Variational Autoencoders Pursue PCA Directions (by Accident)

Michal Rolinek*, Dominik Zietlow* and Georg Martius
Autonomous Learning Group, Max-Planck-Institute for Intelligent Systems
Tübingen, Germany
{mrolinek, dzietlow, gmartius}@tue.mpg.de

December 18, 2018

Abstract

The Variational Autoencoder (VAE) is a powerful architecture capable of representation learning and generative modeling. When it comes to learning interpretable (disentangled) representations, VAE and its variants show unparalleled performance. However, the reasons for this are unclear, since a very particular alignment of the latent embedding is needed but the design of the VAE does not encourage it in any explicit way. We address this matter and offer the following explanation: the diagonal approximation in the encoder together with the inherent stochasticity force local orthogonality of the decoder. The local behavior of promoting both reconstruction and orthogonality matches closely how the PCA embedding is chosen. Alongside providing an intuitive understanding, we justify the statement with full theoretical analysis as well as with experiments.

1 Introduction

The Variational Autoencoder (VAE) [23, 32] is one of the foundational architectures in modern-day deep learning. It serves both as a generative model as well as a representation learning technique. The generative model is predominantely exploited in computer vision [24, 14, 21, 15] with notable exceptions such as generating combinatorial graphs [25]. As for representation learning, there is a variety of applications, ranging over image interpolation [18], one-shot generalization [31], language models [39], speech transformation [4], and more. Aside from direct applications, VAEs embody the success of variational methods in deep learning and have inspired a wide range of ongoing research [22, 40].

Recently, unsupervised learning of interpretable latent representations has received a lot of attention. Interpretability of the latent code is an intuitively clear concept. For instance, when representing faces one latent variable would solely correspond to the gender of the person, another to skin tone, yet another to hair color and so forth. Once such a representation is found it allows for interpretable latent code manipulation, which is desirable in a variety of applications; recently, for example, in reinforcement learning [35, 17, 11, 37, 30].

The term *disentanglement* [10, 3] offers a more formal approach. A representation is considered disentangled if each latent component encodes precisely one "aspect" (a generative factor) of the data. Under the current disentanglement metrics [16, 20, 7], VAE-based architectures (β-VAE [16], TCVAE [7], FactorVAE [20]) dominate the benchmarks, leaving behind other approaches such as InfoGAN [8] and DCIGN [24].

The success of VAE-based architectures on disentanglement tasks comes with a certain surprise. One surprising aspect is that VAEs have been challenged on both of its own design functionalities, as generative models [13] and as log-likelihood optimizers [27, 29]. Yet, no such claims are made in terms of disentanglement. Another surprise stems from the fact that disentanglement requires the following feature: the representative low-dimensional manifold must be aligned well with the coordinate axes. However, the design of the VAE does not suggest any such

*These authors contributed equally to this work.

mechanism. On the contrary, the idealized log-likelihood objective is, for example, invariant to rotational changes in the alignment.

Such observations have planted a suspicion that the inner workings of the VAE are not sufficiently understood. The recent works on the subject made intriguing empirical observations [6, 2], gave a fresh theoretical analysis [9], and raised pressing questions [1]. However, a mechanistic explanation for the VAE's unexpected ability to disentangle is still missing.

In this paper, we isolate an internal mechanism of the VAE (also β-VAE) responsible for choosing a particular latent representation and its alignment. We give theoretical analysis covering also the nonlinear case and explain the discovered dynamics intuitively. We show that this mechanism promotes local orthogonality of the embedding transformation and clarify how this orthogonality corresponds to good disentanglement. Further, we uncover strong resemblance between this mechanism and the classical Principle Components Analysis (PCA) algorithm. We confirm our theoretical findings in experiments.

Our theoretical approach is particular in the following ways: (a) we base the analysis on the *implemented* loss function in contrast to the typically considered idealized loss, and (b) we identify a specific regime, prevalent in practice, and utilize it for a crucial simplification. This simplification is the crucial step in enabling formalization.

The results, other than being significant on their own, also provide a solid explanation of "why β-VAEs disentangle".

2 Background

Let us begin with reviewing the basics of VAE, PCA, and of the Singular Value Decomposition (SVD), along with a more detailed overview of disentanglement.

2.1 Variational Autoencoders

Let $\{\mathbf{x}^i\}_{i=1}^N$ be a dataset consisting of N i.i.d. samples $\mathbf{x}^i \in X = \mathbb{R}^n$ of a random variable $\mathbf{x}$. An autoencoder framework operates with two mappings, the encoder $\mathrm{Enc}_\varphi \colon X \to Z$ and the decoder $\mathrm{Dec}_\theta \colon Z \to X$, where $Z = \mathbb{R}^d$ is called the *latent space*. In case of the VAE, both mappings are probabilistic and a fixed *prior distribution* $p(\mathbf{z})$ over Z is assumed. Since the distribution of $\mathbf{x}$ is also fixed (actual data distribution $q(\mathbf{x})$), the mappings Enc_φ and Dec_θ induce joint distributions $q(\mathbf{x}, \mathbf{z}) = q_\varphi(\mathbf{z}|\mathbf{x})q(\mathbf{x})$ and $p(x, z) = p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$, respectively (omitting the dependencies on parameters θ and φ). The idealized VAE objective is then the marginalized log-likelihood

$$\sum_{i=1}^{N} \log p(\mathbf{x}^i). \tag{1}$$

This objective is, however, not tractable and is approximated by the evidence lower bound (ELBO) [23]. For a fixed $\mathbf{x}^i$ the log-likelihood $\log p(\mathbf{x}^i)$ is lower bounded by

$$\mathop{\mathbb{E}}_{\mathbf{z}\sim q(\mathbf{z}|\mathbf{x}^i)} \log p(\mathbf{x}^i \mid \mathbf{z}) - D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x}^i) \parallel p(\mathbf{z}))\,, \tag{2}$$

where the first term corresponds to the reconstruction loss and the second to the KL divergence between the latent representation $q(\mathbf{z} \mid \mathbf{x}^i)$ and the prior distribution $p(\mathbf{z})$. A variant, the β-VAE [16], introduces a weighting β on the KL term for regulating the trade-off between reconstruction (first term) and the proximity to the prior. Our analysis will automatically cover this case as well.

Finally, the prior $p(\mathbf{z})$ is set to $\mathcal{N}(0, \mathcal{I})$ and the encoder is assumed to have the form

$$\mathrm{Enc}_\varphi(\mathbf{x}) \sim q_\varphi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mu_\varphi(\mathbf{x}), \operatorname{diag} \sigma_\varphi^2(\mathbf{x})\right), \tag{3}$$

where μ_φ and σ_φ are deterministic mappings depending on parameters φ. Note particularly, that **the covariance matrix is enforced to be diagonal**. This turns out to be highly significant for the main result of this work. The KL-divergence in (2) can be computed in closed form as

$$L_{\mathrm{KL}} = \frac{1}{2} \sum_{j=1}^{d} \left(\mu_j^2(\mathbf{x}^i) + \sigma_j^2(\mathbf{x}^i) - \log \sigma_j^2(\mathbf{x}^i) - 1\right). \tag{4}$$

In practical implementations, the reconstruction term from (2) is approximated with either a square loss or a cross-entropy loss.

2.2 Disentanglement

In the context of learning interpretable representations [3, 16, 6, 2, 34] it is useful to assume that the data originates from a process with some generating factors. For

instance, for images of faces this could be face azimuth, skin brightness, hair length, and so on. Disentangled representations can then be defined as ones in which individual latent variables are sensitive to changes in individual generating factors, while being relatively insensitive to other changes [3]. Although quantifying disentanglement is nontrivial, several metrics have been proposed [20, 16, 7].

Note also, that disentanglement is impossible without first learning a sufficiently expressive latent representation capable of good reconstruction.

In an unsupervised setting, the generating factors are of course unknown and the learning has to resort to statistical properties. Linear dimensionality reduction techniques demonstrate the two basic statistical approaches. Principle Components Analysis (PCA) greedily isolates sources of variance in the data, while Independent Component Analysis (ICA) recovers a factorized representation, see [33] for a recent review.

One important point to make is that **disentanglement is sensitive to rotations of the latent embedding**. Following the example above, let us denote by a, s, and h, continuous values corresponding to face azimuth, skin brightness, and hair length. Then, if we change the ideal latent representation as follows

$$\begin{pmatrix} a \\ s \\ h \end{pmatrix} \mapsto \begin{pmatrix} 0.75a + 0.25s + 0.61h \\ 0.25a + 0.75s - 0.61h \\ -0.61a + 0.61s + 0.50h \end{pmatrix}, \tag{5}$$

we obtain a representation that is equally expressive in terms of reconstruction (in fact we only multiplied with a 3D rotation matrix) but individual latent variables entirely lost their interpretable meaning.

2.3 PCA and Latent Representations

Let us examine more closely how PCA chooses the alignment of the latent embedding and why it matters.

It is well known [5] that for a linear autoencoder with encoder $Y' \in \mathbb{R}^{d\times n}$, decoder $Y \in \mathbb{R}^{n\times d}$, and square error as reconstruction loss, the objective

$$\min_{Y,Y'} \sum_{\mathbf{x}^i \in X} \|\mathbf{x}^i - YY'\mathbf{x}^i\|^2 \tag{6}$$

is minimized by the PCA decomposition. Specifically, by setting $Y' = P_d$, and $Y = P_d^\top$, for $P_d = \mathcal{I}_{d\times n} P \in \mathbb{R}^{d\times n}$, where $P \in \mathbb{R}^{n\times n}$ is an orthogonal matrix formed by the n normalized eigenvectors (ordered by the magnitudes of the corresponding eigenvalues) of the sample covariance matrix of X and $\mathcal{I}_{d\times n} \in \mathbb{R}^{d\times n}$ is a trivial projection matrix.

However, there are many minimizers of (6) that do not induce the same latent representation. In fact, it suffices to append Y' with some invertible transformations (e.g. rotations and scaling) and prefix Y with their inverses. This geometrical intuition is well captured using the singular value decomposition (SVD), see also Figure 1.

Theorem 1 (SVD rephrased, [12])**.** *Let $M\colon \mathbb{R}^n \to \mathbb{R}^d$ be a linear transformation (matrix). Then there exist*

- *$U\colon \mathbb{R}^n \to \mathbb{R}^n$, an orthogonal transformation (matrix) of the input space,*
- *$\Sigma\colon \mathbb{R}^n \to \mathbb{R}^d$ a "scale-and-embed" transformation (induced by a diagonal matrix),*
- *$V\colon \mathbb{R}^d \to \mathbb{R}^d$, an orthogonal transformation (matrix) of the output space*

such that $M = V\Sigma U^\top$.

Remark 1. *Since orthogonal transformations of the latent space will play a vital role in further considerations, we will for brevity refer to them (with slight abuse of terminology) simply as* rotations *of the latent space.*

Now we can adequately decribe the minimizers of (6).

Example 1 (Other minimizers of the PCA objective)**.** *Define Y and Y' with their SVDs as $Y = P^\top \Sigma Q$ and and its pseudoinverse $Y' = Y^\dagger = Q^\top \Sigma^\dagger P$ and see that*

$$YY' = P^\top \Sigma Q Q^\top \Sigma^\dagger P = P^\top \mathcal{I}_{d\times n} \mathcal{I}_{n\times d} P = P_d^\top P_d \tag{7}$$

so they are indeed also minimizers of the objective (6) irrespective of our choice of Q and Σ.

It is also straightforward to check that the only choices of Q, which respect the coordinate axes given by PCA, are for $|Q|$ to be a permutation matrix.

The take-away message (valid also in the non-linear case) from this example is:

Different rotations of the same latent space are equally suitable for reconstruction.

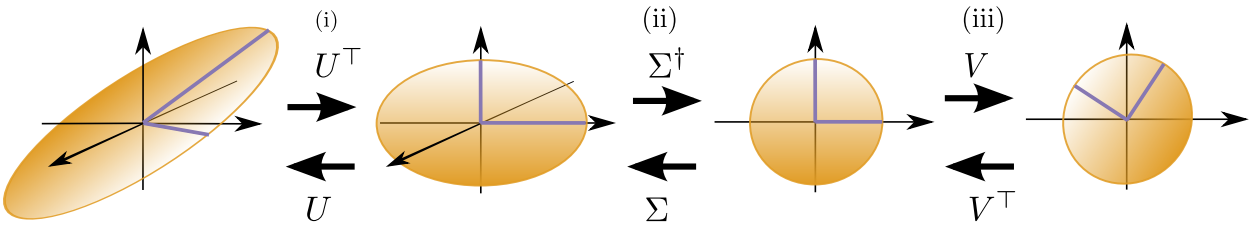

Figure 1: Geometric interpretation of the singular value decomposition (SVD). Sequential illustration of the effects of applying the corresponding SVD matrices of the encoder transformation $V\Sigma^\dagger U^\top$ (left to right) and decoder $U\Sigma V^\top$ (right to left). We notice that steps (i) and (ii) of the encoder preserve the principle directions of the data. Step (iii), however, causes misalignment. In that regard, good encoders are the ones for which step (iii) is trivial. The same argument works for the decoder (in reverse order). This condition is equivalent (for non-degenerate transformations) to $U\Sigma V^\top$ having orthogonal columns (See Proposition 1, where this is phrased for the decoder).

Following the PCA example, we formalize which linear mappings have the desired "axes-preserving" property.

Proposition 1 (Axes-preserving linear mappings)**.** *Assume $M \in \mathbb{R}^{n\times d}$ with $d < n$ has d distinct nonzero singular values. Then the following statements are equivalent:*

(a) The columns of M are (pairwise) orthogonal.

(b) In every SVD of M as $M = U\Sigma V^\top$, $|V|$ is a permutation matrix.

We strongly suggest developing a geometrical understanding for both cases (a) and (b) via Figure 1.

Take into consideration that once the encoder preserves the principle directions of the data, this already ensures an axis-aligned embedding. The same is true also if the decoder is axes-preserving, provided the reconstruction of the autoencoder is accurate.

How the requirement of distinct non-zero singular values also reflects in practical application is discussed in Suppl. 9.2.

2.4 Related work

Due to high activity surrounding VAEs, additional care is needed when it comes to evaluating novelty. To the best of our knowledge, two recent works (one of which is concurrent) address related questions and require special attention.

The authors of [6] also aim to explain good performance of (β–)VAE in disentanglement tasks. A compelling intuitive picture of the underlying dynamics is drawn and supporting empirical evidence is given. In particular, the authors *hypothesize* that "β–VAE finds latent components which make different contributions to the log-likelihood term of the cost function [reconstruction loss]", while suspecting that the diagonal posterior approximation is responsible for this behavior. Our theoretical analysis confirms both conjectures (see Section 4).

Concurrent work [2] develops ISA-VAE; another VAE-based architecture suited for disentanglement. Some parts of the motivation overlap with the content of our work. First, rotationally nonsymmetric priors are introduced for reasons similar to the content of Section 3.1. And second, both orthogonalization and alignment with PCA directions are empirically observed for VAEs applied to toy tasks.

3 Results

3.1 The problem with log-likelihood

The message from Example 1 and from the discussion about disentanglement is clear: latent space *rotation* matters. Let us look how the idealized objectives (1) and (2) handle this.

For a fixed rotation matrix U we will be comparing a baseline encoder-decoder pair $(\mathrm{Enc}_\varphi, \mathrm{Dec}_\theta)$ with a pair

$(\mathrm{Enc}_{\varphi,U}, \mathrm{Dec}_{\theta,U})$ defined as

$$\mathrm{Enc}_{\varphi,U}(\mathbf{x}) = U\,\mathrm{Enc}_{\varphi}(\mathbf{x}), \tag{8}$$

$$\mathrm{Dec}_{\theta,U}(\mathbf{z}) = \mathrm{Dec}_{\theta}(U^{\top}\mathbf{z}). \tag{9}$$

The shortcomings of idealized losses are summarized in the following propositions.

Proposition 2 (Log-likelihood rotation invariance)**.** *Let φ, θ be any choice of parameters for encoder-decoder pair $(\mathrm{Enc}_{\varphi,U}, \mathrm{Dec}_{\theta,U})$. Then, if the prior $p(\mathbf{z})$ is rotationally symmetric, the value of the log-likelihood objective* (1) does not depend *on the choice of U.*

Note that the standard prior $\mathcal{N}(0, \mathcal{I})$ is rotationally symmetric. This defficiency is not salvaged by the ELBO approximation.

Proposition 3 (ELBO rotation invariance)**.** *Let φ, θ be any choice of parameters for encoder-decoder pair $(\mathrm{Enc}_{\varphi,U}, \mathrm{Dec}_{\theta,U})$. Then, if the prior $p(\mathbf{z})$ is rotationally symmetric, the value of the ELBO objective* (1) does not depend *on the choice of U.*

We do not claim novelty of these propositions, however we are not aware of their formalization in the literature. The proofs can be found in Supplementary Material (Suppl. 7). An important point now follows:

Log-likelihood based methods (with rotationally symmetric priors) cannot claim to be designed to produce disentangled representations.

However, enforcing a diagonal posterior of the VAE encoder (3) disrupts the rotational symmetry and consequently the resulting objective (4) escapes the invariance arguments. Moreover, as we are about to see, this diagonalization comes with beneficial effects regarding disentanglement. We assume this diagonalization was primarily introduced for different reasons (tractability, computational convenience), hence the "by accident" part of the title.

3.2 Reformulating VAE loss

The fact that VAEs were *not meant* to promote orthogonality reflects in some technical challenges. For one, we cannot follow a usual workflow of a theoretical argument; set up an idealized objective and find suitable approximations which allow for stochastic gradient descent (a top-down approach). We need to do the exact opposite, start with the *implemented loss function* and find the right simplifications that allow isolating the effects in question while preserving the original training dynamics (a bottom-up approach). This is the main content of this section.

First, we formalize the typical situation in which VAE architectures "shut down" (fill with pure noise) a subset of latent variables and put high precision on the others.

Definition 1. *We say that parameters φ, θ induce a* polarized regime *if the latent coordinates $\{1, 2, \ldots, d\}$ can be partitioned as $V_a \cup V_p$ (sets of active and passive variables) such that*

(a) $\mu_j^2(\mathbf{x}) \ll 1$ *and* $\sigma_j^2(\mathbf{x}) \approx 1$ *for* $j \in V_p$,

(b) $\sigma_j^2(\mathbf{x}) \ll 1$ *for* $j \in V_a$,

(c) The decoder ignores the passive latent components, i.e.

$$\frac{\partial\,\mathrm{Dec}_{\theta}(z)}{\partial z_j} = 0 \quad \forall j \in V_p.$$

The polarized regime simplifies the loss L_{KL} from (4); part (a) ensures zero loss for passive variables and part (b) implies that $\sigma_j^2(\mathbf{x}) \ll -\log(\sigma_j^2(\mathbf{x}))$. All in all, the per-sample-loss reduces to

$$L_{\approx\mathrm{KL}}(\mathbf{x}^i) = \frac{1}{2}\sum_{j\in V_a}\left(\mu_j^2(\mathbf{x}^i) - \log(\sigma_j^2(\mathbf{x}^i)) - 1\right). \tag{10}$$

We will assume the VAE operates in the polarized regime. In Section 5.2, we show on multiple tasks and datasets that the two objectives align very early in the training. This behavior is well-known to practitioners.

Also, we approximate the reconstruction term in (2), as it is most common, with a square loss

$$L_{\mathrm{rec}}(\mathbf{x}^i) = \mathbb{E}\,\|\,\mathrm{Dec}_{\theta}(\mathrm{Enc}_{\varphi}(\mathbf{x}^i)) - \mathbf{x}^i\|^2 \tag{11}$$

where the expectation is over the stochasticity of the encoder. All in all, the loss we will analyze has the form

$$\sum_{\mathbf{x}^i\in X} L_{\mathrm{rec}}(\mathbf{x}^i) + L_{\approx\mathrm{KL}}(\mathbf{x}^i). \tag{12}$$

Moreover, the reconstruction loss can be further decomposed into two parts; deterministic and stochastic. The former is defined by

$$\overline{L}_{\text{rec}}(\mathbf{x}^i) = \| \operatorname{Dec}_\theta(\mu(\mathbf{x}^i)) - \mathbf{x}^i \|^2 \tag{13}$$

and captures the square loss of the mean encoder. Whereas the stochastic loss

$$\hat{L}_{\text{rec}}(\mathbf{x}^i) = \mathbb{E}\, \| \operatorname{Dec}_\theta(\mu(\mathbf{x}^i)) - \operatorname{Dec}_\theta(\operatorname{Enc}_\varphi(\mathbf{x}^i)) \|^2 \tag{14}$$

is purely induced by the noise injected in the encoder.

Proposition 4. *If the stochastic estimate* $\operatorname{Dec}_\theta(\operatorname{Enc}_\varphi(\mathbf{x}^i))$ *is unbiased around* $\operatorname{Dec}_\theta(\mu(\mathbf{x}^i))$*, then*

$$L_{rec}(\mathbf{x}^i) = \overline{L}_{rec}(\mathbf{x}^i) + \hat{L}_{rec}(\mathbf{x}^i). \tag{15}$$

This decomposition resembles the classical bias-variance decomposition of the square error [19].

3.3 The main result

Now, we finally give theoretical evidence for the central claim of the paper:

Optimizing the stochastic part of the reconstruction loss promotes local orthogonality of the decoder.

On that account, we set up an optimization problem which allows us to optimize the stochastic loss (14) independently of the other two. This will isolate its effects on the training dynamics.

In order to make statements about local orthogonality, we introduce for each $\mathbf{x}^i$ the Jacobian (linear approximation) J_i of the decoder at point $\mu(\mathbf{x}^i)$, i.e.

$$J_i = \frac{\partial \operatorname{Dec}_\theta(\mu(\mathbf{x}^i))}{\partial \mu(\mathbf{x}^i)}.$$

Since, according to (3), the encoder can be written as $\operatorname{Enc}_\varphi(\mathbf{x}^i) = \mu(\mathbf{x}^i) + \varepsilon(\mathbf{x}^i)$ with

$$\varepsilon(\mathbf{x}^i) \sim \mathcal{N}\left(0, \operatorname{diag} \sigma^2(\mathbf{x}^i)\right), \tag{16}$$

we can approximate the stochastic loss (14) with

$$\begin{aligned} &\mathop{\mathbb{E}}_{\varepsilon(\mathbf{x}^i)} \left\| \operatorname{Dec}_\theta(\mu(\mathbf{x}^i)) - \left(\operatorname{Dec}_\theta(\mu(\mathbf{x}^i)) + J_i \varepsilon(\mathbf{x}^i) \right) \right\|^2 \\ &\quad = \mathop{\mathbb{E}}_{\varepsilon(\mathbf{x}^i)} \| J_i \varepsilon(\mathbf{x}^i) \|^2, \end{aligned} \tag{17}$$

Although we aim to fix the deterministic loss (13), we do not need to freeze the mean encoder and the decoder entirely. Following Example 1, for each J_i and its SVD $J_i = U_i \Sigma_i V_i^\top$, we are free to modify V_i as long we correspondingly (locally) modify the mean encoder.

Then we state the optimization problem as follows:

$$\min_{V_i, \sigma_j^i > 0} \quad \sum_{\mathbf{x}^i \in X} \log \mathop{\mathbb{E}}_{\varepsilon(\mathbf{x}^i)} \| J_i \varepsilon(\mathbf{x}^i) \|^2 \tag{18}$$

$$\text{s.t.} \quad \sum_{\mathbf{x}^i \in X} L_{\approx \text{KL}}(\mathbf{x}^i) = C, \tag{19}$$

where $\varepsilon(\mathbf{x}^i)$ are sampled as in (16).

A few remarks are now in place.

- This optimization is not over network parameters, rather directly over the values of all V_i, σ_j^i (only constrained by (19)).
- Both the objective and the constraint concern *global losses*, not per sample losses.
- Indeed, none of V_i, σ_j^i interfere with the rest of the VAE objective (12).

The presence of the (monotone) log function has one main advantage; we can describe **all global minima** of (18) in closed form. This is captured in the following theorem, the technical heart of this work.

Theorem 2 (Main result)**.** *The following holds for optimization problem (18, 19):*

(a) Every local minimum is a global minimum.

(b) In every global minimum, the columns of every J_i *are orthogonal.*

The full proof as well as an explicit description of the minima is given in Suppl. 7.1. However, an outline of the main steps is given in the next section on the example of a linear decoder.

The presence of the log term in (18) admittedly makes our argument indirect. There are, however, a couple of points to make. First, as was mentioned earlier, encouraging orthogonality was *not a design feature* of the VAE. In this sense, it is unsurprising that our results are also mildly indirect.

Also, and more importantly, the global optimality of Theorem 2 also implies that, locally, orthogonality is encouraged even for the pure (without logarithm) stochastic loss.

Corollary 1. *For fixed $\mathbf{x}^i \in X$ consider a subproblem of (18) defined as*

$$\min_{V_i, \sigma_j^i > 0} \quad \mathop{\mathbb{E}}_{\varepsilon(\mathbf{x}^i)} \|J_i \varepsilon(\mathbf{x}^i)\|^2 \tag{20}$$

$$\text{s. t.} \qquad L_{\approx KL}(\mathbf{x}^i) = C_i. \tag{21}$$

Also then, the result on the structure of local (global) minima holds:

(a) Every local minimum is a global minimum.

(b) In every global minimum, the columns of every J_i are orthogonal.

All in all, Theorem 2 justifies the central message of the paper stated at the beginning of this section. The analogy with PCA is now also clearer. Locally, VAEs optimize a tradeoff between reconstruction and orthogonality.

This result is unaffected by the potential β term in Equation (2), although an appropriate β might be required to ensure the polarized regime.

4 Proof outline

In this section, we sketch the key steps in the proof of Theorem 2 and, more notably, the intuition behind them. The full proof can be found in Suppl. 7.1.

We will restrict ourselves to a simplified setting. Consider a linear decoder M with SVD $M = U\Sigma V^T$, which removes the necessity of local linearization. This reduces the objective (18) from a "global" problem over all examples $\mathbf{x}^i$ to an objective where we have the same subproblem for each $\mathbf{x}^i$.

As in optimization problem (18, 19), we resort to fixing the mean encoder (imagine a well performing one).

In the next paragraphs, we separately perform the optimization over the parameters $\boldsymbol{\sigma}$ and the optimization over the matrix V.

4.1 Weighting precision

For this part, we fix the decoder matrix M and optimize over values $\boldsymbol{\sigma}^2 = (\sigma_1^2, \dots, \sigma_d^2)$. The simplified objective is

$$\min_{\boldsymbol{\sigma}} \quad \mathop{\mathbb{E}}_{\varepsilon \sim \mathcal{N}(0, \mathrm{diag}(\sigma^2))} \|M\varepsilon\|^2 \tag{22}$$

$$\text{s. t.} \qquad \sum_j -\log \sigma_j^2 = C, \tag{23}$$

where the $\|\mu\|^2$ terms from (10) disappear since the mean encoder is fixed.

The values $-\log(\sigma_j)$ can now be thought of as precisions allowed for different latent coordinates. The log functions even suggests thinking of the number of significant digits. Problem (22) then asks to distribute the "total precision budget" so that the deviation from decoding "uncorrupted" values is minimal.

We will now solve this problem on an example linear decoder $M_1 : \mathbb{R}^2 \to \mathbb{R}^3$ given by

$$M_1 : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} 4x + y \\ -3x + y \\ 5x - y \end{pmatrix}. \tag{24}$$

Already here we see, that the latent variable x seems more influential for the reconstruction. We would expect that x receives higher precision than y.

Now, for $\boldsymbol{\varepsilon} = (\varepsilon_x, \varepsilon_y)$, we compute

$$\|M_1\boldsymbol{\varepsilon}\|^2 = \|4\varepsilon_x + \varepsilon_y\|^2 + \|-3\varepsilon_x + \varepsilon_y\|^2 + \|5\varepsilon_x - \varepsilon_y\|^2$$

and after taking the expectation, we can use the fact that $\boldsymbol{\varepsilon}$ has zero mean and write

$$\mathbb{E}\,\|M_1\boldsymbol{\varepsilon}\|^2 = \\ \mathrm{var}(4\varepsilon_x + \varepsilon_y) + \mathrm{var}(-3\varepsilon_x + \varepsilon_y) + \mathrm{var}(5\varepsilon_x - \varepsilon_y).$$

Finally, we use that for uncorrelated random variables A and B we have $\mathrm{var}(A + cB) = \mathrm{var}\,A + c^2\,\mathrm{var}\,B$. After rearranging we obtain

$$\begin{aligned}\mathbb{E}\,\|M_1\boldsymbol{\varepsilon}\|^2 &= \sigma_x^2(4^2 + (-3)^2 + 5^2) + \sigma_y^2(1^2 + 1^2 + (-1)^2) \\ &= 50\sigma_x^2 + 3\sigma_y^2,\end{aligned}$$

where $\boldsymbol{\sigma} = (\sigma_x^2, \sigma_y^2)$. Note that the coefficients are the **squared norms of the column vectors** of M_1.

This turns the optimization problem (22) into a simple exercise, particularly after realizing that (23) fixes the value of the product $\sigma_x\sigma_y$. Indeed, we can even

set $a^2 = 50\sigma_x$ and $b^2 = 3\sigma_y$ in the trivial inequality $a^2 + b^2 \geq 2ab$ and find that

$$\mathbb{E}\,\|M_1\varepsilon\|^2 = 50\sigma_x^2 + 3\sigma_y^2 \geq 2 \cdot \sqrt{50 \cdot 3} \cdot e^{-C} \approx 24.5e^{-C}, \tag{25}$$

with equality achieved when $\sigma_x^2/\sigma_y^2 = 3/50$. This also implies that the precision $-\log \sigma_x^2$ on variable x will be considerably higher than for y, just as expected.

Two remarks regarding the general case follow.

- The full version of inequality (25) relies on the concavity of the log function; in particular, on (a version of) Jensen's inequality.
- The minimum value of the objective depends on the product of the column norms. This also carries over to the unsimplified setting.

4.2 Isolating sources of variance

Now that we can find optimal values of precision, the focus changes on optimally rotating the latent space. In order to understand how such rotations influence the minimum of objective (22), let us consider the following example in which we again resort to decoder matrix $M_2 \colon \mathbb{R}^2 \to \mathbb{R}^3$.

Imagine, the encoder alters the latent representation by a $45°$ rotation. Then we can adjust the decoder M_1 by first undoing this rotation. In particular, we set $M_2 = M_1 R_{45°}^\top$, where R_θ is a 2D rotation matrix, rotating by angle θ. We have

$$M_2 \colon \begin{pmatrix} x' \\ y' \end{pmatrix} \mapsto \begin{pmatrix} \frac{1}{2}\sqrt{2}(3x' + 5y') \\ \sqrt{2}(-2x' - y') \\ \sqrt{2}(3x' + 2y') \end{pmatrix}$$

and performing analogous optimization as before gives

$$\mathbb{E}\,\|M_2\varepsilon\|^2 = \frac{61}{2}\sigma_x^2 + \frac{45}{2}\sigma_y^2 \geq 2\sqrt{\frac{61 \cdot 45}{4}}e^{-C} \approx 52.4e^{-C}. \tag{26}$$

We see that the minimal value of the objective is more than twice as high, a substantial difference. On a high level, the reason M_1 was a better choice of a decoder is that the variables x and y had very different impact on the reconstruction. This allowed to save some precision on variable y, as it had smaller effect, and use it on x, where it is more beneficial.

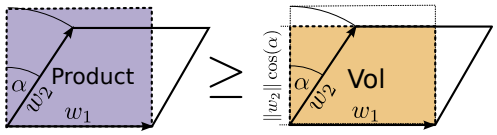

Figure 2: 2D illustration of orthogonality in $MV^\top$. The vectors w_1, w_2 are the columns of $MV^\top$. Minimizing the product $\|w_1\|\|w_2\|$ while maintaining the volume $\|w_1\|\|w_2\|\cos(\alpha)$ results in $w_1 \perp w_2$.

For a higher number of latent variables, one way to achieve a "maximum stretch" among the impacts of latent variables, is to pick them greedily, always picking the next one so that its impact is maximized. This is, at heart, the greedy algorithm for PCA.

Let us consider a slightly more technical statement. We saw in (25) and (26) that after finding optimal values of $\boldsymbol{\sigma}$ the remaining objective is the product of the column norms of matrix M. Let us denote such quantity by $\mathrm{col}_\Pi(M) = \prod_j \|M_{\cdot j}\|$. Then for a fixed matrix M, we optimize

$$\min_V \mathrm{col}_\Pi(MV^\top) \tag{27}$$

over orthogonal matrices V.

This problem can be interpreted geometrically. The column vectors of $MV^\top$ are the images of base vectors e_j. Consequently, the product gives an upper bound on the volume (the image of the unit cube)

$$\prod_j \|MV^\top e_j\| \geq \mathrm{Vol}(\{MV^\top x \colon x \in [0,1]^d\})\,. \tag{28}$$

However, as orthogonal matrices V are isometries, they do not change this volume. Also, the bound (28) is tight precisely when the vectors $MV^\top e_j$ are orthogonal. Hence, the only way to optimize $\mathrm{col}_\Pi(MV^\top)$ is by tightening the bound, that is by finding V for which the column vectors of $MV^\top$ are orthogonal, see Figure 2 for an illustration. In this regards, it is important that M performs a different scaling along each of the axis (using Σ), which allows for changing the angles among the vectors $MV^\top e_j$ (cf. Figure 1).

Table 1: Percentage of training time where $\Delta_{KL} < 3\,\%$ (Eq. (30)) continuously until the end. Reported for β-VAE with low (dataset dependent) and high (10) latent dimension.

	β-**VAE** (dep.)	β-**VAE** (10)
dSprites	97.8 %	90.6 %
fMNIST	99.8 %	97.7 %
MNIST	99.8 %	99.5 %
Synth. Lin.	99.8 %	96.7 %
Synth. Non-Lin.	99.9 %	98.5 %

5 Experiments

We performed several experiments with different architectures and datasets to validate our results empirically. We show the prevalence of the polarized regime, the strong orthogonal effects of the (β-)VAE, as well as the links to disentanglement.

5.1 Setup

Architectures. We evaluate the classical VAE, β-VAE, a plain autoencoder, and β-VAE$_\Sigma$, where the latter removes the critical diagonal approximation (3) and produces a full covariance matrix $\Sigma(\mathbf{x}^i)$ for every sample. The resulting KL term of the loss is changed accordingly (see Suppl. 8.3 for details).

Datasets. We evaluate on the well-known datasets dSprites [28], MNIST [26] and FashionMNIST [38], as well as on two synthetic ones. For both synthetic tasks the input data X is generated by embedding a unit square $V = [0,1]^2$ into a higher dimension. The latent representation is then expected to be disentangled with respect to axes of V. In one case (*Synth. Lin.*) we used a linear transformation $f_{\text{lin}}\colon \mathbb{R}^2 \to \mathbb{R}^3$ and in the other one a non-linear (*Synth. Non-Lin.*) embedding $f_{\text{non-lin}}\colon \mathbb{R}^2 \to \mathbb{R}^6$. The exact choice of transformations can be found in Suppl. 8. Further information regarding network structures and training parameters is also provided in Suppl. 8.4.

Disentanglement metric. For quantifying the disentanglement of a representation, the so called Mutual Information Gap (MIG) was introduced in [7]. As MIG is not well defined for continuous variables, we use an adjusted definition comprising both continuous and discrete variables, simply referred to as *Disentanglement score*. Details are described in Suppl. 8.1. Just as in the case of MIG, the Disentanglement score is a number between 0 and 1, where higher value means stronger disentanglement.

Orthogonality metric. For measuring the practical effects of Theorem 2, we introduce a measure of non-orthogonality. As argued in Proposition 1 and Figure 1, for a good decoder M and its SVD $M = U\Sigma V^\top$, the matrix V should be trivial (a signed permutation matrix). We measure the non-triviality with the *Distance to Orthogonality* (DtO) defined as follows. For each $\mathbf{x}^i$, $i = 1, \ldots, N$, employing again the Jacobian J_i of the decoder at $\mathbf{x}^i$ and its SVD $J_i = U_i \Sigma_i V_i^\top$ and define

$$\text{DtO} = \frac{1}{N} \sum_{i=1}^{N} \|V_i - P(V_i)\|_F, \tag{29}$$

where $\|\cdot\|_F$ is the Frobenius norm and $P(V_i)$ is a signed permutation matrix that is closest to V (in L^1 sense). Finding the nearest permutation matrix is solved to optimality via mixed-integer linear programming (see Suppl. 8.2).

5.2 Polarized regime

In Section 3.2, we assumed VAEs operate in a polarized regime and approximated L_{KL}, the KL term of the implemented objective (4), with $L_{\approx\text{KL}}$ (10). In Table 1 we show that the polarized regime is indeed dominating the training in all examples after a short initial phase. We report the fraction of the training time in which the relative error

$$\Delta_{KL} = \frac{|L_{\text{KL}} - L_{\approx\text{KL}}|}{L_{\text{KL}}} \tag{30}$$

stays below 3 % continuously until the end (evaluated every 500 batches).

5.3 Orthogonality and Disentanglement

Now, we provide evidence for Theorem 2 by investigating the DtO (29) for a variety of architectures and datasets,

Table 2: Results for the distance to orthogonality DtO of the decoder (Equation 29) and disentanglement score for different architectures and datasets. Lower DtO values are better and higher Disent. values are better. Random decoders provide a simple baseline for the numbers.

		β-VAE	VAE	AE	β-VAE$_\Sigma$	Random Decoder
dSprites	**Disent.** ↑	$\mathbf{0.33 \pm 0.15}$	0.21 ± 0.10	0.09 ± 0.04	0.12 ± 0.06	
	DtO ↓	$\mathbf{0.76 \pm 0.08}$	1.08 ± 0.15	1.62 ± 0.03	1.73 ± 0.14	1.86 ± 0.11
Synth. Lin.	**Disent.** ↑	$\mathbf{0.99 \pm 0.01}$	–	0.71 ± 0.19	0.71 ± 0.31	
	DtO ↓	$\mathbf{0.00 \pm 0.00}$	–	0.33 ± 0.18	0.34 ± 0.35	0.79 ± 0.21
Synth. Non-Lin.	**Disent.** ↑	$\mathbf{0.73 \pm 0.16}$	–	0.59 ± 0.30	0.42 ± 0.24	
	DtO ↓	$\mathbf{0.18 \pm 0.02}$	–	0.54 ± 0.13	0.55 ± 0.02	0.89 ± 0.16
MNIST	**DtO** ↓	–	$\mathbf{1.59 \pm 0.08}$	1.83 ± 0.05	1.93 ± 0.08	2.11 ± 0.11
fMNIST	**DtO** ↓	–	$\mathbf{1.36 \pm 0.05}$	1.87 ± 0.03	2.02 ± 0.08	2.11 ± 0.11

see Table 2. The results clearly support the claim that the VAE based architectures indeed strive for local orthogonality. By generalizing the β-VAE architecture, such that the approximate posterior is any multivariate Gaussian (β-VAE$_\Sigma$), the objective becomes rotationally symmetric (just as the idealized objective). As such, no specific alignment is prioritized. The simple autoencoders also do not favor particular orientations of the latent space.

Another important observation is the clear correlation between DtO and the disentanglement score. We show this in Figure 3 where different restarts of the same β-VAE architecture on the dSprites dataset are displayed. We used the state-of-the-art value $\beta = 4$ [16]. Additional experiments are reported in Suppl. 9.

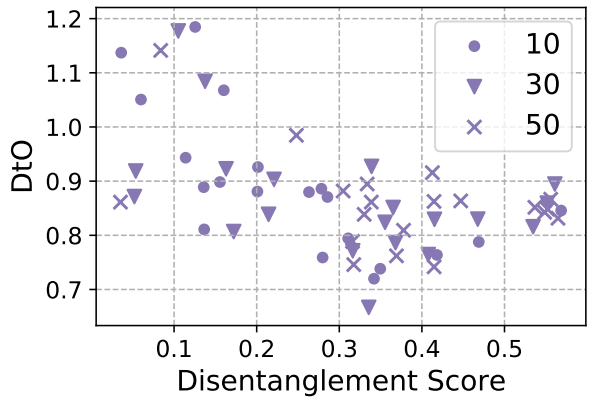

Figure 3: Alignment of the latent representation (low DtO, (29)) results in better disentanglement (higher score). Each datapoint corresponds to an independent run with 10, 30, or 50 epochs.

6 Discussion

We isolated the mechanism of VAE that leads to local orthogonalization and, in effect, to performing local PCA. Additionally, we demonstrated the functionality of this mechanism in intuitive terms, in formal terms, and also in experiments. We also explained why this behavior is desirable for enforcing disentangled representations.

Our insights show that VAEs make use of the differences in variance to form the representation in the latent space. This does not directly encourage factorized latent representations, see also Suppl. 9.2. With this in mind, it makes perfect sense that recent improvements of (β-)VAE [7, 20, 2] incorporate additional terms promoting precisely independence.

It is also unsatisfying that VAEs promote orthogonality somewhat indirectly. It would seem that designing architectures allowing explicit control over this feature would be beneficial.

Acknowledgements

We thank the whole Autonomous Learning Group at MPI IS, as well as Friedrich Solowjow for the fruitful and invaluable discussions. Also, we thank the

International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Dominik Zietlow.

References

[1] A. Alemi, B. Poole, I. Fischer, J. Dillon, R. A. Saurous, and K. Murphy. Fixing a broken ELBO. In *Proc. 35th Intl. Conference on Machine Learning (ICML)*, volume 80, pages 159–168. PMLR, 2018. 2

[2] Anonymous. ISA-VAE: Independent subspace analysis with variational autoencoders. In *Submitted to International Conference on Learning Representations*, 2019. under review (https://openreview.net/forum?id=rJl_NhR9K7). 2, 4, 10

[3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, Aug 2013. 1, 2, 3

[4] M. Blaauw and J. Bonada. Modeling and transforming speech using variational autoencoders. In *INTERSPEECH*, pages 1770–1774, 2016. 1

[5] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. Manuscript M217, Philips Research Laboratory, Brussels, Belgium, 1987. 3

[6] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner. Understanding disentangling in β-vae. *ArXiv e-prints*, abs/1804.03599, 2018. 2, 4

[7] T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. *ArXiv e-prints*, abs/1802.04942, 2018. 1, 3, 9, 10, 18

[8] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. *ArXiv e-prints*, June 2016. 1

[9] B. Dai, Y. Wang, J. Aston, G. Hua, and D. Wipf. Hidden talents of the variational autoencoder. *ArXiv e-prints*, abs/1706.05148, 2018. 2

[10] G. Desjardins, A. Courville, and Y. Bengio. Disentangling factors of variation via generative entangling. *ArXiv e-prints*, abs/1210.5474, 2012. 1

[11] A. Ghadirzadeh, A. Maki, D. Kragic, and M. Bjrkman. Deep predictive policy training using reinforcement learning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2351–2358, Sept 2017. 1

[12] G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, 2(2):205–224, 1965. 3

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 1

[14] K. Gregor, F. Besse, D. Jimenez Rezende, I. Danihelka, and D. Wierstra. Towards conceptual compression. *ArXiv e-prints*, abs/1604.08772, 2016. 1

[15] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra. Draw: A recurrent neural network for image generation. In F. Bach and D. Blei, editors, *Proc. ICML*, volume 37, pages 1462–1471. PMLR, 2015. 1

[16] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. Beta-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017. 1, 2, 3, 10

[17] I. Higgins, A. Pal, A. A. Rusu, L. Matthey, C. P. Burgess, A. Pritzel, M. Botvinick, C. Blundell, and A. Lerchner. DARLA: Improving zero-shot transfer in reinforcement learning. *ArXiv e-prints*, July 2017. 1

[18] X. Hou, L. Shen, K. Sun, and G. Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1133–1141, March 2017. 1

[19] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014. 6

[20] H. Kim and A. Mnih. Disentangling by factorising. In J. Dy and A. Krause, editors, *Proc. ICML*, volume 80, pages 2649–2658. PMLR, 2018. 1, 3, 10

[21] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4743–4751. Curran Associates, Inc., 2016. 1

[22] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4743–4751. Curran Associates, Inc., 2016. 1

[23] D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *ICLR*, 2014. 1, 2, 16

[24] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2539–2547. Curran Associates, Inc., 2015. 1

[25] M. J. Kusner, B. Paige, and J. M. Hernández-Lobato. Grammar variational autoencoder. *ArXiv e-prints*, abs/1703.01925, 2017. 1

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. 9

[27] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv e-print*, abs/1511.05644, 2015. 1

[28] L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017. 9

[29] L. Mescheder, S. Nowozin, and A. Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*. PMLR, Aug. 2017. 1

[30] A. Nair, V. Pong, M. Dalal, S. Bahl, S. Lin, and S. Levine. Visual Reinforcement Learning with Imagined Goals. *ArXiv e-prints*, abs/1807.04742, July 2018. 1

[31] D. J. Rezende, S. Mohamed, I. Danihelka, K. Gregor, and D. Wierstra. One-shot generalization in deep generative models. *ArXiv e-prints*, abs/1603.05106, 2016. 1

[32] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *ICML*, 2014. 1

[33] K. Ridgeway. A survey of inductive biases for factorial representation-learning. *ArXiv e-prints*, abs/1612.05299, 2016. 3

[34] J. Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992. 2

[35] I. V. Serban, C. Sankar, M. Germain, S. Zhang, Z. Lin, S. Subramanian, T. Kim, M. Pieper, S. Chandar, N. R. Ke, S. Mudumba, A. de Brébisson, J. Sotelo, D. Suhubdy, V. Michalski, A. Nguyen, J. Pineau, and Y. Bengio. A deep reinforcement learning chatbot. *ArXiv e-prints*, abs/1709.02349, 2017. 1

[36] L. N. Trefethen and D. Bau III. *Numerical linear algebra*, volume 50. Siam, 1997. 13

[37] H. van Hoof, N. Chen, M. Karl, P. van der Smagt, and J. Peters. Stable reinforcement learning with autoencoders for tactile and visual data. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3928–3934, Oct 2016. 1

[38] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *ArXiv e-prints*, abs/1708.07747, 2017. 9

[39] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In D. Precup and Y. W. Teh, editors, *Proc. ICML*, volume 70, pages 3881–3890. PMLR, 2017. 1

[40] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt. Advances in variational inference. *ArXiv e-prints*, abs/1711.05597, 2017. 1

Supplementary Material

The supplementary information is structured as follows. We start with a remark on Table 1 and then provide the proofs in Section 7.1. Section 8 reports the details of the experiments followed by additional experiments in Section 9.

Remark on Table 1

Some dataset-architecture combinations listed in Table 2 are omitted for the following reasons.

On the one hand, calculating the Disentanglement Score for MNIST and fMNIST does not make sense, as the generating factors are not given (the one categorical label cannot serve as replacement). Consequently, as the values of β are chosen according to this score, we do not report β-VAE numbers for these datasets. On the other hand, for either synthetic task, the regular VAE vastly overprunes, see Figure S4, and the values become meaningless.

7 Proofs

7.1 Proof of Theorem 2

Proof strategy: For part (b), we aim to derive a lower bound on the objective (18), that is independent from the optimization variables $\sigma_j^2(\mathbf{x}^i)$ and V_i. Moreover, we show that this lower bound is tight for some specific choices of $\sigma_j^2(\mathbf{x}^i)$ and V_i, i.e. the global optima. For these choices, all J_i will have orthogonal columns.

The strategy for part (a) is to show that whenever $\sigma_j^2(\mathbf{x}^i)$ and V_i do not induce a global optimum, we can find a small perturbation that decreases the objective function. Thereby showing that local minima do not exist.

Technical lemmas: We begin with introducing a few useful statements. First is the inequality between arithmetic and geometric mean; a consequence of Jensen's inequality.

Lemma S1 (AM-GM inequality)**.** *Let $a_1, \dots, a_N$ be non-negative real numbers. Then*

$$\frac{1}{N}\sum_{i=1}^{N} a_i \geq \left(\prod_{i=1}^{N} a_i\right)^{1/N} \tag{S31}$$

with equality occuring if and only if $a_1 = a_2 = \cdots = a_n$.

The second bound to be used is the classical Hadamard's inequality.

Lemma S2 (Hadamard's inequality [36])**.** *Let $M \in \mathbb{R}^{k\times k}$ be non-singular matrix with column vectors $c_1, \dots, c_k$. Then*

$$\prod_{i=1}^{k} \|c_i\| \geq |\det M| \tag{S32}$$

with equality if and only if the vectors $c_1, \dots, c_k$ are pairwise orthogonal.

And finally a simple lemma for characterizing matrices with orthogonal columns.

Lemma S3 (Column orthogonality)**.** *Let $M \in \mathbb{R}^{n\times d}$ be a matrix and let $M = U\Sigma V^\top$ be its singular value decomposition. Then the following statements are equivalent:*

(a) The columns of M are (pairwise) orthogonal.

(b) The matrix $M^\top M$ is diagonal.

(c) The columns of $\Sigma V^\top$ are (pairwise) orthogonal.

Proof. The equivalence of (a) and (b) is immediate. For equivalence of (a) and (c) it suffices to notice that if we set $M' = \Sigma V^\top$, then

$$M'^\top M' = V\Sigma^\top\Sigma V^\top = M^\top M. \tag{S33}$$

The equivalence of (a) and (b) now implies that M has orthogonal columns if and only if M' does. □

Initial considerations: First, without loss of generality, we will ignore all passive latent variables (in the sense of Definition 1). Formally speaking, we will restrict to the case when the local decoder mappings J_i are non-degenerate (i.e. have non-zero singular values). Now d denotes the dimensionality of the latent space with $d = |V_a|$.

Next, we simplify the loss $L_{\approx \mathrm{KL}}$, Equation 10. Up to additive and multiplicative constants, this loss can be, for a fixed sample $\mathbf{x}^i \in X$, written as

$$\|\mu(\mathbf{x}^i)\|^2 + \sum_{j=1}^{d} -\log(\sigma_j^2(\mathbf{x}^i)). \tag{S34}$$

In the optimization problem (18, 19) the values $\mu(\mathbf{x}^i)$ can only be affected via applying an orthogonal transformation V_i. But such transformation are norm-preserving (isometric) and hence the values $\|\mu(\mathbf{x}^i)\|^2$ do not change in the optimization. As a result, we can restate the constraint (19) as

$$\sum_{\mathbf{x}^i \in X} \sum_{j=1}^{d} -\log(\sigma_j^2(\mathbf{x}^i)) = C_1 \tag{S35}$$

for some constant C_1.

Proof of Theorem 2(b): Here, we explain how Theorem 2(b) follows from the following two propositions.

Proposition S5. *For a fixed sample $\mathbf{x}^i \in X$ let us denote by $c_1, \ldots, c_d$ the column vectors of J_i. Then*

$$\mathop{\mathbb{E}}_{\varepsilon(\mathbf{x}^i)} \|J_i \boldsymbol{\varepsilon}(\mathbf{x}^i)\|^2 \geq d \left(\prod_{j=1}^{d} \|c_j\|^2 \sigma_j^2(\mathbf{x}^i) \right)^{1/d} \tag{S36}$$

with equality if and only if $\|c_j\|^2 \sigma_j^2(\mathbf{x}^i) = \|c_k\|^2 \sigma_k^2(\mathbf{x}^i)$ for every $j, k \in \{1, \ldots, d\}$.

Proposition S6. *Let $M \in \mathbb{R}^{n \times d}$, where $d < n$, be a matrix with column vectors $c_1, \ldots, c_d$ and nonzero singular values $s_1, \ldots, s_d$. Then*

$$\prod_{j=1}^{d} \|c_j\| \geq \det{}^{\dagger}(M), \tag{S37}$$

where by $\det^{\dagger}(M)$ we denote the product of the singular values of M. Equality occurs if and only if $c_1, \ldots, c_d$ are pairwise orthogonal.

First, Proposition S6 allows making further estimates in the inequality from Proposition S5. Indeed, we get

$$\mathop{\mathbb{E}}_{\varepsilon(\mathbf{x}^i)} \|J_i \boldsymbol{\varepsilon}(\mathbf{x}^i)\|^2 \geq d \left(\left(\det{}^{\dagger}(J_i)\right)^2 \prod_{j=1}^{d} \sigma_j^2(\mathbf{x}^i) \right)^{1/d} \tag{S38}$$

and after applying the (monotonous) log function we are left with

$$\log \mathop{\mathbb{E}}_{\varepsilon(\mathbf{x}^i)} \|J_i \boldsymbol{\varepsilon}(\mathbf{x}^i)\|^2 \geq \tag{S39}$$

$$\log(d) + \frac{2}{d} \log(\det{}^{\dagger}(J_i)) + \frac{1}{d} \sum_{j=1}^{d} \log(\sigma_j^2(\mathbf{x}^i)). \tag{S40}$$

Finally, we sum over the samples $\mathbf{x}^i \in X$ and simplify via (S35) as

$$\sum_{\mathbf{x}^i \in X} \log \mathop{\mathbb{E}}_{\varepsilon(\mathbf{x}^i)} \|J_i \boldsymbol{\varepsilon}(\mathbf{x}^i)\|^2 \geq$$

$$N \log(d) - \frac{C_1}{d} + \frac{2}{d} \sum_{\mathbf{x}^i \in X} \log(\det{}^{\dagger}(J_i)). \tag{S41}$$

The right-hand side of this inequality is independent from the values of $\sigma_j^2(\mathbf{x}^i)$, as well as from the orthogonal matrices V_i, since these do not influence the singular values of any J_i.

Moreover, it is possible to make inequality (S41) tight (i.e. reach the global minimum), by setting $\sigma_j^2(\mathbf{x}^i)$ as hinted by Proposition S5 and by choosing the matrices V_i such that every J_i has orthogonal columns (this is clearly possible as seen in Proposition 1).

This yields the desired description of the global minima of (18). □

Proof of Proposition S5: We further denote by $r_1, \ldots, r_n$ the row vectors of J_i, and by $a_{r,c}$ the element of J_i at r-th row and c-th column. With sampling $\boldsymbol{\varepsilon}(\mathbf{x}^i)$ according to

$$\varepsilon(\mathbf{x}^i) \sim \mathcal{N}\left(0, \operatorname{diag} \sigma^2(\mathbf{x}^i)\right), \tag{S42}$$

we begin simplifying the objective (18) with

$$\mathop{\mathbb{E}}_{\varepsilon(\mathbf{x}^i)} \|J_i \boldsymbol{\varepsilon}(\mathbf{x}^i)\|^2 = \mathop{\mathbb{E}}_{\varepsilon(\mathbf{x}^i)} \sum_{k=1}^{n} \|r_k^{\top} \boldsymbol{\varepsilon}(\mathbf{x}^i)\|^2 \tag{S43}$$

$$= \sum_{k=1}^{n} \mathop{\mathbb{E}}_{\varepsilon(\mathbf{x}^i)} \|r_k^{\top} \boldsymbol{\varepsilon}(\mathbf{x}^i)\|^2. \tag{S44}$$

Now, as the samples $\boldsymbol{\varepsilon}(\mathbf{x}^i)$ are zero mean, we can further write

$$\sum_{k=1}^{n} \mathop{\mathbb{E}}_{\boldsymbol{\varepsilon}(\mathbf{x}^i)} \|r_k^\top \boldsymbol{\varepsilon}(\mathbf{x}^i)\|^2 = \sum_{k=1}^{n} \operatorname{var}(r_k^\top \boldsymbol{\varepsilon}(\mathbf{x}^i)). \tag{S45}$$

Now we use the fact that for uncorrelated random variables A and B we have $\operatorname{var}(A+cB) = \operatorname{var} A + c^2 \operatorname{var} B$. This allows to expand the variance of the inner product as

$$\begin{aligned} \operatorname{var}(r_k^\top \boldsymbol{\varepsilon}(\mathbf{x}^i)) &= \operatorname{var}\left(\sum_{j=1}^{d} a_{k,j} \boldsymbol{\varepsilon}_j(\mathbf{x}^i)\right) \\ &= \sum_{j=1}^{d} a_{k,j}^2 \operatorname{var} \boldsymbol{\varepsilon}_j(\mathbf{x}^i) = \sum_{j=1}^{d} a_{k,j}^2 \sigma_j^2(\mathbf{x}^i). \end{aligned} \tag{S46}$$

Now, we can regroup the terms via

$$\begin{aligned} \sum_{k=1}^{n} \operatorname{var}(r_k^\top \boldsymbol{\varepsilon}(\mathbf{x}^i)) &= \sum_{k=1}^{n} \sum_{j=1}^{d} a_{k,j}^2 \sigma_j^2(\mathbf{x}^i) \\ &= \sum_{j=1}^{d} \sum_{k=1}^{n} a_{k,j}^2 \sigma_j^2(\mathbf{x}^i) \\ &= \sum_{j=1}^{d} \|c_j\|^2 \sigma_j^2(\mathbf{x}^i). \end{aligned} \tag{S47}$$

All in all, we obtain

$$\mathop{\mathbb{E}}_{\boldsymbol{\varepsilon}(\mathbf{x}^i)} \|J_i \boldsymbol{\varepsilon}(\mathbf{x}^i)\|^2 = \sum_{j=1}^{d} \|c_j\|^2 \sigma_j^2(\mathbf{x}^i). \tag{S48}$$

from which the desired inequality follows via setting $a_j = \|c_j\|^2 \sigma_j^2(\mathbf{x}^i)$ for $j = 1, \ldots, d$ in Lemma S1. Indeed, then we have

$$\sum_{j=1}^{d} \|c_j\|^2 \sigma_j^2(\mathbf{x}^i) \geq d \left(\prod_{j=1}^{d} \|c_j\|^2 \sigma_j^2(\mathbf{x}^i)\right)^{1/d} \tag{S49}$$

as required. □

Proof of Proposition S6: As the first step, we show that both sides of the desired inequality are invariant to multiplying the matrix M from the left with an orthogonal matrix $U \in \mathbb{R}^{n \times n}$.

For the right-hand side, this is clear as the singular values of UM are identical to those of M. As for the left-hand side, we first need to realize that the vectors c_j are the images of the canonical basis vectors e_j, i.e. $c_j = Me_j$ for $j = 1, \ldots, d$. But since U is an isometry, we have $\|UMe_j\| = \|Me_j\| = \|c_j\|$ for every j, and hence also the column norms are intact by prepending U to M.

This allows us to restrict to matrices M for which the SVD has a simplified form $M = \Sigma V^\top$. Next, let us denote by $\Sigma_{d\times d}$ the $d \times d$ top-left submatrix of Σ. Note that $\Sigma_{d\times d}$ contains all nonzero elements of Σ. As a result, the matrix $M' = \Sigma_{d\times d} V^\top$ contains precisely the nonzero rows of the matrix M. This implies

$$M^\top M = M'^\top M'. \tag{S50}$$

In particular, the column vectors c'_j of M' have the same norms as those of M. Now we can write

$$\prod_{j=1}^{d} \|c_j\| = \prod_{j=1}^{d} \|c'_j\| \geq |\det(M')| = \det^\dagger(M), \tag{S51}$$

where the inequality follows from Lemma S2 applied to nonsingular matrix M'. Equality in Lemma S2 occurs precisely if the columns of M' are orthogonal. However, according to Lemma S3 and (S50), it also follows that the columns of M' are orthogonal if and only if the columns of M are. Note that Lemma S3(c) is needed for covering the reduction performed in the first two paragraphs. □

Proof of Theorem 2(a): We show the nonexistence of local minima as follows. For any values of $\sigma_j^2(\mathbf{x}^i)$ and V_i that do not minimize the objective function (18), we find a small perturbation that improves this objective.

All estimates involved in establishing inequality (S41) rely on either Lemma S1 or Lemma S2, where in both cases, the right-hand side was kept fixed. We show that both of these inequalities can be tightened in such fashion by small perturbations in their parameters.

Lemma S4 (Locally improving AM-GM)**.** *For any nonnegative values $a_1, \ldots, a_N$ for which*

$$\frac{1}{N} \sum_{i=1}^{N} a_i > \left(\prod_{i=1}^{N} a_i\right)^{1/N} \tag{S52}$$

there exists a small perturbation a'_i of a_i for $i = 1, \dots, N$ such that

$$\frac{1}{N}\sum_{i=1}^{N} a_i > \frac{1}{N}\sum_{i=1}^{N} a'_i \geq \tag{S53}$$

$$\left(\prod_{i=1}^{N} a'_i\right)^{1/N} = \left(\prod_{i=1}^{N} a_i\right)^{1/N} \tag{S54}$$

Proof. Since (S52) is a sharp inequality, we have $a_i > a_j$ for some $i \neq j$. Then setting $a'_i = a_i/(1+\delta)$, $a'_j = a_j(1+\delta)$, and $a'_k = a_k$ otherwise, will do the trick. Indeed, we have $a_i a_j = a'_i a'_j$ as well as $a_i + a_j > a'_i + a'_j$ for small enough δ. This ensures both S53 and S54. □

An analogous statement for Lemma S2 has the following form.

Lemma S5 (Locally improving Hadamard's inequality)**.** *Let $M \in \mathbb{R}^{k\times k}$ be a non-singular matrix with SVD $M = U\Sigma V^\top$, and column vectors $c_1, \dots, c_k$, for which*

$$\prod_{i=1}^{k} \|c_i\| > |\det M|. \tag{S55}$$

Then there exists an orthogonal matrix V', a small perturbation of V, such that if we denote by $c'_1, \dots, c'_k$ the column vectors of $M' = U\Sigma V'^\top$, we have

$$\prod_{i=1}^{k} \|c_i\| > \prod_{i=1}^{k} \|c'_i\|. \tag{S56}$$

Proof. We proceed by induction on k. For $k = 2$, it can be verified directly that for some small δ (in absolute value) setting $V' = VR_\delta$, where R_δ is a 2D rotation matrix by angle δ, achieves what is required.

For the general case, the sharp inequality (S55) implies that $c_i^\top c_j \neq 0$ for some pair of $i \neq j$. Without loss of generality, let $i = 1$, $j = 2$. In such case, we consider $V' = VR_\delta^{2D}$, where

$$R_\delta^{2D} = \begin{pmatrix} R_\delta & \\ & \mathcal{I}_{k-2} \end{pmatrix} \tag{S57}$$

is a block diagonal matrix, in which R_δ is again a 2×2 rotation matrix. By design, we have $c_i = c'_i$ for $i > 2$. This, along with the fact that U can be set to $\mathcal{I}_k$ (isometry does not influence either side of (S55)), allows for a full reduction to the discussed two-dimensional case. □

It is easy to see that the performed perturbations continuously translate into perturbations of the parameters $\sigma_j^2(\mathbf{x}^i)$ and V_i in estimates (S49) and (S51). Consequently, any non-optimal values of $\sigma_j^2(\mathbf{x}^i)$ and V_i can be locally improved. This concludes the proof.

7.2 Rotational invariances

Let us start by fleshing out the common elements of the proofs of Propositions 2 and 3. In both cases, the encoder and decoder mappings $\mathrm{Enc}_{\varphi,U}$, $\mathrm{Dec}_{\theta,U}$ induce joint distributions $p_U(\mathbf{x}, \mathbf{z})$, $q_U(\mathbf{x}, \mathbf{z})$ described as

$$p_U(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x} \mid U^\top \mathbf{z}) \tag{S58}$$

$$q_U(\mathbf{x}, \mathbf{z}) = q(\mathbf{x})q(U^\top \mathbf{z} \mid \mathbf{x}) \tag{S59}$$

Lemma S6. *For every $\mathbf{x}^i \in X$ we have $p(\mathbf{x}^i) = p_U(\mathbf{x}^i)$.*

Proof. We simply compute

$$\begin{aligned} p_U(\mathbf{x}^i) &= \int p_U(\mathbf{x}^i, \mathbf{z})\, \mathrm{d}\mathbf{z} \\ &= \int p(\mathbf{z})p(\mathbf{x}^i \mid U^\top \mathbf{z})\, \mathrm{d}\mathbf{z} \\ &= \int p(U\mathbf{z})p(\mathbf{x}^i \mid \mathbf{z})\, \mathrm{d}\mathbf{z} \\ &= \int p(\mathbf{z})p(\mathbf{x}^i \mid \mathbf{z})\, \mathrm{d}\mathbf{z} = p(\mathbf{x}^i), \end{aligned}$$

where in the third equality we used the Change of Variable Theorem to substitute $U\mathbf{z}$ for $\mathbf{z}$ (keep in mind that $|\det(U)| = 1$ as U is an orthogonal matrix). In the fourth equality, we used the rotational symmetry of the prior $p(\mathbf{z})$. □

Proof of Proposition 2. This immediately follows from Lemma S6. □

Proof of Proposition 3. We utilize the full identity from ELBO derivation. For fixed $\mathbf{x}^i \in X$ we have [23]

$$\mathrm{ELBO} = D_{\mathrm{KL}}(q_U(\mathbf{z} \mid \mathbf{x}^i) \parallel p_U(\mathbf{z} \mid \mathbf{x}^i)) + \log p_U(\mathbf{x}^i) \tag{S60}$$

In order to prove invariance of ELBO to the choice of U, it suffices to prove invariance of the right-hand side of (S60). Due to Proposition (3) we only need to focus on

the KL term. Similarly as in the proof of Lemma S6, we calculate

$$
\begin{aligned}
&D_{\mathrm{KL}}(q_U(\mathbf{z} \mid \mathbf{x}^i) \parallel p_U(\mathbf{z} \mid \mathbf{x}^i)) \\
&= \int q_U(\mathbf{z} \mid \mathbf{x}^i) \log \frac{q_U(\mathbf{z} \mid \mathbf{x}^i)}{p_U(\mathbf{z} \mid \mathbf{x}^i)} \, \mathrm{d}\mathbf{z} \\
&= \int q_U(\mathbf{z} \mid \mathbf{x}^i) \log \frac{q_U(\mathbf{z} \mid \mathbf{x}^i) \cdot p_U(\mathbf{x}^i)}{p_U(\mathbf{z}) \cdot p_U(\mathbf{x}^i \mid \mathbf{z})} \, \mathrm{d}\mathbf{z} \\
&\overset{(3)}{=} \int q(U^\top \mathbf{z} \mid \mathbf{x}^i) \log \frac{q(U^\top \mathbf{z} \mid \mathbf{x}^i) \cdot p(\mathbf{x}^i)}{p(\mathbf{z}) \cdot p(\mathbf{x}^i \mid U^\top \mathbf{z})} \, \mathrm{d}\mathbf{z} \\
&\overset{(4)}{=} \int q(\mathbf{z} \mid \mathbf{x}^i) \log \frac{q(\mathbf{z} \mid \mathbf{x}^i) \cdot p(\mathbf{x}^i)}{p(\mathbf{U}\mathbf{z}) \cdot p(\mathbf{x}^i \mid \mathbf{z})} \, \mathrm{d}\mathbf{z} \\
&\overset{(5)}{=} \int q(\mathbf{z} \mid \mathbf{x}^i) \log \frac{q(\mathbf{z} \mid \mathbf{x}^i) \cdot p(\mathbf{x}^i)}{p(\mathbf{z}) \cdot p(\mathbf{x}^i \mid \mathbf{z})} \, \mathrm{d}\mathbf{z} \\
&= \int q(\mathbf{z} \mid \mathbf{x}^i) \log \frac{q(\mathbf{z} \mid \mathbf{x}^i)}{p(\mathbf{z} \mid \mathbf{x}^i)} \, \mathrm{d}\mathbf{z} \\
&= D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x}^i) \parallel p(\mathbf{z} \mid \mathbf{x}^i)),
\end{aligned}
$$

where we again used the Change of Variable Theorem in equality (4), rotational symmetry of $p(\mathbf{z})$ in equality (5), and Lemma S6 in equality (3). □

7.3 Other proofs

Proof of Proposition 1. Recall from Lemma S3 that column orthogonality of M is equivalent to $M^\top M$ being a diagonal matrix.

(b) ⇒ (a): Let $M = U\Sigma V^\top$ where $|V|$ is a permutation matrix. Then

$$M^\top M = V\Sigma^\top U^\top U \Sigma V^\top = V\Sigma' V^\top \tag{S61}$$

where $\Sigma' = \Sigma^\top \Sigma$ is a diagonal matrix. But then $V\Sigma' V^\top$ only permutes the diagonal entries of Σ' (and possibly flips their signs). In particular, $V\Sigma' V^\top$ is also diagonal.

(a) ⇒ (b): Let again $M = U\Sigma V^\top$ be some SVD of M and assume $M^\top M = D$ for some diagonal matrix D. Since M has d distinct nonzero singular values, $M^\top M$ has d distinct nonzero eigenvalues (diagonal elements). Moreover, these eigenvalues are precisely the squares of the singular values captured by Σ. Next, if we denote by P the permutation matrix for which PDP^{-1} has decreasing diagonal elements, we can write

$$PDP^{-1} = \Sigma^\top \Sigma \tag{S62}$$

Then using (S62) and the SVD of M similarly as in (S61), we obtain

$$D = M^\top M = V\Sigma^\top \Sigma V^\top = VPDP^{-1}V^\top. \tag{S63}$$

Further, the resulting identity $(VP)D = D(VP)$ implies that columns of VP are eigenvectors of D, i.e. the canonical basis vectors. Since VP is additionally orthogonal, these eigenvectors are normalized. It follows that $|VP|$ is a permutation matrix and the conclusion follows. □

Proof of Proposition 4. First, note that for any random variable $\mathbf{X} \in \mathbb{R}^k$ with $\mathbb{E}\mathbf{X} = \mu$ and a constant $\mathbf{b} \in \mathbb{R}^k$, the following identity holds

$$\mathbb{E} \|\mathbf{X} - \mathbf{b}\|^2 = \mathbb{E} \|\mathbf{X} - \mu\|^2 + \|\mu - \mathbf{b}\|^2. \tag{S64}$$

In our case, we set $\mathbf{X} = \mathrm{Dec}_\theta(\mathrm{Enc}_\varphi(\mathbf{x}^i))$, the unbiasedness assumption translates to $\mathbb{E}\mathbf{X} = \mathrm{Dec}_\theta(\mu(\mathbf{x}^i))$, and finally we set $\mathbf{b} = \mathbf{x}^i$.

The identity we obtain, is exactly what was required to prove. □

8 Experimental details

8.1 Disentanglement Score

As introduced in the paper, for disentangled representations, single latent variables should be sensitive to individual generating factors and insensitive to all others. To quantify this behavior, for each generating factor w_i, all latent variables are evaluated for their sensitivity to w_i. The sensitivity difference between the two most responsive variables then reflects both desired properties; the sensitivity of the associated best matching latent variable and also the insensitivity of all others. A set of quantities capturing disentanglement can therefore be described as

$$\text{Disent.} = \frac{1}{N_{\text{labels}}} \sum_{i=1}^{N} \left(\frac{A_{i,m(i)} - A_{i,s(i)}}{M_i} \right) \tag{S65}$$

$$\text{for} \quad m(i) = \arg\max_l \left(A_{i,l}\right) \tag{S66}$$

$$\text{for} \quad s(i) = \arg\max_{k \neq m(i)} \left(A_{i,k}\right), \tag{S67}$$

where $A_{i,j}$ is some sort of sensitivity measure of latent variable z_j with respect to the generating factor w_i and M_i is a normalization constant, ensuring the summands fall into the interval $(0, 1)$.

The recently proposed Mutual Information Gap (MIG) [7] uses the Mutual Information as a measure of how the latent variables depend on the generating factors. For the normalisation, the entropy of the generating factor is used.

$$A_{i,j} = \mathrm{MI}(w_i, z_j) \tag{S68}$$
$$M_i = \mathrm{H}(w_i) \tag{S69}$$

For discrete generating factors $\{w_i\}$, the normalization with the entropy $H(w_i)$, binds the MIG to the $(0, 1)$ interval, as expected. For continuous generating factors on the other side, this does not hold. In fact, differential entropy can be zero or even negative and no good normalization is possible.

To treat this shortcoming, we introduce the slightly modified *Disentanglement score* such that it comprises continuous and discrete variables alike. Rather than using mutual information measurements, we employ powerful nonlinear regressors and classifiers for the two different classes of latent variables. The predictability of a generating factor from a given latent coordinate indirectly reflects how much information the two share.

Accordingly, we define the Disentanglement score as in Equation S65 by defining $A_{i,j}$ as the prediction performance of the regressor/classifier for predicting generating factor w_i from the latent coordinate z_j. The normalization factor is then the performance of the best constant classifier/regressor. In case of regression with mean square error, this is simply the standard deviation of the generative factor.

More precisely,

$$A_{i,j} = \begin{cases} \sqrt{\mathrm{var}(w_i)} - \sqrt{\mathrm{mse}_{z_j \to w_i}}, & \text{for regression} \\ \mathrm{accuracy}_{z_j \to w_i}, & \text{for classification} \end{cases} \tag{S70}$$

and

$$M_i = \begin{cases} \sqrt{\mathrm{var}(w_i)}, & \text{for regression.} \\ \mathrm{accuracy}^{\mathrm{const}}_{z_j \to w_i}, & \text{for classification.} \end{cases} \tag{S71}$$

We used the SciPy implementation of a k-nearest-neighbors classifier and regressor with default settings (e.g. $k = 5$) to measure the Disentanglement Score. The regressor/classifier was trained on 80% of the test data and evaluated on the remaining 20%.

8.2 DtO via Integer Programming

The *Distance to Orthogonality* (DtO) describes the Frobenius norm of the difference between a matrix V and its closest signed permutation matrix $P(V)$. Using mixed-integer linear programming (MILP) formulation, we find the closest permutation matrix as the optimum P^* of the following optimization problem

$$\begin{aligned} \min_P \sum_{i,j} |V_{i,j} - P_{i,j}| & & & \text{(S72)} \\ \text{s.t.} \quad P_{i,j} \in \{-1, 0, 1\} & & \forall\,(i,j) & \\ \sum_i |P_{i,j}| = 1 & & \forall\, j & \\ \sum_j |P_{i,j}| = 1 & & \forall\, i. & \end{aligned}$$

Producing a clean MILP formulation, with purely linear objective and binary integer values, can be achieved with a standard technique; introducing new variables. In particular, we set

$$\begin{aligned} P_{i,j} &= P^+_{i,j} - P^-_{i,j} & \text{(S73)} \\ \text{for} \quad P^+_{i,j}, P^-_{i,j} &\in \{0, 1\} \quad \forall\,(i,j) \end{aligned}$$

and introduce (continuous) variables for the differences $V_{i,j} - P_{i,j}$

$$\begin{aligned} V_{i,j} - P_{i,j} \le D_{i,j} & & \forall\,(i,j) & \quad \text{(S74)} \\ P_{i,j} - V_{i,j} \le D_{i,j} & & \forall\,(i,j). & \end{aligned}$$

The final formulation then is

Table S3: Overview over the used datasets and network architectures. The nonlinearities are only applied in the hidden layers. Biases are used for all datasets.

	Optimizer (LR)	**Architecture**	**Latent Dim.**	**Epochs**	β
dSprites	AdaGrad (10^{-2})	**Enc:** $1200 - 1200$ (Relu)	5	50	4
		Dec: $1200 - 1200 - 1200$ (Tanh)			
Synth. Lin.	Adam (10^{-3})	**Enc:** No hidden Layers (Lin)	2	600	10^{-4}
		Dec: No hidden Layers (Lin)			
Synth. Non-Lin.	Adam (10^{-3})	**Enc:** $60 - 40 - 20$ (Tanh)	2	600	10^{-3}
		Dec: $60 - 40 - 20$ (Tanh)			
MNIST	AdaGrad (10^{-2})	**Enc:** 400 (Relu)	6	400	1
		Dec: $500 - 500$ (Tanh)			
fMNIST	AdaGrad (10^{-2})	**Enc:** 400 (Relu)	6	500	1
		Dec: $500 - 500$ (Tanh)			

$$\min_{P} \sum_{i,j} D_{i,j} \tag{S75}$$

$$\begin{aligned} \text{s.t.} \quad & (P^+_{i,j} - P^-_{i,j}) - V_{i,j} \leq D_{i,j} && \forall\,(i,j) \\ & V_{i,j} - (P^+_{i,j} - P^-_{i,j}) \leq D_{i,j} && \forall\,(i,j) \\ & \sum_i \left(P^+_{i,j} + P^-_{i,j}\right) = 1 && \forall\, j \\ & \sum_j \left(P^+_{i,j} + P^-_{i,j}\right) = 1 && \forall\, i. \end{aligned}$$

8.3 β-VAE with Full Covariance Matrix

In the derivation of the VAE loss function, the approximate posterior is set to be a multivariate normal distribution with a diagonal covariance matrix. The claim of the paper is that this diagonalization is responsible for the orthogonalization. As one of the control experiments in Section 5 we also implemented VAE with a full covariance matrix.

Two issues now need to be addressed; computing KL divergence in closed form and adapting the reparametrization trick. Regarding the former, the sought identity is

$$D_{\mathrm{KL}}(\mathcal{N}(\mu,\Sigma) \parallel \mathcal{N}(0,\mathcal{I}_k)) = \tag{S76}$$

$$\frac{1}{2}\left(\|\mu\|^2 + \operatorname{tr}(\Sigma) - \log\det\Sigma - k\right). \tag{S77}$$

As for the reparametrization trick, if $\varepsilon \sim \mathcal{N}(0,\mathcal{I}_k)$, it is easy to check that

$$\mu + \Sigma^{1/2}\varepsilon \sim \mathcal{N}(\mu,\Sigma), \tag{S78}$$

where $\Sigma = \Sigma^{1/2} \cdot \left(\Sigma^{1/2}\right)^\top$ is the unique Cholesky decomposition of the positive definite matrix Σ.

8.4 Network Details and Training

Table S3 contains the training parameters used for the different architectures. The listed latent dimension is chosen to be the number of independent generating factors, if applicable, and chosen large enough to ensure decent reconstruction loss on all architectures.

All reported numbers are calculated using a previously unseen test dataset. To facilitate this, we split the whole datasets randomly into three parts for training, evaluation and test (containing $80\,\%$, $10\,\%$ and $10\,\%$ of all samples respectively). During development, we use the evaluation dataset, for the final reports we use the test dataset.

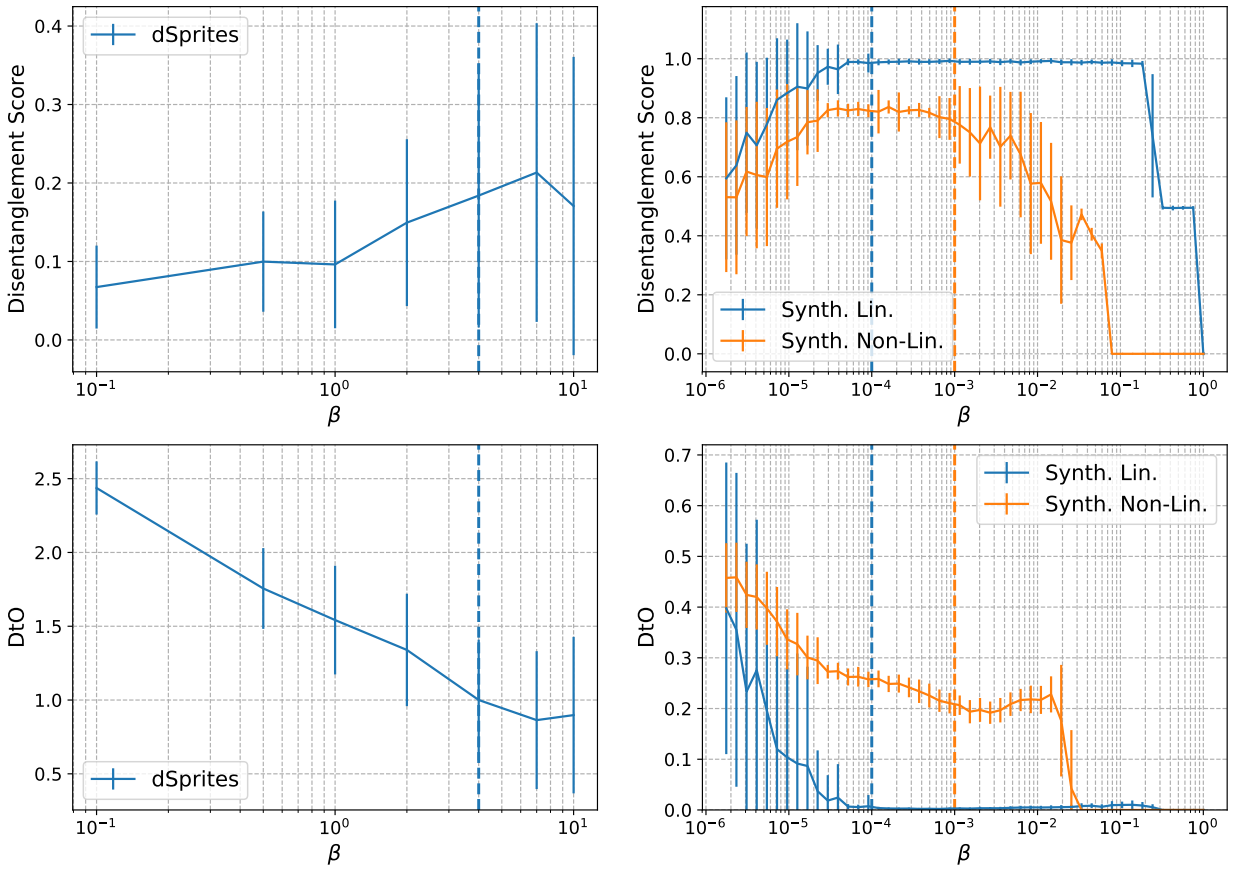

Figure S4: The β hyper-parameter in the β-VAE allows to trade-off reconstruction error and the KL loss such that the desired amount of disentanglement is achieved. The plots show the Disentanglement Score (top) and the DtO (bottom) for dSprites (left) and synthetic datasets (right). The dashed lines indicate the parameter chosen for the experiments.

8.5 Synthetic Datasets

The linear synthetic dataset is generated with a transformation $f_{\text{lin}}\colon \mathbb{R}^2 \to \mathbb{R}^3$, mapping a unit square $V = [0,1]^2$ to a 3-dimensional space. The transformation can be decomposed into:

1. stretching along one axis by a fixed factor of 2,
2. trivial embedding into $\mathbb{R}^3$,
3. rotation of $45°$ along the line containing the vector $(1,-1,1)$.

For the non-linear dataset, the transformation $f_{\text{non-lin}}\colon \mathbb{R}^2 \to \mathbb{R}^6$ is realized by a random initialization of a MLP with one hidden layer (width 10), biases and $\tanh$ nonlinearitites. Both datasets consist of 50000 samples.

Table S4: Overview of Disentanglement Score and DtO for different ratios of importance between the generating factors for the Synth. Lin. task. A ratio of 1.2 means one generating factor is scaled by 1.2.

Ratio	1.0	1.2	1.5
Disent.	0.51 ± 0.28	0.76 ± 0.25	0.98 ± 0.06
DtO	0.49 ± 0.32	0.20 ± 0.24	0.01 ± 0.06

9 Additional Experiments

9.1 Dependence of Disentanglement Score and DtO on β

The choice of β depends on the achievable Disentanglement Score. Figure S4 shows a more thorough analysis of the dependence of both the Disentanglement Score and the DtO. For too small values of β, the effect of the KL term (and thus the orthogonalization) is negligible. In the other extreme case, too large values values of β result in overpruning, such that the number of active latent coordinates drops below the number of generating factors.

9.2 Degenerate case

Proposition 1 insists that the locally linearized decoder have distinct singular values, otherwise orthogonality of the column vectors does not translate into preserving axes. Here, we design an experiment showing, that this condition is also relevant in practice.

The dataset in question will be a version of the linear synthetic task where the generating factors have the same scaling, as visualized in the upper plot of Figure S5. Note that any linear encoder applying a simple rotation has both orthogonal columns and equal singular values. But it does not respect the alignment of the original square, as it does not meet the assumptions of Proposition 1.

Behavior of the β-VAE with a linear encoder/decoder network is consistent with this. The bottom part of Figure S5 shows β-VAE latent representations of four random restarts; they expose random alignments. The same effect results in high variances for both the Disentanglement Score and the DtO, as shown in Table S4.

This degeneracy also occurs for PCA. It is easy to check that *any* projection of a unit square on a line *has equal*

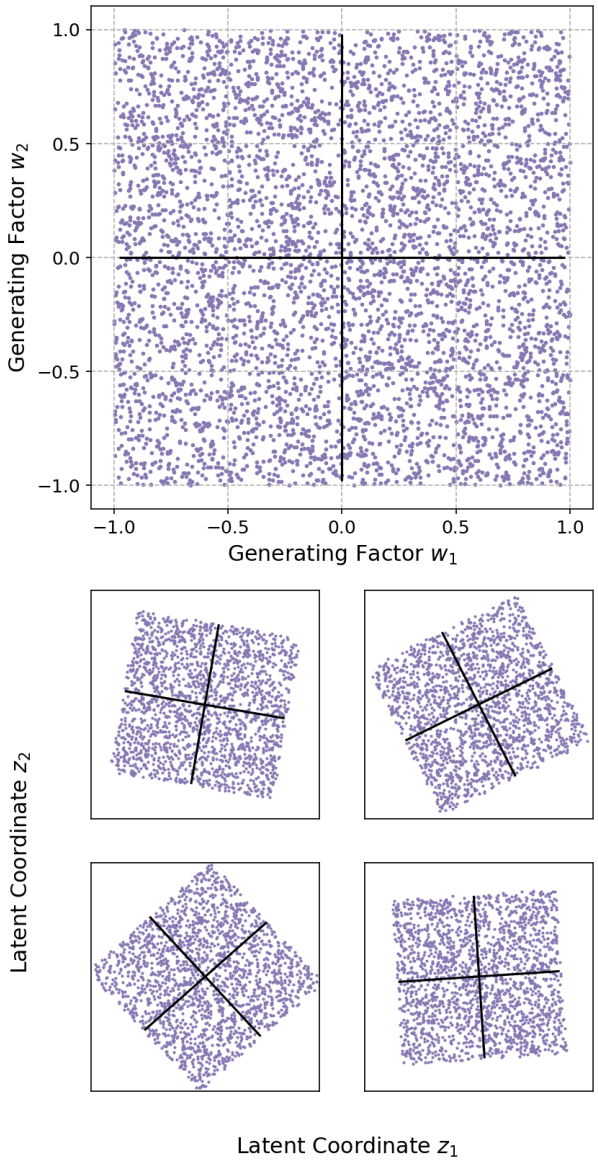

Figure S5: For strong degeneracy, e.g. in the synthetic dataset with the two generating factors w_1 and w_2 on equal, uniform scale (top), the linear β-VAE generates arbitrarily rotated latent representations (bottom) here for the linear synthetic dataset.

variance. Hence the greedy PCA algorithm has no preference over which alignment to choose, and the practical choice of alignment is implementation dependent.

This insight reinforces our point that β-VAE (just like PCA) looks for sources of variance rather than for statistical independence.

We can also see in Table S4, that the degeneracy disappears even for small rescaling of the ground truth factors. Since β-VAE promotes normalized latent representations (zero mean, unit variance), the singular values will no longer be equal and the right alignment is found. The same is true for PCA.