (Yedidia et al., 2001; Opper and Saad, 2001), under idealized conditions (Lücke and Henniges, 2012), or for linear boundary cases (Dai et al., 2018; Lucas et al., 2019). Idealizations and/or boundary cases are especially relevant for deep generative models, for which general and exact theoretical results are notoriously difficult to obtain. Here we studied VAEs, which are one of the most popular deep generative models. Even though VAEs range among the currently most intricate models because of the DNNs they are based on, we can here report an exact and novel theoretical result: at all stationary points the VAE learning objective (the ELBO) becomes equal to a sum of three entropies. Theorems 2 and 3 report these concise results, which imply that the ELBO value at stationary points is determined exclusively by the entropies of those distributions defining a VAE. For the most common standard VAEs, the result notably provides closed-form expressions for the ELBO at stationary points. Importantly, only very mild assumptions have to apply for the involved DNNs: We here essentially excluded weight-sharing between DNNs, and decoder DNNs required linear output units if also decoder covariances are set by DNNs. Furthermore, the derived results apply under realistic conditions, and we have numerically verified that the derived entropy-form of the ELBO has very high accuracy also in those vicinities of stationary points reached in practice. The reported results consequently go beyond being of purely theoretical relevance, i.e., the derived expressions can directly be used in practice for any task that requires the estimation of the ELBO itself. To give an intuition, we discussed (theoretically and numerically) tasks such as ELBO estimation, model selection for streaming data, or the analysis of posterior collapse. We do stress, however, that the main contribution remains the theoretical result itself: the ELBOs of the most common VAEs converge to entropy sums.

Future work will be based on the observation that only subsets of parameters have to be at stationary points, which can allow for purely entropy-based learning objectives to optimize, e.g., encoder DNNs. Also the investigation of VAEs with less standard prior distributions as well as the investigation of stacked VAEs or other deep generative models (e.g., Bond-Taylor et al., 2022) represent natural future research directions.

**Acknowledgments**

This work is funded by the German Research Foundation (DFG) within the priority program SPP 2298 "Theoretical Foundations of Deep Learning" - project 464104047 (FI 2583/1-1 and LU 1196/9-1).

## References

Zeyuan Allen-Zhu and Yuanzhi Li. First efficient convergence for streaming k-PCA: a global, gap-free, and near-optimal rate. In *58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 487–492, 2017.

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. In *International Conference on Learning Representations*, 2018.

Andrea Asperti. Variance loss in variational autoencoders. In *International Conference on Machine Learning, Op-*

*timization, and Data Science*, pages 297–308. Springer, 2020.

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

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1):1–9, 2014.

Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G Willcocks. Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models. *IEEE transactions on pattern analysis and machine intelligence*, 44:7327–7347, 2022.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, 2016.

Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

Chi-Ning Chou and Mien B Wang. ODE-inspired analysis for the biological version of Oja's rule in solving streaming PCA. In *Conference on Learning Theory*, pages 1339–1343. PMLR, 2020.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2006.

Bin Dai, Yu Wang, John Aston, Gang Hua, and David Wipf. Connections with robust PCA and the role of emergent sparsity in variational autoencoder models. *The Journal of Machine Learning Research*, 19(1): 1573–1614, 2018.

Adji B Dieng, Yoon Kim, Alexander M Rush, and David M Blei. Avoiding latent variable collapse with generative skip models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2397–2405. PMLR, 2019.

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

Garoe Dorta, Sara Vicente, Lourdes Agapito, Neill DF Campbell, and Ivor Simpson. Training VAEs under structured residuals. *arXiv preprint arXiv:1804.01050*, 2018.

Giuseppe Di Guglielmo, Farah Fahim, Christian Herwig, Manuel B Valentin, and Javier Duarte et. al. A reconfigurable neural network ASIC for detector front-end data compression at the HL-LHC. *IEEE Transactions on Nuclear Science*, 68(8):2179–2186, 2021.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. Lagging inference networks and posterior collapse in variational autoencoders. In *International Conference on Learning Representations*, 2018.

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. *ICLR*, 2017.

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.

Guy Holtzman, Adam Soffer, and Dan Vilenchik. A greedy anytime algorithm for sparse PCA. In *Conference on Learning Theory*, pages 1939–1956. PMLR, 2020.

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

Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019.

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

Daniel Kunin, Jonathan M Bloom, Aleksandrina Goeva, and Cotton Seed. Loss landscapes of regularized linear autoencoders. *arXiv preprint arXiv:1901.08168*, 2019.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

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

Jörg Lücke. On the convergence of the ELBO to entropy sums. *arXiv preprint arXiv:2209.03077*, 2022.

Jörg Lücke and Marc Henniges. Closed-form entropy limits - A tool to monitor likelihood optimization of probabilistic generative models. In *Proc. AISTATS*, pages 731–740. PMLR, 2012.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International conference on machine learning*, pages 2391–2400. PMLR, 2017.

Nico Migenda, Ralf Möller, and Wolfram Schenck. Adaptive dimensionality reduction for neural network-based online principal component analysis. *PLoS ONE*, 16(3): e0248896, 2021.

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

Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 1747–1756, 2016.

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

Danilo J Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

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

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

Oleh Rybkin, Kostas Daniilidis, and Sergey Levine. Simple and effective vae training with calibrated decoders. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 9179–9189. PMLR, 18–24 Jul 2021.

Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978.

Alexander Shekhovtsov, Dmitrij Schlesinger, and Boris Flach. VAE approximation error: ELBO and exponential families. In *International Conference on Learning Representations*, 2022.

Anand K Subramanian. PyTorch-VAE. `https://github.com/AntixK/PyTorch-VAE`, 2020.

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

Ernst Wit, Edwin vd Heuvel, and Jan-Willem Romeijn. 'All models are wrong...': an introduction to model uncertainty. *Statistica Neerlandica*, 66(3):217–236, 2012.

Tong Wu, Wenfeng Zhao, Hongsun Guo, Hubert H Lim, and Zhi Yang. A streaming PCA VLSI chip for neural data compression. *IEEE Transactions on Biomedical Circuits and Systems*, 11(6):1290–1302, 2017.

Tong Wu, Wenfeng Zhao, Edward Keefer, and Zhi Yang. Deep compressive autoencoder for action potential compression in large-scale neural recording. *Journal of Neural Engineering*, 15(6):066019, 2018.

Huan Xu, Constantine Caramanis, and Shie Mannor. Principal component analysis with contaminated data: The high dimensional case. In *Conference on Learning Theory*, 2010.

Dmitry Yarotsky. Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114, 2017.

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

## A LINEAR VAES AND STREAMING APPLICATIONS

For the proof of Corollary 1 note that for the linear DNN of the decoder (i.e., for the matrix multiplication) of Eqn. (18) the same reparametrization is possible as was used for VAE–1 (i.e., using $\tilde{W}$ with constraint columns instead of $W$ in Eqn. (18)). Furthermore, no conditions were imposed on the encoder DNN for Theorem 2, and the used properties of the stationary points (Eqn. (7)) are the same for the linear VAE as for the VAEs above. Theorem 2 consequently applies for the linear VAE as a special case of VAE–1.

The result of Corollary 1 then 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 $\sigma^2$ and $\tau_h^2$. 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. Linear VAEs can be used to recover the maximum likelihood solution (compare Dai et al., 2018; Lucas et al., 2019) of probabilistic PCA (p-PCA). Lucas et al. (2019), for instance, showed that no stable stationary points of the variational lower bound exist other than 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. (20) is equal to the p-PCA log-likelihood.

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

**Corollary 2.** *Consider the linear VAE defined by Eqns.* (18) *and* (19) *with decoder parameters* $\Theta = (\sigma^2, W, \boldsymbol{\mu}_0)$*. Then after convergence, the parameters* $(\sigma^2, W, \boldsymbol{\mu}_0)$ *represent the maximum likelihood solution for p-PCA, and the value of the log-likelihood* $\mathcal{L}(\Theta)$ *is given by:*

$$\mathcal{L}(\Theta) = \frac{1}{2}\sum_h \log\big(\tau_h^2\big) - \frac{D}{2}\log\big(2\pi e\sigma^2\big), \tag{25}$$

*where* $\tau_h^2$ *are the learned variances of the VAE encoder.*

*Proof.* As Eqn. (20) 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 (e.g., 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. (25) simply use the well-known closed-form solution of the p-PCA likelihood (Tipping and Bishop, 1999):

$$\mathcal{L}(\Theta) = -\frac{D}{2}\log(2\,\pi) - \frac{1}{2}\log\big(\det(C)\big) - \frac{1}{2}\,\mathrm{Tr}\,(C^{-1}S), \tag{26}$$

$$\text{where } C = WW^T + \sigma^2\mathbb{I} \text{ and where } \quad S = \frac{1}{N}\sum_n(\mathbf{x}^{(n)} - \boldsymbol{\mu})(\mathbf{x}^{(n)} - \boldsymbol{\mu})^T$$

is the data covariance matrix. At convergence, we thus have two alternatives to compute the log-likelihood: Eqn. (26) and Eqn. (25). The two expressions differ, however. While both are closed-form, the well-known p-PCA likelihood (Eqn. (26)) requires the data in the form of the data covariance matrix $S$. And even neglecting the computational effort to compute $S$, the computation of (Eqn. (26)) is much more expensive than computing (25): 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 Eqn. (25) no such computations nor data is required (although we require a linear VAE that has sufficiently converged). The derived result of Eqn. (25) can therefore be of theoretical and practical relevance especially considering the exceptionally widespread use of PCA in Machine Learning, Statistics and beyond.

**Streaming Applications** One concrete practical example is PCA applied to streaming data, which is an increasingly common setting especially in recent years, see e.g., Allen-Zhu and Li (2017) for an overview and Chou and Wang (2020) for a recent example.. In such a setting the application of a linear VAE is straight-forward, and it can be operated as alternative or in parallel to other PCA algorithms. For streaming data (as for other data) the value of the likelihood itself can be of high interest, e.g., to monitor the fit to the data or for the selection of PCA dimensions. In this context, Eqn. (25) 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. (25) can be used to track the changes of the likelihood 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 use the linear VAE (Appendix D). The result can then directly be compared to the well known closed-form log-likelihood (Eqn. (26)) as well as to sampling based approximations of the variational lower bound. Appendix D also shows empirically that the derived three entropy expressions match the lower bound with high accuracy also in vicinity of stationary points that are reached in practice.

For our experiments with streaming VAEs, we construct an artificial dataset that comprises three chunks of data of dimensionality $D = 10$. Each chunk is limited to have different intrinsic dimensionality $H_{data} \in \{2, 4, 6\}$ by setting the remaining Gaussian covariance matrix eigenvalues to zero. We feed this dataset into three separate linear VAEs with latent space dimensionality $H$ of 2, 4, and 6 by providing small batches of 10 data points each in order to simulate the streaming training regime.

To compute the BIC for model $\mathcal{M}$ we used the following equation (Wit et al., 2012):

$$\mathrm{BIC}(\mathcal{M}) = -2\,\hat{l}(X) + P\log(N), \tag{27}$$

where $\hat{l}(X)$ is the maximum log-likelihood of the data $X$, $N$ is the number of data points, and $P$ is the number of parameters which have only point estimation (*maximum likelihood* or *maximum a posteriori*). In our experiment with linear VAEs $P$ is equal to the number of elements in the generative matrix plus one for the observation noise variance: $P = HD + 1$. As learning the posterior distribution for a subset of the parameters (in our case it is posterior over $\mathbf{z}$) for VAEs yields a lower bound of the log-likelihood, effectively integrating them out, we exclude the latent points $\mathbf{z}$ and the amortized encoder from the set of point-estimated parameters.

In the numerical experiment for training we used 100 (reparameterized) random samples for each data point to estimate the ELBO and the gradient. Each batch of 10 data points was used to run only one step of ADAM optimization over the parameters. We set the learning rate to 0.002, which we found to be a good trade-off between the speed of convergence and the update noise for such a simple linear model.

## B PROOF OF THEOREM 3

The proof of Theorem 3 which applies for general VAE decoders of VAE–3 of Def. 3 follows a similar intuition as the proofs of Theorems 1 and 2. In many aspects it is, however, significantly more intricate. An important element shared by the more general proof and the proofs for VAE–1 is a reparameterization of the original VAE model by using part of the DNN weights of the decoder to parameterize the prior distribution.

Before we reparameterize, let us reconsider the DNNs of the VAE-3 decoder which are given by:

$$\boldsymbol{\mu}_\Theta(\mathbf{z}) = \boldsymbol{\mu}(\mathbf{z}, W) \;\text{ and} \tag{28}$$

$$\Sigma_\Theta(\mathbf{z}) = \mathrm{diag}\big(\tilde{\sigma}_1(\mathbf{z}, M), \ldots, \tilde{\sigma}_D(\mathbf{z}, M)\big) \;\text{ and }\; \tilde{\boldsymbol{\sigma}}(\mathbf{z}, M) = \big(\tilde{\sigma}_1(\mathbf{z}, M), \ldots, \tilde{\sigma}_D(\mathbf{z}, M)\big)^{\mathrm{T}}. \tag{29}$$

The expressions make explicit that the two DNNs depend on two different sets of parameters ($W$ and $M$, respectively). Furthermore, let us reiterate the DNNs themselves which are, in more detail, given as follows:

$$\boldsymbol{\mu}(\mathbf{z}, W) = \mathrm{DNN}_\mu(\mathbf{z}; W) = W^L\mathcal{S}(W^{L-1}\mathcal{S}(\cdots\mathcal{S}(W^0\mathbf{z} + \mathbf{b}^0)\cdots + \mathbf{b}^{L-1}) + \mathbf{b}^L, \tag{30}$$

$$\tilde{\boldsymbol{\sigma}}(\mathbf{z}, M) = \mathrm{DNN}_\sigma(\mathbf{z}; M) = M^L\mathcal{S}(M^{L-1}\mathcal{S}(\cdots\mathcal{S}(M^0\mathbf{z} + \,\mathbf{c}^0)\cdots + \mathbf{c}^{L-1}) + \mathbf{c}^L, \tag{31}$$

where $W^l$ and $\mathbf{b}^l$ are the weight matrices and biases of $\mathrm{DNN}_\mu(\mathbf{z}; W)$, and where $M^l$ and $\mathbf{c}^l$ are the weight matrices and biases of $\mathrm{DNN}_\sigma(\mathbf{z}; M)$. $\mathrm{DNN}_\mu(\mathbf{z}; W)$ and $\mathrm{DNN}_\sigma(\mathbf{z}; M)$ *can* have different numbers of layers (in the main text we used $L$ and $L'$ for maximal layer indices). To simplify notation for the proof, we dropped the distinction (we just use $L$ for both DNNs), and mainly $\mathrm{DNN}_\sigma(\mathbf{z}; M)$ will be of interest. $W$ and $M$ we take to contain all weight matrices and biases for their corresponding DNN. Both DNNs have a linear output layer and point-wise non-linearities $\mathcal{S}(\cdot)$, and each of these two properties is a standard assumption for DNNs (compare, e.g., Yarotsky, 2017; Arora et al., 2018). The non-linearities also *can* differ between the DNNs, and they can be different from layer to layer. For the variance DNN we assume that learning does not result in degeneracies at stationary points, i.e., we assume that $\tilde{\sigma}_d(\mathbf{z};\Theta)$ is never equal to zero in order to avoid singularities. In practice, variances equal zero mean perfect data reconstruction, so our assumption at stationary points only excludes rather artificial conditions.

After reiterating the DNNs in more detail, the first step of the proof is now a reparameterization of VAE–3 along the same lines as done for VAE–1 in Section 3.1, i.e., we consider the following auxiliary VAE model:

$$p_A(\tilde{\mathbf{z}}) = \mathcal{N}(\tilde{\mathbf{z}};\mathbf{0},A) \ , \tag{32}$$

$$q_\Phi(\tilde{\mathbf{z}}|\mathbf{x}) = \mathcal{N}\big(\tilde{\mathbf{z}};\tilde{\boldsymbol{\nu}}_\Phi(\mathbf{x}),\tilde{\mathcal{T}}_\Phi(\mathbf{x})\big) \ , \tag{33}$$

$$p_\Theta(\mathbf{x}|\tilde{\mathbf{z}}) = \mathcal{N}\big(\mathbf{x};\tilde{\boldsymbol{\mu}}_\Theta(\tilde{\mathbf{z}}),\tilde{\Sigma}_\Theta(\tilde{\mathbf{z}})\big) \ , \tag{34}$$

where decoder mean $\tilde{\boldsymbol{\mu}}_\Theta(\tilde{\mathbf{z}}) = \mathrm{DNN}_{\tilde{\boldsymbol{\mu}}}(\tilde{\mathbf{z}};W)$ and decoder covariance $\tilde{\Sigma}_\Theta(\mathbf{z}) = \mathrm{DNN}_{\tilde{\sigma}}(\tilde{\mathbf{z}};M)$ are the same as the DNNs for VAE–3 in Eqns. (30) and (31) with the exception that their first weights ($W^0$ and $M^0$) now have their column vectors constrained to unit norm. The prior distribution is now parameterized by the diagonal matrix $A = \mathrm{diag}\,(\tilde{\alpha}_1,\ldots,\tilde{\alpha}_H)$. For $A$ we used with $\tilde{\alpha}_h$ instead of $(\alpha_h)^2$ an abbreviation analogous to the one for the observable variances (and we assume $\tilde{\alpha}_h > 0$ for all $h$).

With the same argumentation as given in Section 3, the reparameterized VAE does parameterize the same model as VAE–3 of Def. 3. As preparation for the proof steps below, we abbreviate the decoder DNN for the variances (Eqn. (31)) as follows:

$$\tilde{\sigma}_d(\tilde{\mathbf{z}};\Theta) \ = \ \big(\mathrm{DNN}_{\tilde{\sigma}}(\tilde{\mathbf{z}};M)\big)_d \ = \ \sum_j M^L_{dj}\,\varphi_j(\tilde{\mathbf{z}};M) \ + \ c^L_d, \tag{35}$$

where $\varphi_j(\tilde{\mathbf{z}};\Theta)$ is simply the remainder of the DNN after the lowest weights are removed (i.e., $\varphi_j(\tilde{\mathbf{z}};\Theta)$ is the output of unit $j$ in layer $L-1$). We demanded in Def. 3 that the network has at least one hidden layer, i.e., $L \geq 1$, which ensures that the weights $M^L$ and $M^0$ represent two different weight matrices (and $\mathbf{c}^L$ and $\mathbf{c}^0$ two different bias vectors). The constraint on $M^0$ (i.e., that it has unit length columns) does therefore not effect $M^L$.

Using the parameterization in Eqns. (32) and (34) we could now in principle prove Theorem 3 explicitly. However, for our purposes it is more convenient to make use of a recent generalization of the proof of Theorem 2 which includes also non-Gaussian distributions (see Lücke, 2022). We note that the work by Lücke (2022) is work in parallel to this contribution, and that it does not treat VAEs nor other DNN-based models. We will, however, show in the following how the VAE–3 model can be shown to satisfy the conditions for which the general result applies.

The general result (see Lücke, 2022, Theorem 1) applies for any generative model with one set of latents and one set of observables, where latent and observable distributions are both exponential family distributions (with constant base measure). If the link between latents and observables then fulfills a specific parameterization condition then we can conclude a convergence to entropy sums. The crucial property of VAE–3 is that the link is given by the decoder DNNs of Eqns. (30) and (31). The task in the following will be to use the reparameterized version of VAE–3 in Equations (32) to (34), and to then show that the conditions for the general theorem (Lücke, 2022, Theorem 1) are fulfilled.

The conditions for the theorem are formulated for the exponential family parameterization of a generative model's distributions. An exponential form of the VAE in Equations (32) to (34) can be given as follows. First, the Gaussian prior can be rewritten as:

$$p_A(\tilde{\mathbf{z}}) = \mathcal{N}(\tilde{\mathbf{z}};\mathbf{0},A) = h(\tilde{\mathbf{z}})\exp\big(\boldsymbol{\xi}(A)^{\mathrm{T}}\,\mathbf{T}(\tilde{\mathbf{z}}) \ - \ \mathcal{A}(\boldsymbol{\xi}(A))\big) = p_{\boldsymbol{\xi}(A)}(\tilde{\mathbf{z}}) \tag{36}$$

$$\text{where} \ \ \mathbf{T}(\tilde{\mathbf{z}}) = \tilde{\mathbf{z}}\odot\tilde{\mathbf{z}}, \ \ \xi_h(A) = -\frac{1}{2\tilde{\alpha}_h} \ \ \text{and} \ \ h(\tilde{\mathbf{z}}) = (2\pi)^{-\frac{H}{2}}, \ \ \mathcal{A}\big(\boldsymbol{\xi}(A)\big) = \frac{1}{2}\sum_h \log(\tilde{\alpha}_h), \tag{37}$$

with $\odot$ denoting point-wise multiplication, and with $\boldsymbol{\xi}(A)$ denoting the natural parameters of the prior. The exponential family form of the observable distribution is more intricate because it contains the DNNs and it has a non-zero mean. An exponential family parameterization can be given as follows:

$$p_\Theta(\mathbf{x}|\tilde{\mathbf{z}}) = \mathcal{N}\big(\mathbf{x};\tilde{\boldsymbol{\mu}}_\Theta(\tilde{\mathbf{z}}),\tilde{\Sigma}_\Theta(\tilde{\mathbf{z}})\big) = h(\mathbf{x})\exp\Big(\boldsymbol{\eta}(\tilde{\mathbf{z}};\Theta)^{\mathrm{T}}\,\mathbf{T}(\mathbf{x}) \ - \ \mathcal{A}\big(\boldsymbol{\eta}(\tilde{\mathbf{z}};\Theta)\big)\Big) = p_{\boldsymbol{\eta}(\tilde{\mathbf{z}};\Theta)}(\mathbf{x}) \tag{38}$$

$$\text{where} \ \ \mathbf{T}(\mathbf{x}) = \begin{pmatrix} \mathbf{x} \\ \mathbf{x}\odot\mathbf{x} \end{pmatrix}, \ \ \boldsymbol{\eta}(\tilde{\mathbf{z}};\Theta) = \begin{pmatrix} \boldsymbol{\mu}(\tilde{\mathbf{z}};W)\odot\mathbf{s}(\tilde{\mathbf{z}};M) \\ -\frac{1}{2}\,\mathbf{s}(\tilde{\mathbf{z}};M) \end{pmatrix}, \ \text{with} \ \ \mathbf{s}(\tilde{\mathbf{z}};M) = \begin{pmatrix} \frac{1}{\tilde{\sigma}_1(\tilde{\mathbf{z}};M)} \\ \vdots \\ \frac{1}{\tilde{\sigma}_D(\tilde{\mathbf{z}};M)} \end{pmatrix}, \tag{39}$$

$$\text{and} \quad h(\mathbf{x}) = (2\pi)^{-\frac{D}{2}}, \;\; \mathcal{A}\big(\boldsymbol{\eta}(\tilde{\mathbf{z}};\Theta)\big) = \frac{1}{2}\sum_d \Big( \frac{\big(\mu_d(\tilde{\mathbf{z}};W)\big)^2}{\tilde{\sigma}_d(\tilde{\mathbf{z}};M)} \,+\, \log\big(\tilde{\sigma}_d(\tilde{\mathbf{z}};M)\big)\Big). \tag{40}$$

The reformulation of the VAE in terms of an exponential family parameterization (and with $h(\tilde{\mathbf{z}})$ and $h(\mathbf{x})$ being constant base measures) is a first prerequisite for the result in (Lücke, 2022) to be applicable. The crucial property that is needed for the theorem to apply now concerns the natural parameter vectors $\boldsymbol{\xi}(A)$ and $\boldsymbol{\eta}(\tilde{\mathbf{z}};\Theta)$, where the second is defined by potentially intricate and large DNNs. The condition for the first function, $\boldsymbol{\xi}(A)$, is relatively easy to show. We do require that for any vector $\boldsymbol{v} \in \mathbb{R}^H$ applies that:

$$\mathcal{I}^{\mathrm{T}}_{(A)}\,\boldsymbol{v} \;=\; \mathbf{0} \quad\Rightarrow\quad \boldsymbol{\xi}^{\mathrm{T}}_{(A)}\,\boldsymbol{v} \;=\; 0\,, \;\; \text{where} \;\; \big(\mathcal{I}^{\mathrm{T}}_{(A)}\big)_{hh'} \;=\; \tfrac{\partial}{\partial\tilde{\alpha}_h}\,\xi_{h'}(A) \tag{41}$$

is the (transposed) Jacobian of $\boldsymbol{\xi}(A)$. Using Eqn. (37) for $\xi_{h'}(A)$, we get $\mathcal{I}^{\mathrm{T}}_{(A)} \;=\; \frac{1}{2}\,\mathrm{diag}(\tilde{\alpha}_1^{-2},\ldots,\tilde{\alpha}_H^{-2})$, i.e., $\mathcal{I}^{\mathrm{T}}_{(A)}$ is a squared and diagonal matrix with positive entries on the diagonal. Condition (41) is consequently almost trivially fulfilled as the first equation implies $\boldsymbol{v}=\mathbf{0}$ which implies $\boldsymbol{\xi}^{\mathrm{T}}_{(A)}\,\boldsymbol{v} \;=\; 0$.

The corresponding condition for the function $\boldsymbol{\eta}(\tilde{\mathbf{z}};\Theta)$ is the for our purposes crucial part because this function (defined by Eqn. (39)) links the latents to the observables using the decoder DNNs. For the link $\boldsymbol{\eta}(\tilde{\mathbf{z}};\Theta)$, it will turn out to be easier if we show a more general result for VAEs than stated by Theorem 3. Concretely, it will be easier to proof that equality to entropy sums holds also if only a subset of the parameters $\Theta$ have converged to a stationary point. We denote this subset by $\boldsymbol{\theta}$ which we take to contain all weights and biases of the output layer of the variance network, i.e., $M^L$ and $\mathbf{c}^L$ (see Appendix B). We take $\boldsymbol{\theta}$ to contain all these weights and biases arranged in one long column vector with scalar entries. All other parameters we take as being fixed (but they can have arbitrary values).

For the link function $\boldsymbol{\eta}_{(\tilde{\mathbf{z}};\Theta)}$, we now have to show (see Lücke, 2022, Def. C) that the following specific condition is true:

For any function $\boldsymbol{g}:\mathbb{R}^H\rightarrow\mathbb{R}^{2D}$ from the latents to the space of natural parameters $\boldsymbol{\eta}$ of the observable distribution it has to hold that

$$\int \mathcal{J}^{\mathrm{T}}_{(\tilde{\mathbf{z}};\boldsymbol{\theta})}\,\boldsymbol{g}(\tilde{\mathbf{z}})\,\mathrm{d}\tilde{\mathbf{z}} \;=\; \mathbf{0} \;\;\Rightarrow\;\; \int \boldsymbol{\eta}^{\mathrm{T}}_{(\tilde{\mathbf{z}};\Theta)}\,\boldsymbol{g}(\tilde{\mathbf{z}})\,\mathrm{d}\tilde{\mathbf{z}} \;=\; 0\,, \tag{42}$$

where $\mathcal{J}^{\mathrm{T}}_{(\tilde{\mathbf{z}};\boldsymbol{\theta})}$ denotes the (transposed) Jacobian of $\boldsymbol{\eta}_{(\tilde{\mathbf{z}};\Theta)}$ constructed only using the subset $\boldsymbol{\theta}$, i.e.,

$$\mathcal{J}^{\mathrm{T}}_{(\tilde{\mathbf{z}};\,\boldsymbol{\theta})} \;=\; \Big(\tfrac{\partial}{\partial\boldsymbol{\theta}}\eta_1(\tilde{\mathbf{z}};\Theta),\ldots,\tfrac{\partial}{\partial\boldsymbol{\theta}}\eta_{2D}(\tilde{\mathbf{z}};\Theta)\Big)\,. \tag{43}$$

Some intuition on how the general proof then works is the following. Similar to the proof of Theorem 1, we rewrite the different terms of the ELBO, Eqn. (2), in terms of entropies. For $\mathcal{F}_2(\Phi,\boldsymbol{\theta})$ this results (in exponential family notation) in the following expression:

$$\mathcal{F}_2(\Phi,\Theta) \;=-\frac{1}{N}\sum_n \mathbb{E}_{q^{(n)}_\Phi}\{\mathcal{H}[p_{\boldsymbol{\eta}(\tilde{\mathbf{z}};\Theta)}(\mathbf{x})]\} \;-\; \int \boldsymbol{\eta}^{\mathrm{T}}_{(\tilde{\mathbf{z}};\,\Theta)}\,\boldsymbol{g}(\tilde{\mathbf{z}})\,\mathrm{d}\tilde{\mathbf{z}} \tag{44}$$

where $\boldsymbol{g}(\tilde{\mathbf{z}})$ is a function whose concrete properties are not relevant for the proof. At stationary points of the parameters $\boldsymbol{\theta}$ we can show that $\int \mathcal{J}^{\mathrm{T}}_{(\tilde{\mathbf{z}};\boldsymbol{\theta})}\,\boldsymbol{g}(\tilde{\mathbf{z}})\,\mathrm{d}\tilde{\mathbf{z}} \;=\; \mathbf{0}$ applies for the same function $\boldsymbol{g}(\tilde{\mathbf{z}})$. So if condition (42) holds, then the integral in expression (44) vanishes, and $\mathcal{F}_2(\Phi,\Theta)$ becomes equal to an (expected) entropy. For more details see (Lücke, 2022).

To show that condition (42) holds in our case, we now write $\boldsymbol{g}(\tilde{\mathbf{z}}) = \begin{pmatrix} \boldsymbol{g}^{\mathrm{A}}(\tilde{\mathbf{z}}) \\ \boldsymbol{g}^{\mathrm{B}}(\tilde{\mathbf{z}}) \end{pmatrix}$, i.e., we take the function to consist of two functions $\boldsymbol{g}^{\mathrm{A}}(\tilde{\mathbf{z}})$ and $\boldsymbol{g}^{\mathrm{B}}(\tilde{\mathbf{z}})$ mapping to $\mathbb{R}^D$. Now using the definition of $\boldsymbol{\eta}_{(\tilde{\mathbf{z}};\Theta)}$ in Equation (39), we obtain:

$$\mathbf{0} = \int \mathcal{J}^{\mathrm{T}}_{(\tilde{\mathbf{z}};\boldsymbol{\theta})}\,\boldsymbol{g}(\tilde{\mathbf{z}})\,\mathrm{d}\tilde{\mathbf{z}} \tag{45}$$

$$\Rightarrow\quad \mathbf{0} = \int \Big(\frac{\partial}{\partial\boldsymbol{\theta}}\big(\boldsymbol{\mu}(\tilde{\mathbf{z}};W)\odot\mathbf{s}(\tilde{\mathbf{z}};M)\big)^{\mathrm{T}},\, -\frac{1}{2}\,\frac{\partial}{\partial\boldsymbol{\theta}}\big(\mathbf{s}(\tilde{\mathbf{z}};M)\big)^{\mathrm{T}}\Big)\begin{pmatrix} \boldsymbol{g}^{\mathrm{A}}(\tilde{\mathbf{z}}) \\ \boldsymbol{g}^{\mathrm{B}}(\tilde{\mathbf{z}}) \end{pmatrix} d\tilde{\mathbf{z}} \tag{46}$$

$$\Rightarrow\quad \mathbf{0} = \int \Big(\sum_{d=1}^{D}\,\mu_d(\tilde{\mathbf{z}};W)\,\big(\frac{\partial}{\partial\boldsymbol{\theta}}s_d(\tilde{\mathbf{z}};M)\big)\,g^{\mathrm{A}}_d(\tilde{\mathbf{z}}) \,-\, \frac{1}{2}\sum_{d=1}^{D}\big(\frac{\partial}{\partial\boldsymbol{\theta}}s_d(\tilde{\mathbf{z}};M)\big)\,g^{\mathrm{B}}_d(\tilde{\mathbf{z}})\Big)\,d\tilde{\mathbf{z}}\,. \tag{47}$$

As we took $\boldsymbol{\theta}$ to contain all weights $M^L_{dj}$ of the output layer of $\mathrm{DNN}_{\tilde{\sigma}}(\tilde{\mathbf{z}}; M)$ and the corresponding biases $c^L_d$, we conclude that Eqn. (47) implies that

$$\text{for all } d \text{ and } j: \quad \int \sum_{d'=1}^{D} \Big( \mu_{d'}(\tilde{\mathbf{z}}; W) \, \big( \frac{\partial \, s_{d'}(\tilde{\mathbf{z}}; M)}{\partial M^L_{dj}} \big) \, g^{\mathrm{A}}_{d'}(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \big( \frac{\partial \, s_{d'}(\tilde{\mathbf{z}}; M)}{\partial M^L_{dj}} \big) \, g^{\mathrm{B}}_{d'}(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0, \tag{48}$$

$$\text{and for all } d: \quad \int \sum_{d'=1}^{D} \Big( \mu_{d'}(\tilde{\mathbf{z}}; W) \, \big( \frac{\partial \, s_{d'}(\tilde{\mathbf{z}}; M)}{\partial c^L_{d}} \big) \, g^{\mathrm{A}}_{d'}(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \big( \frac{\partial \, s_{d'}(\tilde{\mathbf{z}}; M)}{\partial c^L_{d}} \big) \, g^{\mathrm{B}}_{d'}(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0, \tag{49}$$

We now evaluate the derivatives using the definition of $\mathbf{s}(\tilde{\mathbf{z}}; M)$ in Eqn. (39) and by inserting expression (35) for $\tilde{\sigma}_{d'}(\tilde{\mathbf{z}}; \boldsymbol{\theta})$. We obtain:

$$\frac{\partial \, s_{d'}(\tilde{\mathbf{z}}; M)}{\partial M^L_{dj}} = \frac{\partial}{\partial M^L_{dj}} \big( \frac{1}{\tilde{\sigma}_d(\tilde{\mathbf{z}}; M)} \big) \; = \; - \, \delta_{dd'} \, \frac{\varphi_j(\mathbf{z}; M)}{\big( \tilde{\sigma}_{d'}(\tilde{\mathbf{z}}; M) \big)^2}, \tag{50}$$

$$\text{and} \quad \frac{\partial \, s_{d'}(\tilde{\mathbf{z}}; M)}{\partial c^L_{d}} = \frac{\partial}{\partial c^L_{d}} \big( \frac{1}{\tilde{\sigma}_{d'}(\tilde{\mathbf{z}}; M)} \big) \; = \; - \, \delta_{dd'} \, \frac{1}{\big( \tilde{\sigma}_{d}(\tilde{\mathbf{z}}; M) \big)^2}. \tag{51}$$

By inserting the derivatives into expressions (48) and (49), we obtain:

$$\text{for all } d \text{ and } j: \quad \int \Big( \mu_d(\tilde{\mathbf{z}}; W) \, \frac{\varphi_j(\mathbf{z}; M)}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{A}}_d(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \frac{\varphi_j(\mathbf{z}; M)}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{B}}_d(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0, \tag{52}$$

$$\text{and for all } d: \quad \int \Big( \mu_d(\tilde{\mathbf{z}}; W) \, \frac{1}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{A}}_d(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \frac{1}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{B}}_d(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0. \tag{53}$$

We now multiply expression (52) by $W^L_{dj}$ and then sum over $j$ to obtain:

$$\text{For all } d: \quad \sum_j W^L_{dj} \int \Big( \mu_d(\tilde{\mathbf{z}}; W) \, \frac{\varphi_j(\mathbf{z}; M)}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{A}}_d(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \frac{\varphi_j(\mathbf{z}; M)}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{B}}_d(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0, \tag{54}$$

$$\Rightarrow \quad \text{For all } d: \quad \int \Big( \mu_d(\tilde{\mathbf{z}}; W) \, \frac{\sum_j \, W^L_{dj} \, \varphi_j(\mathbf{z}; M)}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{A}}_d(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \frac{\sum_j \, W^L_{dj} \, \varphi_j(\mathbf{z}; M)}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{B}}_d(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0. \tag{55}$$

Expression (53) we multiply by $c^L_d$ to obtain:

$$\text{For all } d: \quad \int \Big( \mu_d(\tilde{\mathbf{z}}; W) \, \frac{c^L_d}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{A}}_d(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \frac{c^L_d}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{B}}_d(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0. \tag{56}$$

By adding equations (55) and (56) we can conclude:

$$\text{For all } d: \quad \int \Big( \mu_d(\tilde{\mathbf{z}}; W) \, \frac{\sum_j \, W^L_{dj} \, \varphi_j(\mathbf{z}; M) \, + \, c^L_d}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{A}}_d(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \frac{\sum_j \, W^L_{dj} \, \varphi_j(\mathbf{z}; M) \, + \, c^L_d}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{B}}_d(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0 \tag{57}$$

$$\Rightarrow \quad \text{For all } d: \quad \int \Big( \mu_d(\tilde{\mathbf{z}}; W) \, \frac{\tilde{\sigma}_d(\tilde{\mathbf{z}}; M)}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{A}}_d(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \frac{\tilde{\sigma}_d(\tilde{\mathbf{z}}; M)}{\big( \tilde{\sigma}_d(\tilde{\mathbf{z}}; M) \big)^2} \, g^{\mathrm{B}}_d(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0 \tag{58}$$

$$\Rightarrow \quad \text{For all } d: \quad \int \Big( \mu_d(\tilde{\mathbf{z}}; W) \, \frac{1}{\tilde{\sigma}_d(\tilde{\mathbf{z}}; M)} \, g^{\mathrm{A}}_d(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \frac{1}{\tilde{\sigma}_d(\tilde{\mathbf{z}}; M)} \, g^{\mathrm{B}}_d(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0. \tag{59}$$

By summing expression (59) over $d$ we finally obtain:

$$\sum_d \int \Big( \mu_d(\tilde{\mathbf{z}}; W) \, \frac{1}{\tilde{\sigma}_d(\tilde{\mathbf{z}}; M)} \, g^{\mathrm{A}}_d(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \frac{1}{\tilde{\sigma}_d(\tilde{\mathbf{z}}; M)} \, g^{\mathrm{B}}_d(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0 \tag{60}$$

$$\Rightarrow \quad \int \Big( \sum_d \mu_d(\tilde{\mathbf{z}}; W) \, s_d(\tilde{\mathbf{z}}; M) \, g^{\mathrm{A}}_d(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \sum_d \, s_d(\tilde{\mathbf{z}}; M) \, g^{\mathrm{B}}_{d'}(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0 \tag{61}$$

$$\Rightarrow \quad \int \Big( \big( \boldsymbol{\mu}(\tilde{\mathbf{z}}; W) \, \odot \, \mathbf{s}(\tilde{\mathbf{z}}; M) \big)^{\mathrm{T}} \, \boldsymbol{g}^{\mathrm{A}}(\tilde{\mathbf{z}}) \; - \; \frac{1}{2} \, \big( \mathbf{s}(\tilde{\mathbf{z}}; M) \big)^{\mathrm{T}} \, \boldsymbol{g}^{\mathrm{B}}_d(\tilde{\mathbf{z}}) \Big) \, d\tilde{\mathbf{z}} \; = \; 0 \tag{62}$$

$$\Rightarrow \quad \int \begin{pmatrix} \boldsymbol{\mu}(\tilde{\mathbf{z}};W) \odot \mathbf{s}(\tilde{\mathbf{z}};M) \\ -\frac{1}{2}\,\mathbf{s}(\tilde{\mathbf{z}};M) \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \boldsymbol{g}^{\mathrm{A}}(\tilde{\mathbf{z}}) \\ \boldsymbol{g}^{\mathrm{B}}(\tilde{\mathbf{z}}) \end{pmatrix} d\tilde{\mathbf{z}} \;=\; 0 \tag{63}$$

$$\Rightarrow \quad \int \boldsymbol{\eta}^{\mathrm{T}}_{(\tilde{\mathbf{z}};\Theta)}\, \boldsymbol{g}(\tilde{\mathbf{z}})\, \mathrm{d}\tilde{\mathbf{z}} \;=\; 0\,, \tag{64}$$

where for the last step we have used the definition of $\boldsymbol{\eta}_{(\tilde{\mathbf{z}};\Theta)}$ in Eqn. (39).

To summarize, we have for the VAE derived from expression (45) the expression (64), i.e., we have shown that for the VAE generative model condition (42) holds.

For a generative model that satisfies the parameterization conditions (41) and (42) it holds at all stationary points (Theorem 1, Lücke, 2022) that

$$\mathcal{F}(\Phi,\Theta) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{H}[q_{\Phi}(\tilde{\mathbf{z}}|\mathbf{x}^{(n)})] - \mathcal{H}[p_A(\tilde{\mathbf{z}})] \;-\; \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{q_{\Phi}^{(n)}}\big\{\mathcal{H}[p_{\Theta}(\mathbf{x}\,|\,\tilde{\mathbf{z}})]\big\}\,. \tag{65}$$

The result notably applies even though we only used that derivatives w.r.t. $A$ and w.r.t. $\boldsymbol{\theta}$ vanish where $\boldsymbol{\theta}$ only contains the output layer parameters of the decoder network $\mathrm{DNN}_{\tilde{\sigma}}(\tilde{\mathbf{z}};M)$, see Eqn. (35). In other words, we only required

$$\forall h:\;\; \frac{\partial}{\partial\tilde{\alpha}_h}\,\mathcal{F}(\Phi,\Theta) \;=\; 0,\;\; \forall d,j:\;\; \frac{\partial}{\partial M^{L}_{dj}}\,\mathcal{F}(\Phi,\Theta) \;=\; 0,\;\; \text{and} \;\;\; \forall d:\;\; \frac{\partial}{\partial c^{L}_{d}}\,\mathcal{F}(\Phi,\Theta) \;=\; 0 \tag{66}$$

to show equality of the ELBO to expression (65). Additional information about vanishing derivatives for the remaining parameters is not required. Remaining parameters can therefore take on any value (including values that correspond to stationary points, of course). Hence, we can conclude that at all stationary points applies:

$$\begin{aligned}\mathcal{F}(\Phi,\Theta) &= \frac{1}{N}\sum_{n=1}^{N} \mathcal{H}[q_{\Phi}(\tilde{\mathbf{z}}|\mathbf{x}^{(n)})] - \mathcal{H}[p_A(\tilde{\mathbf{z}})] \;-\; \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{q_{\Phi}^{(n)}}\big\{\mathcal{H}[p_{\Theta}(\mathbf{x}\,|\,\tilde{\mathbf{z}})]\big\} \\ &= \frac{1}{N}\sum_{n=1}^{N} \frac{1}{2}\log\big(\det(2\pi e\tilde{\mathcal{T}}_{\Phi}(\mathbf{x}^{(n)})\big) - \frac{1}{2}\log\big(\det(2\pi e A)\big) - \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{q_{\Phi}^{(n)}}\big\{\sum_{d=1}^{D}\frac{1}{2}\log\big(2\pi\, e\, \tilde{\sigma}_d(\tilde{\mathbf{z}};M)\big)\big\}\,.\end{aligned} \tag{67}$$

As the last step, we transform the result back to the original VAE–3 parameterization, which we do along the same lines as was done in the proof of Theorem 2. Concretely, we again express $\tilde{\mathcal{T}}_{\Phi}(\mathbf{x})$ in terms of $\mathcal{T}_{\Phi}(\mathbf{x})$. We drop subscript and argument of $\tilde{\mathcal{T}}$ for readability, and note again that $\tilde{\mathcal{T}}$ is the covariance matrix of a Gaussian distribution defined in the space of $\tilde{\mathbf{z}}$. The random variable $\mathbf{z}$ depends according to the reparameterization Eqn. (32) on $\tilde{\mathbf{z}}$ via $\mathbf{z} = A^{-\frac{1}{2}}\tilde{\mathbf{z}}$. Consequently, if $\tilde{\mathbf{z}}$ is Gaussian distributed with covariance $\tilde{\mathcal{T}}$, then $\mathbf{z}$ 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. (65) we observe the first term to cancel with part of the last term:

$$\begin{aligned}\mathcal{F}(\Phi,\Theta) &= \frac{1}{N}\sum_{n=1}^{N} \frac{1}{2}\log\big(\det(2\pi e A\mathcal{T}_{\Phi}(\mathbf{x}^{(n)})\big) - \frac{1}{2}\log\big(\det(2\pi e A)\big) - \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{q_{\Phi}^{(n)}}\big\{\sum_{d=1}^{D}\frac{1}{2}\log\big(2\pi\, e\, \tilde{\sigma}_d(\tilde{\mathbf{z}};M)\big)\big\} \\ &= \frac{1}{N}\sum_{n=1}^{N} \frac{1}{2}\log\big(\det(2\pi e\mathcal{T}_{\Phi}(\mathbf{x}^{(n)})\big) - \frac{1}{2}\log\big(\det(2\pi e)\big) - \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{q_{\Phi}^{(n)}}\big\{\frac{1}{2}\log\Big(\det\big(2\pi\, e\, \Sigma_{\Theta}(\mathbf{z})\big)\Big)\big\} \\ &= \frac{1}{N}\sum_{n=1}^{N} \mathcal{H}[q_{\Phi}(\mathbf{z}|\mathbf{x}^{(n)})] - \mathcal{H}[p(\mathbf{z})] \;-\; \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{q_{\Phi}^{(n)}}\big\{\mathcal{H}[p_{\Theta}(\mathbf{x}\,|\,\mathbf{z})]\big\},\end{aligned} \tag{68}$$

which proofs the claim of Theorem 3. □

As a comment on the proof: at first it may not sound intuitive that we can only consider stationary points of the subset $\boldsymbol{\theta}$ instead of all parameters. Do note, though, that the stationary point condition serves to show that the integral of expression (44) vanishes. The integrand contains all model parameters $\Theta$ within the natural parameters $\boldsymbol{\eta}^{\mathrm{T}}_{(\tilde{\mathbf{z}};\Theta)}$. If we now just use a subset of parameters to construct the Jacobian of the left-hand-side expression of Eqn. (42), then such a subset is sufficient as long as we can conclude the right-hand-side of Eqn. (42). This is again in analogy to the explicit proof of Theorem 1 where the subset of parameters to show that $\mathcal{F}_2(\Phi, \Theta)$ becomes equal to an entropy is just consisting of the single parameter $\sigma^2$. In our case, we required all weights and biases of the output layer of $\mathrm{DNN}_\sigma(\mathbf{z}; M)$ as subset (but none of the DNN's other parameters). Choosing a smaller subset would not have been sufficient. Already using fixed biases $\mathbf{c}^L$ would break the proof (unless all biases are fixed to zero). Larger subsets $\boldsymbol{\theta}$ would just have made the proof more intricate without changing the final results.

*Author Contributions:* We remark that Theorem 1 to Theorem 3 were hypothesized by JL who provided the first versions of the proofs with significant contributions by SD.

## C ENTROPY-BASED ANALYSIS OF VAE OPTIMIZATION

### C.1 Differential Entropies and Typical Sets

At all stationary points the ELBO decomposes into three entropies which implies that all model parameters determine the ELBO value through the same mathematical object: entropy. This novel observation give rise to an entropy-based interpretation of VAE optimization.

Let us briefly recall a potential interpretation of (differential) entropies in terms of the typical set (e.g., MacKay, 2003; Cover and Thomas, 2006). Consider sequences of i.i.d. samples from a continuous probability distribution $p(\mathbf{x})$ over $\mathcal{X}$. The typical set of this distribution $p(\mathbf{x})$ in dependence of $N \in \mathbb{N}$ and $\epsilon > 0$ is then defined as

$$A_\epsilon^{(N)} = \Big\{ (\mathbf{x}_1, \ldots, \mathbf{x}_N) \in \mathcal{X}^N : \Big| -\frac{1}{N} \log(p(\mathbf{x}_1, \ldots, \mathbf{x}_N) - \mathcal{H}[p(\mathbf{x})] \Big| \leq \epsilon \Big\} \ . \tag{69}$$

Recall that we have $\mathbb{P}\big(A_\epsilon^{(N)}\big) > 1 - \epsilon$ for large enough $N$, i.e., most sequences that occur in practice will belong to the typical set with high probability. Moreover, the entropy of a distribution can be related to the volume of the typical set:

$$(1-\epsilon) 2^{N(\mathcal{H}[p(\mathbf{x})] - \epsilon)} \leq \mathrm{Vol}(A_\epsilon^{(N)}) \leq 2^{N(\mathcal{H}[p(\mathbf{x})] + \epsilon)} \ . \tag{70}$$

The second inequality holds for all $N$, and the first for sufficiently large $N$ (Cover and Thomas, 2006, Theorem 8.2.2). Thus, the volume of the typical set $A_\epsilon^{(N)}$ is characterized by the differential entropy (besides the growth in sequence length $N$). Considering the entropies of discrete random variables the volume of the typical set translates to its cardinality. To conclude, continuous random variables with small entropy are “confined to a small effective volume”, while random variables with high entropy are “widely dispersed”(Cover and Thomas, 2006).

With this interpretation in mind, Theorems 2 and 3 allow to re-interpret VAE learning. First, recall that ELBO maximization can be expressed in terms of reconstruction score $S_{\mathrm{rec}}$ and regularization score $S_{\mathrm{reg}}$, i.e., $\mathcal{F} = S_{\mathrm{rec}} + S_{\mathrm{reg}}$ (see Eqn. (2)). We are confronted with two conflicting goals here as with $S_{\mathrm{reg}}$ too large, i.e., $D_{\mathrm{KL}}\big(q_\Phi(\mathbf{z}|\mathbf{x}) \,\|\, p(\mathbf{z})\big)$ close to zero, no successful reconstruction (high $S_{\mathrm{rec}}$) can be expected. Given Theorems 2 and 3 we can now expressed both scores based on entropies as in Eqn. (23)

$$\begin{aligned} S_{\mathrm{reg}}(\Phi, \Theta) &= \frac{1}{N} \sum_n \mathcal{H}[q_\Phi(\mathbf{z}|\mathbf{x}^{(n)})] \; - \; \mathcal{H}[p_\Theta(\mathbf{z})] \ , \\ S_{\mathrm{rec}}(\Theta) &= -\, \mathcal{H}[p_\Theta(\mathbf{x}^{(n)} \,|\, \mathbf{z})] \end{aligned}$$

in which the entropy $\mathcal{H}[p_\Theta(\mathbf{x}^{(n)} \,|\, \mathbf{z})]$ is replaced by the expected entropy in the case of VAE–3.

Optimization of the decoder therefore seeks to *decrease* its entropy $\mathcal{H}[p_\Theta(\mathbf{x}|\mathbf{z})]$ (or its expectation in case of VAE–3), which corresponds to reducing the volume of the typical set, the effective volume of the decoding distribution in data space $\mathcal{X}$. As discussed in Section 4 the decoder entropy naturally resembles the reconstruction performance of the model (see also Eqn. (76) that makes this relationship explicit). Thus, a small volume of the decoder's typical set corresponds to a low reconstruction error/uncertainty (or equivalently a high reconstruction score $S_{\mathrm{rec}}(\Theta)$).

On the other hand, the average encoder entropy $\frac{1}{N}\sum_n \mathcal{H}[q_\Phi(\mathbf{z}|\mathbf{x}^{(n)})]$ is maximized. Hence, the (logarithm of the) volume of the typical set in latent space $\mathcal{Z}$ should be increased. Intuitively, this corresponds to a larger variety in the encoding distribution (per data sample $\mathbf{x}^{(n)}$). Note that the prior entropy $\mathcal{H}[p(\mathbf{z})]$ represents an upper bound on the average encoder entropy. Accordingly, the regularization score $S_{\mathrm{reg}}(\Phi, \Theta)$ captures (and minimizes) the difference between the (log-)volumes of typical sets of prior and encoder distribution. With too widely dispersed latent representations $\mathbf{z}$, i.e., when $\frac{1}{N}\sum_n \mathcal{H}[q_\Phi(\mathbf{z}|\mathbf{x}^{(n)})]$ is too close to $\mathcal{H}[p_\Theta(\mathbf{z})]$, no successful reconstruction can be demanded. Overall, VAE optimization translates to increase of encoder and decrease of decoder entropy, which in turn reflect the increase or decrease of the volumes of the corresponding typical sets. And notably, central figures for monitoring VAEs are easily accessible solely based on entropies (and even more reliably as their conventional counter-parts, as demonstrated in Section 4).

## C.2 Posterior Collapse

Theorems 2 and 3 represent by themselves theoretical results. Section 4 discussed some practical applicability in terms of entropy-based definitions for regularization and reconstruction scores as well as for posterior collapse. We here first derive and elaborate on the criterion (Eqn. (24)) used in Section 4 to measure the percentage of collapsed latents in practice. The accompanying experimental results are depicted in Fig. 7 in Appendix D.3. Our entropy-based results can, however, also be used to further study VAE optimization theoretically. Below we, therefore, secondly discuss optimization and turning points for learning that can be described using entropies, and we will finally come back to posterior collapse from this theoretical perspective.

Quantifying posterior collapse has been of interest previously (see, e.g., Bowman et al. (2016); He et al. (2018); Dieng et al. (2019)). The contribution by Lucas et al. (2019) concretely and systematically investigates the effect and its dependence on fixed decoder variance $\sigma^2$. The paper first argues for a quantification of posterior collapse in terms of percentage of collapsed latents, and then introduces an $(\epsilon, \delta)$-measure based on the KL-divergence (Lucas et al., 2019, p. 7) between encoder and prior distribution (for individual latents $h$). Values of the two thresholds $\epsilon$ and $\delta$ are hand-set, and a latent dimensions $h$ is defined to be collapsed whenever

$$\mathbb{P}_{\mathbf{x}\sim p_{\mathrm{data}}}\Big[D_{\mathrm{KL}}\big(q_\Phi^{\mathrm{enc}}(z_h|\mathbf{x}) \,\|\, p_\Theta^{\mathrm{prior}}(z_h)\big) < \epsilon\Big] \leq 1-\delta \ . \tag{71}$$

The measure is then used as a core VAE analysis tool monitoring individual latents or the fraction of collapsed latent dimensions.

Based on the convergence to entropies results, an alternative measure for posterior collapse offers itself if we consider the entropy-based regularization measure $S_{\mathrm{reg}}(\Phi,\Theta)$ of Eqn. (23). Posterior collapsed latents increase the regularization score while they usually have negligible effects on the reconstruction score. Because of the entropy-based formulation of the regularization score, it is now very straight-forward to break down the score into a sum over latents. Using the conventional assumption of Gaussian encoder and prior distributions both with diagonal covariance matrices, we obtain:

$$\begin{aligned}
S_{\mathrm{reg}}(\Phi,\Theta) &= \Big(\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[q_\Phi(\mathbf{z}|\mathbf{x}^{(n)})] \;-\; \mathcal{H}[p_\Theta(\mathbf{z})]\Big)\\
&= \Big(\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[\prod_{h=1}^{H} q_\Phi(z_h|\mathbf{x}^{(n)})] \;-\; \mathcal{H}[\prod_{h=1}^{H} p_\Theta(z_h)]\Big)\\
&= \sum_{h=1}^{H}\Big(\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[q_\Phi(z_h|\mathbf{x}^{(n)})] \;-\; \mathcal{H}[p_\Theta(z_h)]\Big)\\
&= \sum_{h=1}^{H}\Big(\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[q_\Phi(z_h|\mathbf{x}^{(n)})] \;-\; \frac{1}{2}\log(2\pi e)\Big) \ .
\end{aligned} \tag{72}$$

As $S_{\mathrm{reg}}(\Phi,\Theta)$ becomes equal to the conventional regularization measure

$$S_{\mathrm{reg}}^{\mathrm{conv}}(\Phi,\Theta) = -\frac{1}{N}\sum_n D_{\mathrm{KL}}\big(q_\Phi^{(n)}(\mathbf{z}) \,\|\, p_\Theta(\mathbf{z})\big) \tag{73}$$

at stationary points, $S_{\mathrm{reg}}(\Phi,\Theta)$ can at stationary points never be positive. It can, furthermore, be shown for VAE–1 that also each summand of Eqn. (72) (i.e., the regularization score per latent $h$) can never be positive at stationary points (we discuss further below). In turn, this means that there is a clearly defined highest possible value the encoder entropy (per latent variable $h$) can converge to, concretely

$$\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[q_\Phi(z_h|\mathbf{x}^{(n)})] \leq \frac{1}{2}\log(2\pi e) \approx 1.42 \ . \tag{74}$$

Values of the encoder entropy close to the prior entropy (for standard normal prior dimensions a value of $1.42$) can consequently be used to identify posterior collapsed latents, which motivates the definition presented in Eqn. (24).

Notably, we do not claim that no other sensible measures for posterior collapse can be defined. For instance, the measure (Eqn. (71)) represents a perfectly valid definition. Also the decomposition into sums over latents can be done using the original KL-divergence[4]. We would argue, however, that the previous measure (Eqn. (71)) is more ad hoc and requires two hand-set parameters $\epsilon$ and $\delta$ for the threshold. The measure (Eqn. (24)) may consequently be perceived as more natural, and may be a better starting point to define similar measures also for non-Gaussian VAEs or VAEs with learnable priors. Furthermore, the entropy-based definitions may be perceived as being easier to use, and the derivation (Eqn. (72)) may serve as an example.

We like to remark that we *do not* require the conditions of Theorems 2 and 3 to be fulfilled in order to apply the entropy-based criterion given in Eqn. (24). In fact, the definition is largely independent of the decoder architecture. In particular, the entropy-based measurement of posterior collapse is also valid and meaningful when the decoder (co-)variance is fixed, e.g., to $\sigma = 1$ in case of VAE–1, as commonly done in practice. See also the accompanying experiments in Appendix D.3 where we include this scenario (Fig. 7(c)).

Importantly, the entropy-based measures (e.g., for ELBO, regularization or reconstruction) are more than just reformulations of the conventional definitions. There are qualitative differences. The regularization score does, for instance, only depend on the encoder variances: it contains neither dependencies on decoder parameters (the priors only formally depend on $\Theta$) nor does it depend on the encoder means. The latter is in contrast to the KL-divergence, which does contain the encoder means (for example the measure in Lucas et al. (2019) given in Eqn. (71)). Similarly (and more drastically), the entropy-based reconstruction score for VAE–1 exclusively depends on one single decoder parameter, the decoder variance. In contrast, the conventionally defined decoder variance does depend on decoder and encoder parameters. In turn, this means that at stationary points dependencies on all parameters except of the decoder variance necessarily have to vanish for the conventional reconstruction score.

For completeness, the treatment of VAEs of type VAE–3 would be different but similar. For instance, the reconstruction score would (based on Theorem 3) now be defined as:

$$S_{\mathrm{rec}}(\Theta) = -\frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{q_{\Phi}^{(n)}}\big\{\mathcal{H}[p_{\Theta}(\mathbf{x}\,|\,\mathbf{z})]\big\}\,, \tag{75}$$

and consequently depends on encoder and decoder parameters. The definition of the regularization score remains unchanged, however.

## C.3 Optimization landscape

To elaborate on an application of the results for the theoretical analysis of VAE optimization, consider first VAEs of type VAE–1 (Def. 1). For such VAEs the closed-form bound of Theorem 2 applies at all stationary points. The result was notably derived using properties of two types of variance parameters: decoder variance and prior variance. Importantly, the prior variance is in the usual parametrization of VAEs not part of the prior but part of the decoder DNN, i.e., the $\alpha_h$ are part of the first DNN layer (see VAE–2 for an explicit encoding with prior variance). While Eqn. (14) of Theorem 2 does by itself not describe an optimization landscape, we can recover a description of an optimization landscape if we insert solutions for $\alpha_h^2$ for $\sigma^2$ into Eqn. (14). Solutions for $\alpha_h$ and $\sigma^2$ are given by the following explicit functions:

$$\alpha_h^2 = \frac{1}{N}\sum_{n=1}^{N}\int q_{\Phi}(\mathbf{z}|\mathbf{x}^{(n)})\, z_h^2\, d\mathbf{z}\,, \quad \sigma^2 \;=\; \frac{1}{DN}\sum_{n=1}^{N}\int q_{\Phi}(\mathbf{z}|\mathbf{x}^{(n)})\, \|\mathbf{x}^{(n)} - \boldsymbol{\mu}_{\Theta}(\mathbf{z})\|^2\, d\mathbf{z}, \tag{76}$$

and the $\alpha_h^2$ expression further simplifies for diagonal covariances. After the solutions for $\alpha_h$ and $\sigma^2$ are inserted into Eqn. (14), the resulting expression describes the optimization landscape within submanifolds of the parameter space (Fig. 4 shows an illustration). The description of the optimization landscape within submanifolds can be used to study the optimization landscape of the whole parameter space. Using this view, we further below relate back to the above discussed posterior collapse.

Similar approaches can be used for more general VAEs but an analysis necessarily becomes more intricate, e.g., because of the dependence of decoder variances on the latents for VAEs of type VAE–3 (but note the similar properties for $\alpha_h$).

[4] This is how one can show that encoder entropy of an individual latent is upper-bounded by $\frac{1}{2}\log(2\pi e)$.

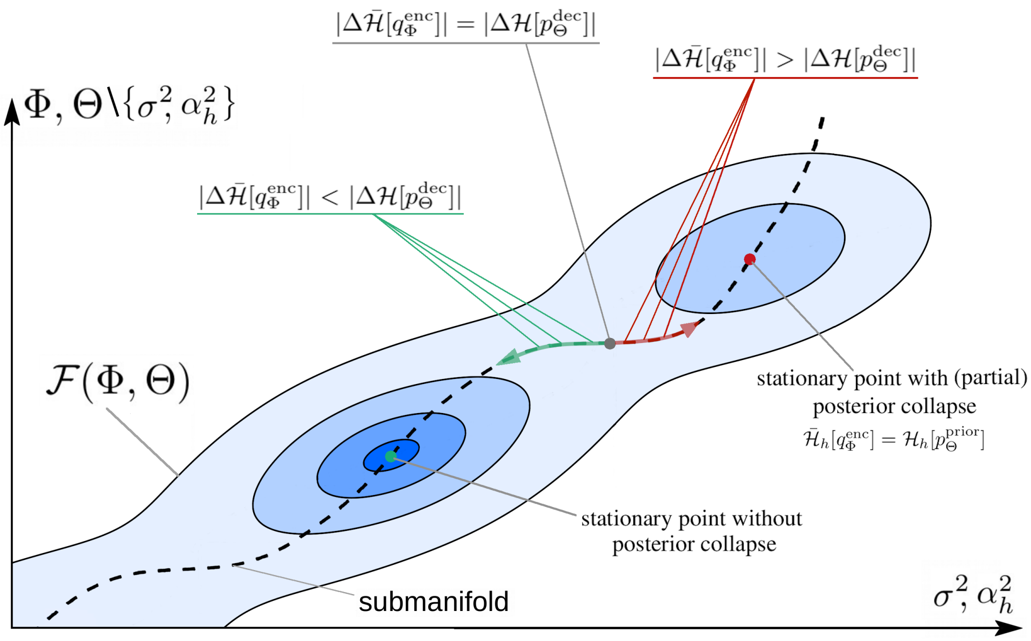

$|\Delta\bar{\mathcal{H}}[q_\Phi^{\mathrm{enc}}]| = |\Delta\mathcal{H}[p_\Theta^{\mathrm{dec}}]|$
$|\Delta\bar{\mathcal{H}}[q_\Phi^{\mathrm{enc}}]| > |\Delta\mathcal{H}[p_\Theta^{\mathrm{dec}}]|$
$\Phi, \Theta\backslash\{\sigma^2, \alpha_h^2\}$
$|\Delta\bar{\mathcal{H}}[q_\Phi^{\mathrm{enc}}]| < |\Delta\mathcal{H}[p_\Theta^{\mathrm{dec}}]|$
$\mathcal{F}(\Phi,\Theta)$
stationary point with (partial) posterior collapse
$\bar{\mathcal{H}}_h[q_\Phi^{\mathrm{enc}}] = \mathcal{H}_h[p_\Theta^{\mathrm{prior}}]$
stationary point without posterior collapse
submanifold
$\sigma^2, \alpha_h^2$

Figure 4: Visualization of the variational lower bound and its relation to the three entropies expression. The figure shows a two dimensional visualization with the following axes: the x-axis represents the hyperplane of variance parameters $\alpha_1^2$ to $\alpha_H^2$ together with decoder variance $\sigma^2$. We assume a VAE of type VAE–1 for the figure. The $\alpha_1^2$ to $\alpha_H^2$ we take to be implicitly defined by the decoder weights (compare VAE–2). The dotted black line represents a submanifold in which the parameters of the x-axis have converged. Within the submanifold, the variational lower bound is equal to a sum of three entropies (Theorem 2). In the illustrated example, the submanifold connects a stationary point without posterior collapse to a stationary point with (partial) posterior collapse. Depending on the location on the manifold, learning is dominated by the change in the reconstruction score $S_{\mathrm{rec}}(\Theta)$ (green arrow) or dominated by the change in the regularization score $S_{\mathrm{reg}}(\Phi,\Theta)$ (red arrow), with qualitatively different outcomes. As both scores can be defined based on entropies (see Eqn. (23)), changes of the scores directly translate to changes in entropies such that the optimization landscapes can be characterized using changes in entropies. Let us denote by $|\Delta\mathcal{H}[p_\Theta^{\mathrm{dec}}]|$ the absolute change of the decoder entropy, while we denote by $|\Delta\bar{\mathcal{H}}[q_\Phi^{\mathrm{enc}}]|$ the absolute average change of the encoder entropy, i.e., $\bar{\mathcal{H}}[q_\Phi^{\mathrm{enc}}] = \frac{1}{N}\sum_n \mathcal{H}[q_\Phi^{\mathrm{enc}}(\mathbf{z}|\mathbf{x}^{(n)})]$. If we start at the high local optimum and traverse the submanifold from left to right then the reconstruction score $S_{\mathrm{rec}}(\Theta)$ decreases while the regularization score $S_{\mathrm{reg}}(\Phi,\Theta)$ will tend to increase (to a lesser extent). Translated to entropies, we have within the submanifold between high maximum and saddle point $|\Delta\bar{\mathcal{H}}[q_\Phi^{\mathrm{enc}}]| < |\Delta\mathcal{H}[p_\Theta^{\mathrm{dec}}]|$, and ELBO optimization would favor reconstruction improvements. At the saddle point (the turning point), the changes of the entropies become equal. To the right of the turning point, learning is dominated by the regularization term, which means that the change in encoder entropy is dominant $|\Delta\bar{\mathcal{H}}[q_\Phi^{\mathrm{enc}}]| > |\Delta\mathcal{H}[p_\Theta^{\mathrm{dec}}]|$. ELBO optimization in this part of the manifold would result in (partial) posterior collapse. $|\Delta\bar{\mathcal{H}}[q_\Phi^{\mathrm{enc}}]| = |\Delta\mathcal{H}[p_\Theta^{\mathrm{dec}}]|$ At the local optimum with (partly) collapsed posterior, the encoder entropy, $\bar{\mathcal{H}}_h[q_\Phi^{\mathrm{enc}}]$, of a collapsed latent $h$ will be equal to the prior entropy.

## C.4 Theoretical Analysis of Posterior Collapse

To which extend posterior collapse is problematic (e.g., neglecting large parts of the models capacity) or not harmful (or even beneficial) is still discussed in the community (see, e.g., Asperti, 2020). It is important for our study to observe that the “mathematical mechanism of this phenomenon is not well understood” (as, e.g., stated by Lucas et al., 2019). Hence, the phenomenon requires additional analysis and this is where the contribution of the presented entropy perspective (provided by Theorems 2 and 3) can also go beyond the practical aspects of the introduced entropy-based measures. Concretely, the theorems allow a contribution to the question *why* some latent variables sometimes cease to participate in encoding/decoding and instead do assume a high variance.

Considering Theorem 2 and the entropy-based measures for regularization score and reconstruction score, we can investigate the optimization of the bound $\mathcal{F}(\Phi,\Theta)$ a bit more systematically. The bound is finally given by the sum of the two scores:

$$\mathcal{F}(\Phi,\Theta) = S_{\mathrm{reg}}(\Phi,\Theta) + S_{\mathrm{rec}}(\Theta) \tag{77}$$

By changing the parameters, there are essentially two ways to obtain a higher bound: (1) One can increase the reconstruction score $S_{\mathrm{rec}}(\Theta)$ without too much decreasing the regularization score $S_{\mathrm{reg}}(\Phi,\Theta)$. (2) One can (visa versa) obtain a large bound with an increased regularization score $S_{\mathrm{reg}}(\Phi,\Theta)$ as long as the reconstruction score is not decrease too much.

The considerations may (at first sight) only serve for comparisons of ELBO values at different stationary points. However, if we combine the expression (77) with our discussion of the optimization landscape above, we can furthermore obtain information about a continuous change of the bound within the above discussed submanifold of parameter space. By inserting[5] the expressions (76) into (Eqn. (77)), we obtain a description of the optimization landscape within a submanifold in parameter space (see dotted line of Fig. 4), i.e., within the submanifold the three entropy expression becomes identical to the original bound. The submanifold in the figure connects an optimum without posterior collapsed latents to an optimum with (partial) posterior collapse. When traversing from one optimum to the other along the submanifold, a saddle point will be reached. On the one side of the saddle point, gradient learning does increase the reconstruction score $S_{\mathrm{rec}}(\Theta)$ (as is usually desired for representation learning; green arrow in Fig. 4); on the other side of the saddle point, the regularization score $S_{\mathrm{reg}}(\Phi, \Theta)$ is increased (the posterior of the latent starts to collapse; red arrow in Fig. 4). Regularization increase of a single latent directly corresponds to that latent's increase to high entropy, removing the latent's influence on decoding is then a plausible consequence. Otherwise, the increasingly high entropy would increase the decoder entropy and, therefore, would negatively effect the bound.

The discussion of optimization landscapes adds a further example for the utility of entropy-based descriptions. In this case, changes of entropies $\Delta\mathcal{H}$ naturally characterize saddle points in the optimization landscape (see Fig. 4).

## D EXPERIMENTAL SET-UP AND RESULTS

We here provide further details on the numerical experiments of Section 4. An example implementation can be found at the end of this section (Appendix D.7). The code to reproduce the experiments is available at github.com/Learning-with-Entropies.

This section is organized as follows. In Appendix D.1 we elaborate on the verification of the main result itself, invoking Linear VAE, VAE–1 and VAE–3 (i.e., Figs. 1 and 5). We continue with two proposed applications of the entropy results for non-linear VAEs: The entropy-based estimation of the ELBO in Appendix D.2 and the results of the entropy-based posterior collapse analysis in Appendix D.3. We then provide details on the experimental setup including the data sets considered as well as the used network architectures in Appendixes D.4 and D.5. Lastly, we provide further insight on the stochasticity and small remaining gaps between ELBO and the sum of entropies in Appendix D.6.

### D.1 Verification (Figs. 1 and 5)

**Linear VAEs** The linear VAE given by Eqns. (18) and (19) allows for the most direct investigation of the result of Theorem 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 D.5). In Fig. 1(a), top plot, we then compare the three entropies of Eqn. (20) with the standard lower bound (Eqn. (2)), estimated by sampling, and the known exact log-likelihood solution (Appendix A, Eqn. (25)) 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 Theorem 2, the lower bound (Eqn. (2)) and the three entropies (Eqn. (20)) only have to be identical at stationary points of the variance parameters. This suggests that e.g., for fixed $\sigma^2$ 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, which shows two examples with $\sigma^2$ fixed to two sub-optimal values.

**Standard VAEs** We demonstrate the validity of the theoretical result on MNIST, CelebA and SUSY using VAE–1 (see Fig. 1 in the paper itself and Fig. 5 for the latter two). For verification of Theorem 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 (details in Appendix D.5). Since no ground-truth log-likelihood is available for these models, we only show the standard lower bound (2), estimated by sampling, and compare it to the three entropies (14) and (22), respectively. More details and visualizations are provided at the end of this section. Additionally, we considered more practically relevant and large data sets, namely SUSY and CelebA.

[5] A VAE–1 can be regarded as being reparameterized as VAE–2 (see Theorem 2). In the conventional VAE–1 form, the converged $\alpha_h$ can be taken as implicit in the decoder weights.

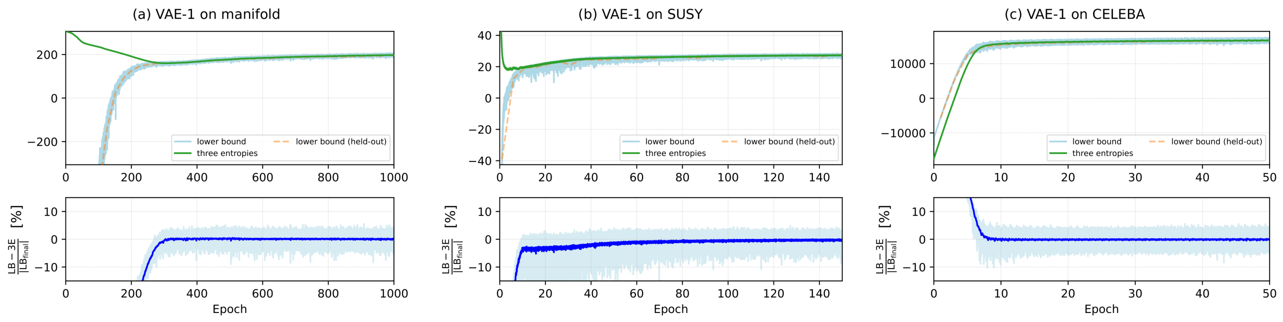


Figure 5: **Additional (verification) experiments** on artificial manifold data, SUSY and CelebA with VAE–1 models. Depicted are the lower bound (ELBO), the lower bound on held-out-test data and the sum of entropies. The lower plot depicts the relative difference with an exponential moving average in dark blue. We remark the the ELBO fluctuates around the three entropies expression. See Appendix D.4 for further details.

**Experimental Results: Verification** As can be observed, the sampled lower bound converges towards the sum of three entropies when the variance parameters are converging towards stationary points (which usually happens within the first epochs). In Fig. 1(a), top, we also see that both the lower bound and the three entropies converge to the log-likelihood (with the tight lower bound mostly invisible under the log-likelihood) and recover the ground-truth well. Regarding Fig. 5, we used the held out computation of the lower-bound to ensure the absence of over-fitting.

Notably, 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 the gradients of the variances have to vanish for the results to hold. Consequently, we observe high approximation qualities also distant from convergence points including, e.g., at saddle points or saddle surfaces, which commonly occur for DNN optimization.

The bottom plots in Fig. 1 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. Remaining gaps we can account primarily to fluctuations caused by finite learning rates, batch sizes and numbers of samples. E.g., for VAE–3, $\sigma_d(\mathbf{z};\Theta)$ 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 $\sigma_d(\mathbf{z};\Theta)$, which makes the three entropies fluctuate around the lower bound (see also Fig. 10).

There are some subtleties to pay attention to in the case of practical experiments for VAE–3. We have assumed non-degenerated values for the decoder variance $\mathrm{DNN}_\sigma(\mathbf{z};M)$ (Eqn. (21)) at stationary points, for instance. However, during learning and due to stochasticity and finite learning rates, close to zero or negative values for $\tilde{\sigma}_d(\mathbf{z};\Theta)$ can and do sometimes occur during learning. To explicitly exclude such degeneracies in transient behavior, non-negativity can be ensured using constraints. In practice, we, for instance, used ReLU units with a small offset for the output layer or softplus activation. For our numerical results near stationary points, we then made sure that the activation functions of observables do operate in a linear regime after learning has converged, which is then again consistent with the conditions assumed for the variance DNNs (Eqn. (21)). Concretely, we measured in experiments the percentage of variance DNN output units where enforcing non-negativity remained necessary. When the variance DNNs converged to stationary points, the percentage of non-linear units converged to zero.

**Experimental Details of Fig. 1** Regarding these experiments we used a batch size of 2000, learning rates of $10^{-3}$ and 100 samples (with the exception of Fig. 9(b) and (c), 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.

## D.2 Entropy-based ELBO estimation

While the verification experiments discussed in the former section (regarding Fig. 1) focus on demonstrating the theoretical result itself, thereby trying to reduce the noise-level (see also Appendix D.6), Fig. 5 demonstrates a more practical perspective as outlined in Section 4: For the prominent VAE–1 the lower bound fluctuates around the sum of entropies after convergence. Thus, a first handy utility is straight-forward entropy-based computation of ELBO values, which are routinely used to monitor and compare models on the training set.

That is, our theoretical results allow for using *analytical* expressions (Theorem 2) instead of numerical approximations of the analytical integrals of the original objective (Eqn. (2)). Please also see the discussion in Section 4 in the main paper. In our experiments, we observed convergence of the variance parameters after several epochs (depending on architecture, data set and learning rate), and thus to the point at which the conditions for obtaining our closed-form expression are (approximately) fulfilled. To demonstrate the advantage of the sum of entropies, we compared it to the naïve approximation of the ELBO for different number of samples (Figs. 2 and 6) using Monte-Carlo sampling. The three entropies expression can provide very precise estimates of the training-ELBO with very few samples already.

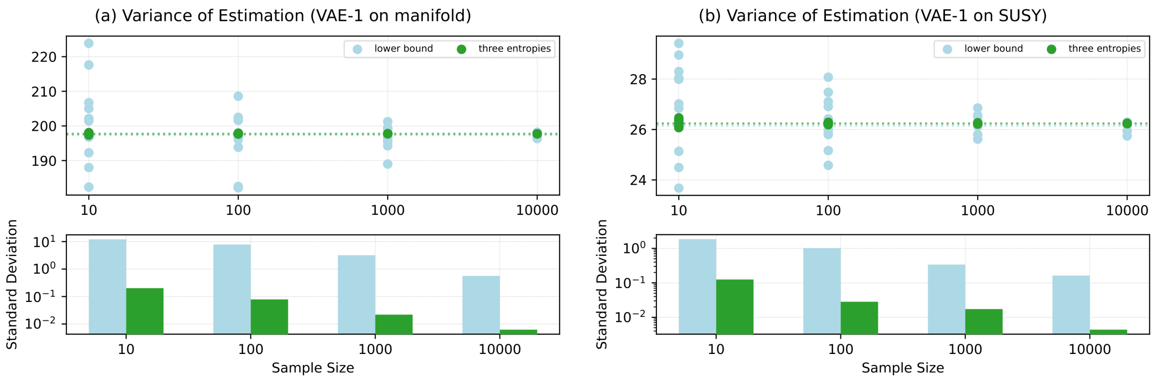


Figure 6: **ELBO Estimation** for VAE–1 models on (a) artificial manifold data set and (b) SUSY data set. Approximation of ELBO and sum of entropies for the fully trained model with different sample sizes, repeated 10 times. Mean of the estimates with 10k samples is visualized as a dotted horizontal line. Considering the three entropies expression, the variance in the estimates is barely visible in the scatter plot (upper plot). Note, that the standard deviation (lower plot) is displayed on logarithmic scale. See Section 4 for the accompanying discussion and Appendix D.4 for details on the experimental setup.

We remark that this effect is slightly stronger for high-dimensional data. Also note that the decoder entropy of a trained VAE–1 model is a property of the model and not data-dependent. It follows that the decoder entropy clearly should not be confused with a reconstruction measure on unseen data.

## D.3 Entropy-based Posterior Collapse Detection.

As outlined in the main part of this paper, the entropy perspective paves the way to elegantly define and analyze posterior collapse based on entropies. The criterion, given in Eqn. (24), states that a latent variable $z_h$ is $\delta$-collapsed, if the encoder entropy matches the corresponding prior entropy too closely, i.e., $\frac{1}{N}\sum_{n=1}^{N}\mathcal{H}[q_\Phi(z_h|\mathbf{x}^{(n)})] > \mathcal{H}[p(z_h)] - \delta$. For the derivation and additional discussion see Appendix C.3. The results are presented in Fig. 7 to which we here add the experiment-specific details. **Artificial Data.** We used $H = 20$ latents while the true dimensionality of the artificial manifold data set is 9. Thus, 8 non-collapsed latent dimensions are reasonable. **SUSY.** Again, we have knowledge of the manifold dimensionality on which the data is concentrated. In this case it is 8 as all remaining features are functions of the first 8 features. In the considered experiment the intrinsic dimensionality of the data can be successfully recovered. **CelebA.** The manifold hypothesis seems to be valid for this data set as well, however we cannot determine the dimensionality of the data manifold. Notably, we do not observe posterior collapse for the full VAE–1 which we relate to the dimension-depending scaling of encoder/prior entropy (or equivalently the KL divergence, which scale with latent dimension $H$) and decoder entropy (scaling with data dimension $D$). Clearly, this is an instance of under-regularization. In contrast, when restricting $\sigma^2 = 1$ we observe severe posterior collapse (Fig. 7(c)). This relates to the observation that fixing $\sigma^2$ to some sub-optimal value is related to the re-balancing issue of KL divergence and reconstruction term (which is also addressed by Higgins et al. (2017)). Here, enforcing $\sigma^2 = 1$ increases the weight of the KL-divergence $S_{\mathrm{reg}}$ (relative to reconstruction score $S_{\mathrm{rec}}$) such that many latent variables are pruned out. As noted in the main text, posterior collapse is not an issue *per se* but needs to be investigated carefully with respect to the (assumed) intrinsic dimensionality of the data in order not to over- or under-regularize. For further discussion see Section 4 and for details of the experimental set-up the Appendix D.4.

An interesting finding from these experiments is that the relative dimensionality ($D$ vs. $H$) contributes in preventing posterior collapse in VAE optimization (provided that, in particular, the decoder variance is learnable). With $D \gg H$ as on CelebA we did not find a single collapsed latent variable ($D = 64^2 \times 3 = 12,288$ vs. $H = 128$). We plan to investigate these findings in future.

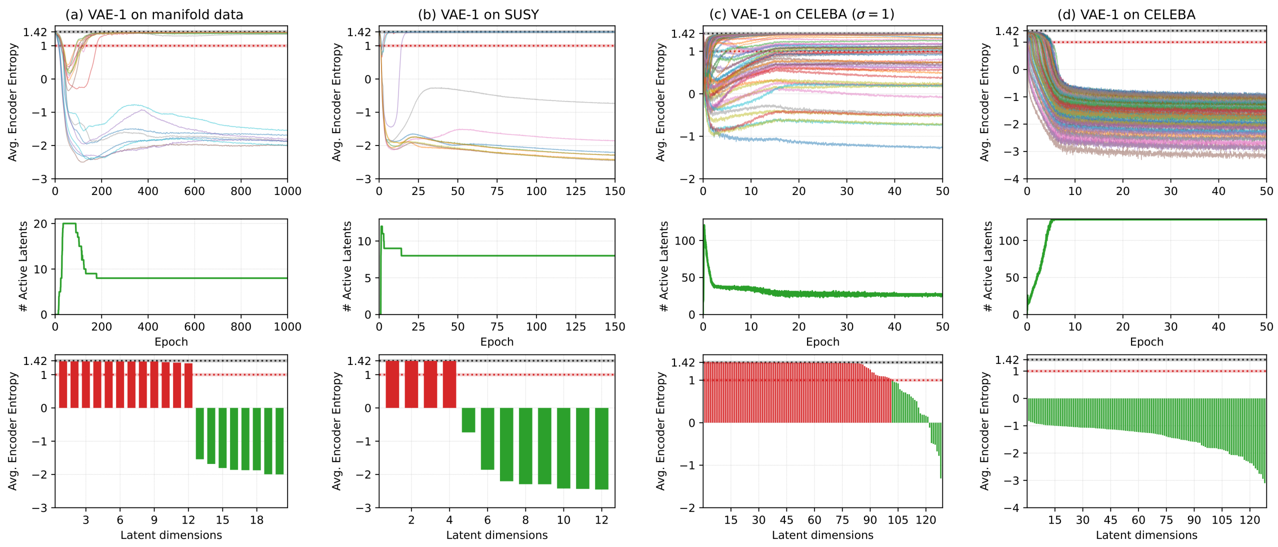


Figure 7: **Posterior collapse** analysis for VAE–1 models on different data sets based on the entropy criterion (Eqn. (24)). For the threshold we used $\frac{1}{N}\sum_n \mathcal{H}[q_\Phi(z_h|\mathbf{x}^{(n)})] > 1$. *Top plots:* Average encoder entropies $\frac{1}{N}\sum_n \mathcal{H}[q_\Phi(z_h|\mathbf{x}^{(n)})]$ over the course of training with upper bound $\mathcal{H}[p(z_h)] \approx 1.42$ and threshold of one. *Middle plots:* Number of non-collapsed latent dimensions over the course of training based on the entropy criterion. *Bottom plots:* Average encoder entropies of the fully trained model per latent in descending order. Collapsed latent dimensions are colored in red. Note that in (c) we consider a VAE–1 with *fixed* decoder variance which in turn leads to posterior collapse. Encoder entropies are calculated on ten batches for the trained models. See Section 4 and Appendix C.2 for details on posterior collapse and further discussion and Appendix D for the experimental set-up.

## D.4 Experimental Set-up and Implementation Details

We here provide the specifications of the considered experiments on SUSY, CelebA and artificial manifold data.

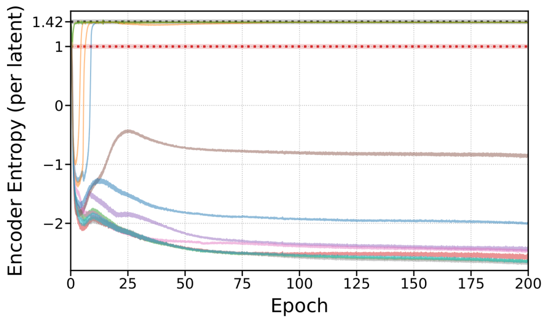


Figure 8: **Additional Posterior Collapse** experiment with VAE–1 on the SUSY data set. We altered the latent dimension, here to $H = 13$, and observed reliable recovery of the intrinsic data dimensionality across different random seeds.

**Experiments on CelebA (VAE–1)** In order to demonstrate the theoretical result on a more complex architecture we used the publicly available VAE–1 implementation of Subramanian (2020) that involves convolutional layers and batch normalization. As this implementation (like many other) use a fixed decoder variance $\sigma^2$ (and then rely on rescaling the KL-Term in the ELBO), the only architectural change we made is to turn $\sigma^2$ into a learnable parameter which we then train end-to-end with the rest of the model. For the experiments on posterior collapse (Fig. 7) we trained another model with fixed decoder variance $\sigma^2 = 1$ and simple linear KL-annealing schedule from $\beta = 0.1$ to $\beta = 1.0$ over the first 15 epochs. We resized the data set such that the data dimension amounts to $D = 64 \times 64 \times 3 = 12,288$ and set the latent dimension to $H = 128$. We used a batch size of 256, a train-test-split of $60\%/40\%$, and optimized for 50 epochs with ADAM (using default learning rate of $10^{-3}$). We employed data augmentation using random horizontal flips.

**Experiments on SUSY (VAE–1)** SUSY is a machine learning data set from the field of high-energy physics and consists of 5 million measurements of processes that potentially produce super-symmetric particles (Baldi et al., 2014). We decided to include experiments on this real-world dataset to illustrate the phenomenon of posterior collapse: Despite having data dimensionality $D = 18$, only the first 8 features are measurements of kinetic properties, while the remaining 10 are functions of the first 8 features.[6] Thus, we have knowledge about the dimensionality of the manifold on which the data is located and can therefore assess posterior collapse on a solid basis.[7]

[6]For further description see the corresponding website `https://archive.ics.uci.edu/ml/datasets/SUSY` and Baldi et al. (2014).

[7]On usual high-dimensional data sets (thinking of image data sets like CIFAR or CelebA) we cannot discern the dimensionality of the manifold on which the data is concentrated.

We used a VAE of type VAE–1 implemented as a simple two-layer perceptron with $128$ and $24$ hidden neurons and Leaky ReLU as activation function for the encoder and in reversed order for the decoder. We use $H = 12$ latent dimensions and optimized with ADAM with the standard initial learning rate of $10^{-3}$ for (at least) $150$ epochs with a batch size of $1024$. We reduced the data set size to $2.5$ million samples, used a train-test-split of $50\%/50\%$ and scaled the features to range $[0, 1]$ before training.

In order to learn a proper embedding in the latent space first (and avoid over-regularization in early training stages) we initialized $\ln(\sigma^2_{\text{init}}) = -9$. This small value puts high weight on reducing the reconstruction term first. During training $\sigma^2$ becomes optimal which restores the balance between regularization and reconstruction. We found this simple trick helpful to avoid widespread posterior collapse in the first few epochs for this data set.

**Experiments on artificial manifold data set (VAE–1)** Besides the real world data sets described above we want to test our ideas regarding posterior collapse on a data set where we have explicit control over the intrinsic dimensionality of the data. For this reason we created a $D = 100$ dimensional data set for which we can control the dimensionality of the lower-dimensional manifold on which the data is concentrated (just as in the case of SUSY). In the experiments presented here we use $k = 9$ dimensions which we initialized with Gaussian noise. The remaining dimensions are then created using non-linear functions applied to (a subset) of the first $k$ dimensions. For the experiments presented here, we simply applied the cosine function to a random linear combination of a pair of the first $k$ dimensions (squared). E.g., dimension $p \in [10, 100]$ is simply (element-wise) $\mathbf{x}_p = \cos(\mathbf{x}_r^2 + \mathbf{x}_s^2)$ for $r, s$ randomly drawn among the first $k = 9$ dimensions (with replacement). We added small Gaussian noise to this manifold and resized all features back to range $[0, 1]$.

The experiments are conducted on $50.000$ samples from this data set with a split of $50\%$ training set and $50\%$ test set. We set the latent dimension to $H = 20$. Again we used simple two-layer perceptrons with $128$ and $40$ hidden neurons with Leaky ReLU for the encoder, and in reversed order for the decoder. For the same reason as in the SUSY experiments, we initialized $\ln(\sigma^2_{\text{init}}) = -9$. Again, we used ADAM with the standard initial learning rate of $10^{-3}$ and a batch size of $512$.

### D.5 Generation of Artificial PCA Data Sets and Learning Visualizations

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

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbb{I}) \tag{78}$$

$$p(\mathbf{x} \,|\, \mathbf{z}) = \mathcal{N}(\mathbf{x}; W_{\text{gen.}}\mathbf{z} + \mu_{\text{gen.}}; \sigma^2_{\text{gen.}}\mathbb{I}). \tag{79}$$

The generative parameters $W_{\text{gen.}}$ and $\mu_{\text{gen.}}$ 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 $\mathbf{x}' = g(\mathbf{x})$ with

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbb{I}) \tag{80}$$

$$p(\mathbf{x} \,|\, \mathbf{z}) = \mathcal{N}(\mathbf{x}; W_{\text{gen.}}\mathbf{z}; \sigma^2_{\text{gen.}}\mathbb{I}) \tag{81}$$

$$\mathbf{x}' = \mu_{\text{gen.}} + \frac{\mathbf{x}}{10} + \frac{\mathbf{x}}{|\mathbf{x}|} \tag{82}$$

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

Fig. 10(c), on the last page, shows a PCA projection of the training data set, visualizations of the $\mathbf{z}-$ and $\mathbf{x}$-space during training of VAE–3 as well as plots of the lower bound and the three entropies. Figs. 10(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 value slightly above of the ground-truth. However, even with early stopping at around 700 iterations (see third segment of Fig. 10(a), we see that the three entropies already compute the lower bound very well.

### D.6 Noise in Three Entropies and ELBO

The optimization of VAEs is stochastic due to finite learning rates, finite batch sizes and approximations of integrals over $\mathbf{z}$ 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. (22) of Theorem 3. As can be observed, and as stated in the main text, the values for the standard variational bound (Eqns. (2) and (22)) 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. (22).

We can also further study the effect of stochasticity by using smaller batch sizes (Fig. 9(b) or higher learning rates (Fig. 9c). 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. 9, 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. 9, bottom row). However, for completeness, we also provide the relative deviation which is (as expected) smaller than the error (Fig. 9, bottom row). This means that the difference between variational lower bound and Eqn. (22) 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 Theorem 3 (and the other Theorems), and (B) they show that the three entropies results of Theorems 2 and 3 can provide very accurate estimations of the variational bound in practice. As demonstrated in Figs. 2 and 6 the three entropy expression is even able to significantly reduce the variance in ELBO estimation for trained VAE–1 models. 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 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 9 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 Theorem 2) which in turn enable us to reduce the variance in ELBO estimation by at least a factor of 10 (Figs. 2 and 6 and the accompanying discussion).

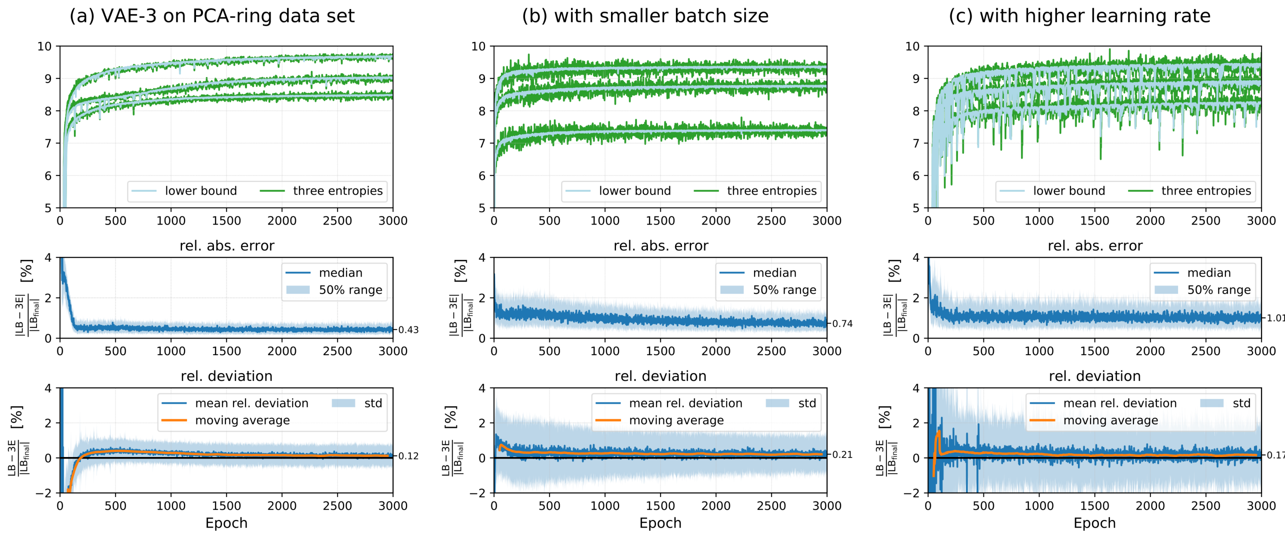


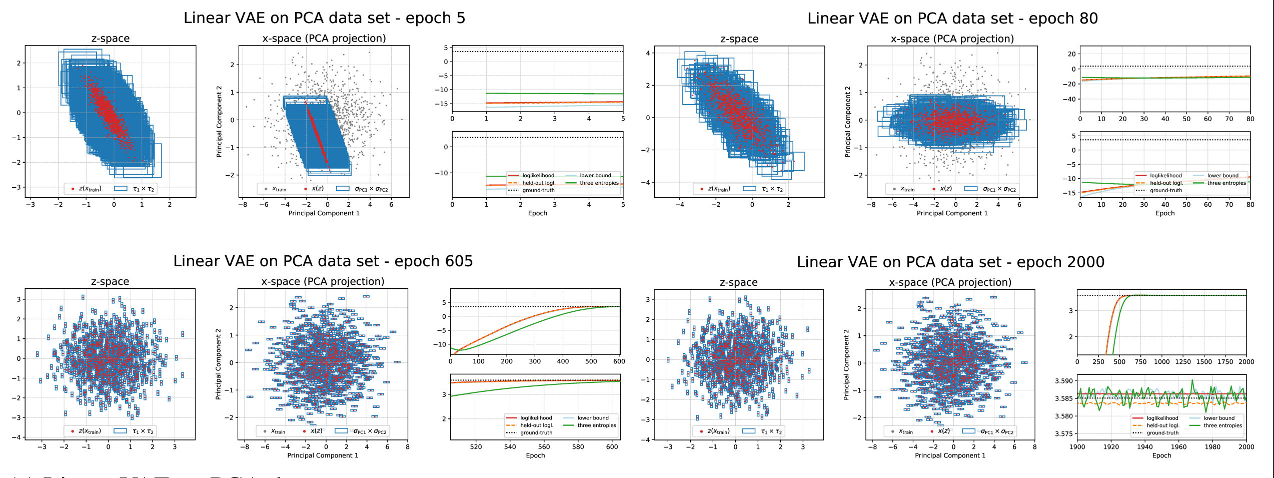
Figure 9: VAE–3 on PCA-ring data sets. The top two plots show experiments in the same way as Fig. 1: The top plot shows three independent runs on new randomly generated PCA-ring data each, while the middle plot shows the median of the relative absolute error between lower bound and three entropies over 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).

## D.7 Example Implementation

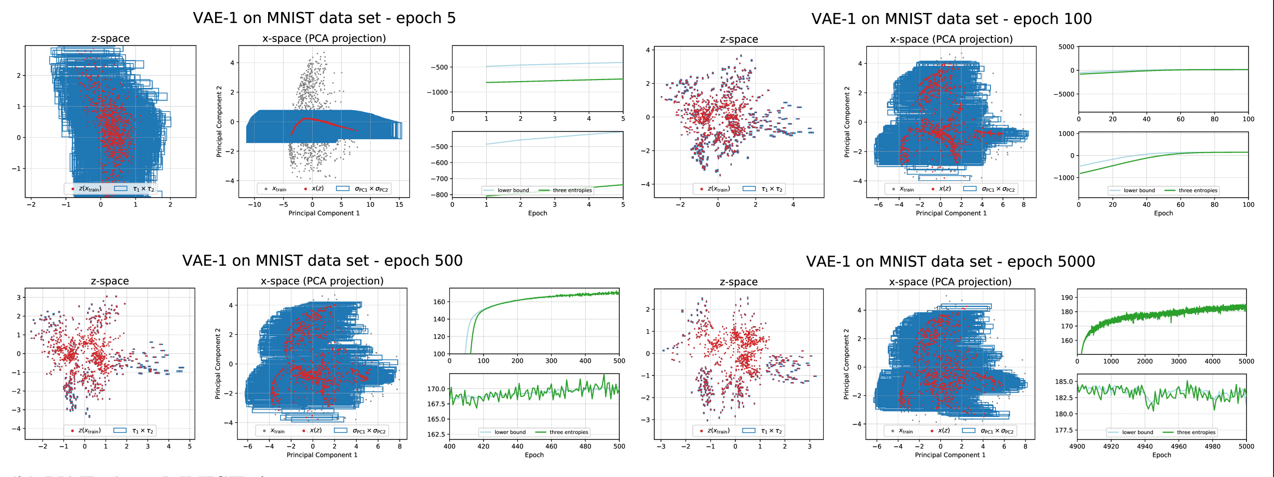
Listing 1 shows an example `PyTorch` implementation of the linear VAE as well as functions to compute the three entropies for the linear VAE, as well as VAE–1 and VAE–3, either directly or based on `torch.distributions`. For the full code see the link provided in the beginning of Appendix D.

Listing 1: Example Implementations

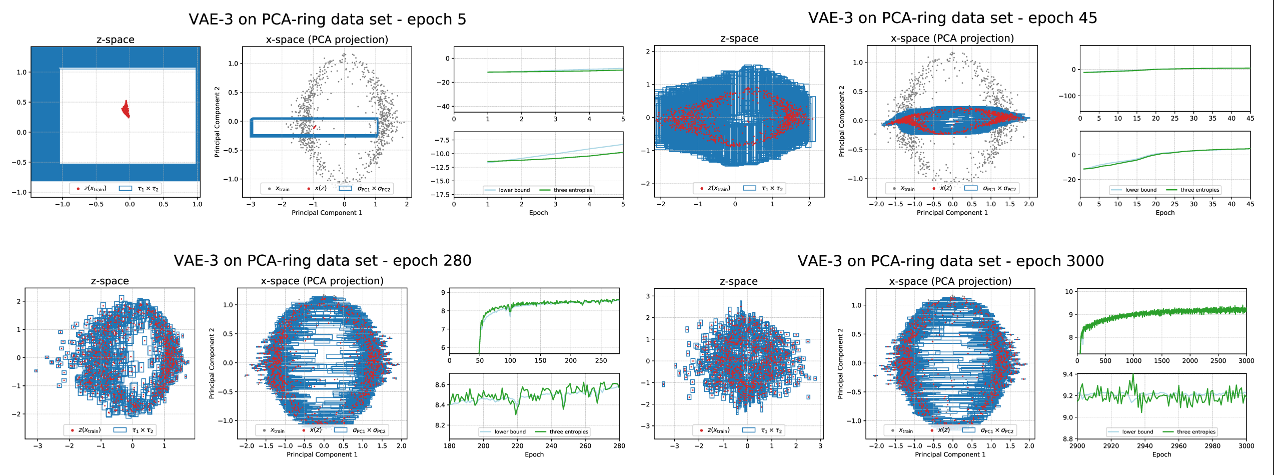
```python
import torch
from torch.nn as import Linear, Module, Parameter
from torch.distributions import Normal, kl_divergence
import numpy as np


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, see Corollary 1.
        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), see Theorem 2.
        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), see Theorem 3.
        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()


def ELBO_three_entropies(vae, x, x_recon, enc_mu, enc_logvar, dec_logvar,
                         device):
    """
    Calculate ELBO and the sum of three entropies (H_sum); see Theorem 2 & 3
    :param vae: VAE instance (either VAE-1, i.e., dec_logar = log(sigma^2)
                              or VAE-3, i.e., dec_logar = DNN(z))
    :param x: input data (usually a batch)
    :param x_recon: reconstructed data (by VAE)
    :param enc_mu: Encoder means
    :param enc_logvar: Encoder log-variances
    :param dec_logvar: Decoder log-variance(s)
    :return: ELBO, H_sum
    """
    # Encoder Distribution q(z|x)
    q_z_x = Normal(enc_mu, (0.5 * enc_logvar).exp())

    # Decoder Distribution p(x|z) with learned variance
    p_x_z = Normal(x, (0.5 * dec_logvar).exp())

    # Prior Distribution p(z) (standard normal prior)
    p_z = Normal(torch.zeros(vae.H).to(device), torch.ones(vae.H).to(device))

    # Calculate the ELBO (mean of batch, sum over data dims d/latent dims h)
    log_prob = p_x_z.log_prob(x_recon).mean(0).sum()
    kl_div = kl_divergence(q_z_x, p_z).mean(0).sum()

    ELBO = log_prob - kl_div

    # Calculate the Three Entropies
    H_prior = p_z.entropy().sum() # sum over latents
    H_enc = q_z_x.entropy().mean(0).sum() # mean of batch, sum over latents
    H_dec = p_x_z.entropy().mean(0).sum() # mean of batch, sum over data dims

    H_sum = - H_dec - H_prior + H_enc

    return ELBO, H_sum
```



(a) Linear VAE on PCA data set



(b) VAE–1 on MNIST data set



(c) VAE–3 on PCA-ring data set

Figure 10: 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 $\mathbf{z}$-samples from the encoder in the 2-dimensional z-space with the encoder standard deviation $\tau_h$ 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 $\mathbf{x}(\mathbf{z})$-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 $2\sigma_{\text{PC1}}$ and height $2\sigma_{\text{PC2}}$. 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.