
Explicit Disentanglement of Appearance and Perspective in Generative Models

Nicki S. Detlefsen *
nsde@dtu.dk

Søren Hauberg *
sohau@dtu.dk

Abstract

Disentangled representation learning finds compact, independent and easy-to-interpret factors of the data. Learning such has been shown to require an inductive bias, which we explicitly encode in a generative model of images. Specifically, we propose a model with two latent spaces: one that represents spatial transformations of the input data, and another that represents the transformed data. We find that the latter naturally captures the intrinsic appearance of the data. To realize the generative model, we propose a Variationally Inferred Transformational Autoencoder (VITAE) that incorporates a spatial transformer into a variational autoencoder. We show how to perform inference in the model efficiently by carefully designing the encoders and restricting the transformation class to be diffeomorphic. Empirically, our model separates the visual style from digit type on MNIST, separates shape and pose in images of human bodies and facial features from facial shape on CelebA.

1 Introduction

Disentangled Representation Learning (DRL) is a fundamental challenge in machine learning that is currently seeing a renaissance within deep generative models. DRL approaches assume that an AI agent can benefit from separating out (disentangle) the underlying structure of data into disjointed parts of its representation. This can furthermore help interpretability of the decisions of the AI agent and thereby make them more accountable.

Even though there have been attempts to find a single formalized notion of disentanglement [Higgins et al., 2018], no such theory exists (yet) which is widely accepted. However, the intuition is that a disentangled representation \mathbf{z} should separate different informative factors of variation in the data [Bengio et al., 2012]. This means that changing a single latent dimension z_i should only change a single interpretable feature in the data space \mathcal{X} .

Within the DRL literature, there are two main approaches. The first is to hard-wire disentanglement into the model, thereby creating an inductive bias. This is well known *e.g.* in convolutional neural networks, where the convolution operator creates an inductive bias towards translation in data. The second approach is to instead learn a representation that is faithful to the underlying data structure, hoping that this is sufficient to disentangle the representation. However, there is currently little to no agreement in the literature on how to learn such representations [Locatello et al., 2019].

We consider disentanglement of two explicit groups of factors, the *appearance* and the *perspective*. We here define the appearance as being the factors of data that are left after transforming \mathbf{x} by its perspective. Thus, the appearance is the *form* or *archetype* of an object and the perspective represents the specific realization of that archetype. Practically speaking, the perspective could correspond to an image rotation that is deemed irrelevant, while the appearance is a representation of the rotated image, which is then invariant to the perspective. This interpretation of the world goes back to Plato’s allegory of the cave, from which we also borrow our terminology. This notion of removing

*Section for Cognitive Systems, Technical University of Denmark

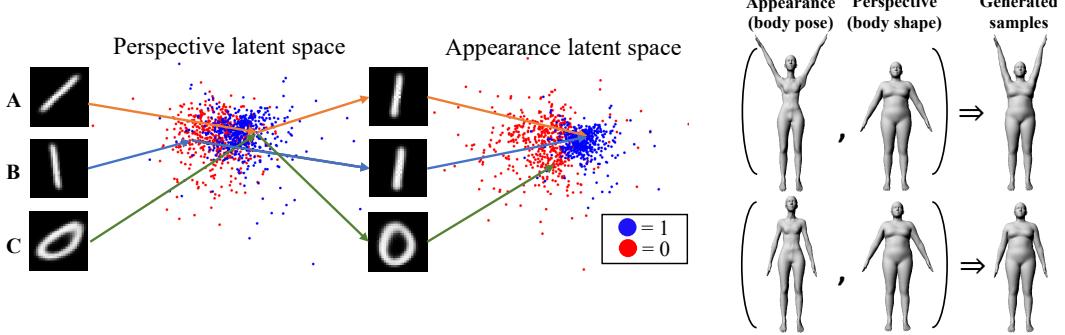


Figure 1: We disentangle data into *appearance* and *perspective* factors. First, data are encoded based on their *perspective* (in this case image A and C are rotated in the same way), which is then removed from the original input. Hereafter, the transformed samples can be encoded in the *appearance* space (image A and B are both ones), that encodes the factors left in data.

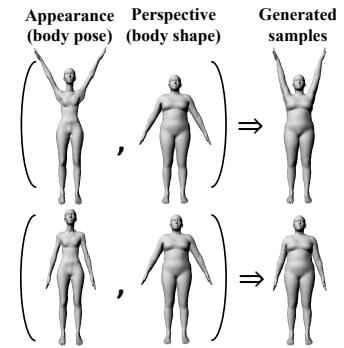


Figure 2: Our model, VITAE, disentangles appearance from perspective. Here we separate body pose (arm position) from body shape.

perspective before looking at the appearance is well-studied within supervised learning, *e.g.* using *spatial transformer nets (STNs)* [Jaderberg et al., 2015].

This paper contributes an explicit model for disentanglement of appearance and perspective in images, called the *variational inferred transformational autoencoder (VITAE)*. As the name suggests, we focus on variational autoencoders as generative models, but the idea is general (Fig. 1). First we encode/decode the perspective features in order to extract an appearance that is perspective-invariant. This is then encoded into a second latent space, where inputs with similar appearance are encoded similarly. This process generates an inductive bias that disentangles perspective and appearance. In practice, we develop an architecture that leverages the inference part of the model to guide the generator towards better disentanglement. We also show that this specific choice of architecture improves training stability with the right choice of parametrization of perspective factors. Experimentally, we demonstrate that our model on four datasets: standard disentanglement benchmark dSprites, disentanglement of style and content on MNIST, pose and shape on images of human bodies (Fig. 2) and facial features and facial shape on CelebA.

2 Related work

Disentangled representations learning (DRL) have long been a goal in data analysis. Early work on *non-negative matrix factorization* [Lee and Seung, 1999] and *bilinear models* [Tenenbaum and Freeman, 2000] showed how images can be composed into semantic “parts” that can be glued together to form the final image. Similarly, *EigenFaces* [Turk and Pentland, 1991] have often been used to factor out lighting conditions from the representation [Shakunaga and Shigenari, 2001], thereby discovering some of the physics that govern the world of which the data is a glimpse. This is central in the long-standing argument that for an AI agent to understand and reason about the world, it must disentangle the explanatory factors of variation in data [Lake et al., 2016]. As such, DRL can be seen as a poor man’s approximation to discovering the underlying causal factors of the data.

Independent components are, perhaps, the most stringent formalization of “disentanglement”. The seminal *independent component analysis (ICA)* [Comon, 1994] factors the signal into statistically independent components. It has been shown that the independent components of *natural images* are edge filters [Bell and Sejnowski, 1997] that can be linked to the receptive fields in the human brain [Olshausen and Field, 1996]. Similar findings have been made for both *video* and *audio* [van Hateren and Ruderman, 1998, Lewicki, 2002]. DRL, thus, allows us to understand both the data and ourselves. Since independent factors are the optimal compression, ICA finds the most compact representation, implying that the predictive model can achieve maximal capacity from its parameters. This gives DLR a predictive perspective, and can be taken as a hint that a well-trained model might be disentangled. In

the linear case, independent components have many successful realizations [Hyvärinen and Oja, 2000], but in the general non-linear case, the problem is not identifiable [Hyvärinen et al., 2018].

Deep DRL was initiated by Bengio et al. [2012] who sparked the current interest in the topic. One of the current state-of-the-art methods for doing disentangled representation learning is the β -VAE [Higgins et al., 2017], that modifies the *variational autoencoder* (VAE) [Kingma and Welling, 2013, Rezende et al., 2014] to learn a more disentangled representation. β -VAE enforces more weight on the KL-divergence in the VAE loss, thereby optimizing towards latent factors that should be axis aligned *i.e.* disentangled. Newer models like β -TCVAE [Chen et al., 2018] and DIP-VAE [Kumar et al., 2017] extend β -VAE by decomposing the KL-divergences into multiple terms, and only increase the weight on terms that analytically disentangles the models. InfoGAN [Chen et al., 2016] extends the latent code z of the standard GAN model [Goodfellow et al., 2014] with an extra latent code c and then penalize low mutual information between generated samples $G(c, z)$ and c . DC-IGN [Kulkarni et al., 2015] forces the latent codes to be disentangled by only feeding in batches of data that vary in one way (*e.g.* pose, light) while only having small disjoint parts of the latent code active.

Shape statistics is the key inspiration for our work. The shape of an object was first formalized by Kendall [1989] as being what is left of an object when *translation*, *rotation* and *scale* are factored out. That is, the intrinsic shape of an object should not depend on viewpoint. This idea dates, at least, back to D’Arcy Thompson [1917] who pioneered the understanding of the development of biological forms. In Kendall’s formalism, the rigid transformations (translation, rotation and scale) are viewed as group actions to be factored out of the representation, such that the remainder is *shape*. Higgins et al. [2018] follow the same idea by defining disentanglement as a factoring of the representation into group actions. Our work can be seen as a realization of this principle within a deep generative model. When an object is represented by a set of landmarks, *e.g.* in the form of discrete points along its contour, then Kendall’s *shape space* is a Riemannian manifold that exactly captures all variability among the landmarks except translation, rotation, and scale of the object. When the object is not represented by landmarks, then similar mathematical results are not available. Our work shows how the same idea can be realized for general image data, and for a much wider range of transformations than the rigid ones. Learned-Miller [2006] proposed a related linear model that generate new data by transforming a prototype, which is estimated by joint alignment.

Transformations are at the core of our method, and these leverage the architecture of spatial transformer nets (STNs) [Jaderberg et al., 2015]. While these work well within supervised learning, [Lin and Lucey, 2016, Annunziata et al., 2018, Detlefsen et al., 2018] there has been limited uptake within generative models. Lin et al. [2018] combine a GAN with an STN to compose a foreground (*e.g* a furniture) into a background such that it look neutral. The AIR model [Eslami et al., 2016] combines STNs with a VAE for object rendering, but do not seek disentangled representations. In supervised learning, *data augmentation* is often used to make a classifier partially invariant to select transformations [Baird, 1992, Hauberg et al., 2016].

3 Method

Our goal is to extend a variational autoencoder (VAE) [Kingma and Welling, 2013, Rezende et al., 2014] such that it can disentangle appearance and perspective in data. A standard VAE assumes that data is generated by a set of latent variables following a standard Gaussian prior,

$$p(\mathbf{x}) = \int p(\mathbf{x}|z)p(z)dz \quad (1)$$

$$p(z) = \mathcal{N}(\mathbf{0}, \mathbb{I}_d), \quad p(\mathbf{x}|z) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_p(z), \boldsymbol{\sigma}_p^2(z)) \text{ or } P(\mathbf{x}|z) = \mathcal{B}(\mathbf{x}|\boldsymbol{\mu}_p(z)).$$

Data \mathbf{x} is then generated by first sampling a latent variable z and then sample \mathbf{x} from the conditional $p(\mathbf{x}|z)$ (often called the decoder). To make the model flexible enough to capture complex data distributions, $\boldsymbol{\mu}_p$ and $\boldsymbol{\sigma}_p^2$ are modeled as deep neural nets. The marginal likelihood is then intractable and a variational approximation q to $p(z|\mathbf{x})$ is needed,

$$p(z|\mathbf{x}) \approx q(z|\mathbf{x}) = \mathcal{N}(z|\boldsymbol{\mu}_q(\mathbf{x}), \boldsymbol{\sigma}_q^2(\mathbf{x})), \quad (2)$$

where $\boldsymbol{\mu}_q(\mathbf{x})$ and $\boldsymbol{\sigma}_q^2(\mathbf{x})$ are deep neural networks, see Fig. 3(a).

When training VAEs, we therefore simultaneously train a generative model $p_\theta(\mathbf{x}|z)p_\theta(z)$ and an inference model $q_\phi(z|\mathbf{x})$ (often called the encoder). This is done by maximizing a variational lower

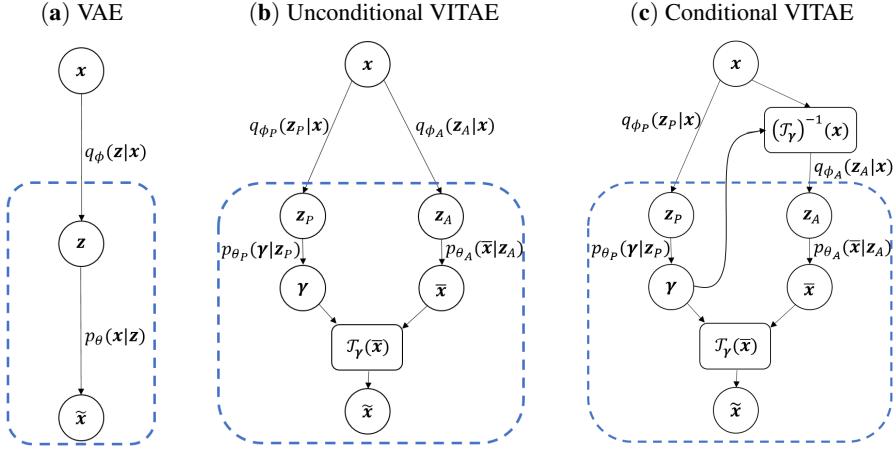


Figure 3: Architectures of standard VAE and our proposed U-VITAE and C-VITAE models. Here q denotes encoders, p denotes decoders, \mathcal{T}^γ denotes a ST-layer with transformation parameters γ . The dotted box indicates the generative model.

bound to the likelihood $p(\mathbf{x})$ called the *evidence lower bound (ELBO)*

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] = \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})]}_{\text{data fitting term}} - \underbrace{KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}))}_{\text{regularization term}}. \quad (3)$$

The first term measures the reconstruction error between \mathbf{x} and $p_\theta(\mathbf{x}|\mathbf{z})$ and the second measures the KL-divergence between the encoder $q_\phi(\mathbf{z}|\mathbf{x})$ and the prior $p(\mathbf{z})$. Eq. 3 can be optimized using the reparametrization trick [Kingma and Welling, 2013]. Several improvements to VAEs have been proposed [Burda et al., 2015, Kingma et al., 2016], but our focus is on the standard model.

3.1 Incorporating an inductive bias

To incorporate an inductive bias that is able to disentangle appearance from perspective, we change the underlying generative model to rely on two latent factors \mathbf{z}_A and \mathbf{z}_P ,

$$p(\mathbf{x}) = \int \int p(\mathbf{x}|\mathbf{z}_A, \mathbf{z}_P) p(\mathbf{z}_A) p(\mathbf{z}_P) d\mathbf{z}_A d\mathbf{z}_P, \quad (4)$$

where we assume that \mathbf{z}_A and \mathbf{z}_P both follow standard Gaussian priors. Similar to a VAE, we also model the generators as deep neural networks. To generate new data \mathbf{x} , we combine the appearance and perspective factors using the following 3-step procedure that uses a spatial transformer (ST) layer [Jaderberg et al., 2015] (dotted box in Fig. 3(b)):

1. Sample \mathbf{z}_A and \mathbf{z}_P from $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbb{I}_d)$.
2. Decode both samples $\tilde{\mathbf{x}} \sim p(\mathbf{x}|\mathbf{z}_A)$, $\gamma \sim p(\mathbf{x}|\mathbf{z}_P)$.
3. Transform $\tilde{\mathbf{x}}$ with parameters γ using a spatial transformer layer: $\mathbf{x} = \mathcal{T}_\gamma(\tilde{\mathbf{x}})$.

This process is illustrated by the dotted box in Fig. 3(b).

Unconditional VITAE inference. As the marginal likelihood (12) is intractable, we use variational inference. A natural choice is to approximate each latent group of factors $\mathbf{z}_A, \mathbf{z}_P$ independently of the other *i.e.*

$$p(\mathbf{z}_P|\mathbf{x}) \approx q_P(\mathbf{z}_P|\mathbf{x}) \text{ and } p(\mathbf{z}_A|\mathbf{x}) \approx q_A(\mathbf{z}_A|\mathbf{x}). \quad (5)$$

The combined inference and generative model is illustrated in Fig. 3(b). For comparison, a VAE model is shown in Fig. 3(a). It can easily be shown that the ELBO for this model is merely a VAE with a KL-term for each latent space (see supplements).

Conditional VITAE inference. This inference model does *not* mimic the generative process of the model, which may be suboptimal. Intuitively, we expect the encoder to approximately perform the inverse operation of the decoder, *i.e.* $\mathbf{z} \approx \text{encoder}(\text{decoder}(\mathbf{z})) \approx \text{decoder}^{-1}(\text{decoder}(\mathbf{z}))$. Since the proposed encoder (5) does not include an ST-layer, it may be difficult to train an encoder to approximately invert the decoder. To accommodate this, we first include an ST-layer in the encoder for the appearance factors. Secondly, we explicitly enforce that the predicted transformation in the encoder \mathcal{T}^{γ_e} is the inverse of that of the decoder \mathcal{T}^{γ_d} , *i.e.* $\mathcal{T}^{\gamma_e} = (\mathcal{T}^{\gamma_d})^{-1}$ (more on invertibility in Sec. 3.2). The inference of appearance is now dependent on the perspective factor \mathbf{z}_P , *i.e.*

$$p(\mathbf{z}_P|\mathbf{x}) \approx q_P(\mathbf{z}_P|\mathbf{x}) \text{ and } p(\mathbf{z}_A|\mathbf{x}) \approx q_A(\mathbf{z}_A|\mathbf{x}, \mathbf{z}_P). \quad (6)$$

These changes to the inference architecture are illustrated in Fig. 3(c). It can easily be shown that the ELBO for this model is given by

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_A, q_P} [\log(p(\mathbf{x}|\mathbf{z}_A, \mathbf{z}_P)) - D_{KL}(q_P(\mathbf{z}_P|\mathbf{x})||p(\mathbf{z}_P))] - \mathbb{E}_{q_P} [D_{KL}(q_A(\mathbf{z}_A|\mathbf{x})||p(\mathbf{z}_A))]. \quad (7)$$

which resembles the standard ELBO with a additional term (derivation in supplementary material), corresponding to the second latent space. We will call both models *variational inferred transformational autoencoders (VITAE)* and we will denote the first model (5) as *unconditional/U-VITAE* and the second model (12) as *conditional/C-VITAE*. The naming comes from Eq. 5 and 12, where \mathbf{z}_A is respectively unconditioned and conditioned on \mathbf{z}_P . Experiments will show that the conditional architecture is essential for inference (Sec. 4.2).

3.2 Transformation classes

Until now, we have assumed that there exists a class of transformations \mathcal{T} that captures the perspective factors in data. Clearly, the choice of \mathcal{T} depends on the true factors underlying the data, but in many cases an affine transformation should suffice.

$$\mathcal{T}_\gamma(\mathbf{x}) = \mathbf{Ax} + \mathbf{b} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{14} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ 1 \end{bmatrix}. \quad (8)$$

However, the C-VITAE model requires access to the inverse transformation \mathcal{T}^{-1} . The inverse of Eq. 8 is given by $\mathcal{T}_\gamma^{-1}(\mathbf{x}) = \mathbf{A}^{-1}\mathbf{x} - \mathbf{b}$, which only exist if \mathbf{A} has a non-zero determinant.

One, easily verified, approach to secure invertibility is to parametrize the transformation by two scale factors s_x, s_y , one rotation angle α , one shear parameter m and two translation parameters t_x, t_y :

$$\mathcal{T}_\gamma(\mathbf{x}) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} t_x \\ t_y \end{bmatrix}. \quad (9)$$

In this case the inverse is trivially

$$\mathcal{T}_{(s_x, s_y, \gamma, m, t_x, t_y)}^{-1}(\mathbf{x}) = \mathcal{T}_{(\frac{1}{s_x}, \frac{1}{s_y}, -\gamma, -m, -t_x, -t_y)}(\mathbf{x}), \quad (10)$$

where the scale factors must be strictly positive.

An easier and more elegant approach is to leverage the matrix exponential. That is, instead of parametrizing the transformation in Eq. 8, we instead parametrize the velocity of the transformation

$$\mathcal{T}_\gamma(\mathbf{x}) = \text{expm} \left(\begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{14} \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ 1 \end{bmatrix}. \quad (11)$$

The inverse² is then $\mathcal{T}_\gamma^{-1} = \mathcal{T}_{-\gamma}$. Then \mathcal{T} in Eq. 11 is a C^∞ -diffeomorphism (*i.e.* a differentiable invertible map with a differentiable inverse) [Duistermaat and Kolk, 2000]. Experiments show that diffeomorphic transformations stabilize training and yield tighter ELBOs (see supplements).

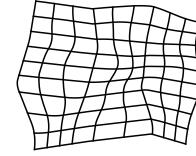
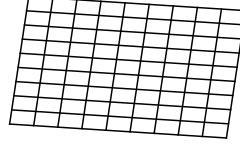


Figure 4: Random deformation field of an affine transformation (top) compared to a CPAB (bottom). We clearly see that CPAB transformations offers a much more flexible and rich class of diffeomorphic transformations.

²Follows from \mathcal{T}_γ and $\mathcal{T}_{-\gamma}$ being commuting matrices.

Often we will not have prior knowledge regarding which transformation classes are suitable for disentangling the data. A natural way forward is then to apply a highly flexible class of transformations that are treated as “black-box”. Inspired by Detlefsen et al. [2018], we also consider transformations \mathcal{T}_γ using the highly expressive diffeomorphic transformations *CPAB* from Freifeld et al. [2015]. These can be viewed as an extension to Eq. 11: instead of having a single affine transformation parametrized by its velocity, the image domain is divided into smaller cells, each having their own affine velocity. The collection of local affine velocities can be efficiently parametrized and integrated, giving a fast and flexible diffeomorphic transformation, see Fig. 4 for a comparison between an affine transformation and a CPAB transformation. For details, see Freifeld et al. [2015].

We note, that our transformer architecture are similar to the work of Lorenz et al. [2019] and Xing et al. [2019] in that they also tries to achieve disentanglement through spatial transformations. However, our work differ in the choice of transformation. This is key, as the theory of Higgins et al. [2018] strongly relies on disentanglement through *group actions*. This places hard constrains on which spatial transformations are allowed: *they have to form a smooth group*. Both thin-plate-spline transformations considered in Lorenz et al. [2019] and displacement fields considered in Xing et al. [2019] are not invertible and hence do not correspond to proper group actions. Since diffeomorphic transformations form a smooth group, this choice is paramount to realize the theory of Higgins et al. [2018].

4 Experimental results and discussion

For all experiments, we train a standard VAE, a β -VAE [Higgins et al., 2017], a β -TCVAE [Chen et al., 2018], a DIP-VAE-II [Kumar et al., 2017] and our developed VITAE model. We model the encoders and decoders as multilayer perceptron networks (MLPs). For a fair comparison, the number of trainable parameters is approximately the same in all models. The models were implemented in Pytorch [Paszke et al., 2017] and the code is available at <https://github.com/SkaftefNICKI/unsupervised/>.

Evaluation metric. Measuring disentanglement still seems to be an unsolved problem, but the work of Locatello et al. [2019] found that most proposed disentanglement metrics are highly correlated. We have chosen to focus on the DIC-metric from Eastwood and Williams [2019], since this metric has seen some uptake in the research community. This metric measures how well the generative factors can be predicted from latent factors. For the MNIST and SMPL datasets, the generative factors are discrete instead of continuous, so we change the standard linear regression network to a kNN-classification algorithm. We denote this metric D_{score} in the results.

4.1 Disentanglement on shapes

We initially test our models on the dSprites dataset [Matthey et al., 2017], which is a well established disentanglement benchmarking dataset to evaluate the performance of disentanglement algorithms. The results can be seen in Table 1. We find that our proposed C-VITAE model perform best, followed

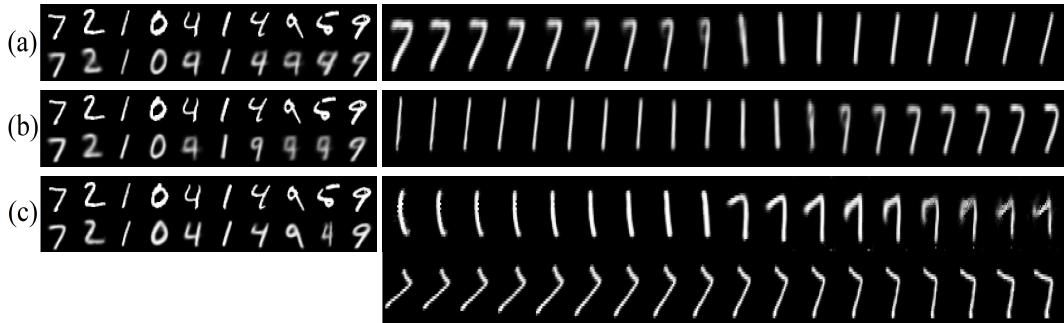


Figure 5: Reconstructions (left images) and manipulation of latent codes (right images) on MNIST for the three different models: VAE (a), β -VAE (b) and C-VITAE (c). The right images are generated by varying one latent dimension in all models, while keeping the rest fixed. For the C-VITAE model, we have shown this for both the appearance and perspective spaces.

	dSprite			MNIST			SMPL		
	ELBO	$\log p(\mathbf{x})$	D_{score}	ELBO	$\log p(\mathbf{x})$	D_{score}	ELBO	$\log p(\mathbf{x})$	D_{score}
VAE	-47.05	-49.32	0.05	-169	-172	0.579	-8.62×10^3	-8.62×10^3	0.485
β -VAE	-79.45	-81.38	0.18	-150	-152	0.653	-8.62×10^3	-8.60×10^3	0.525
β -TCVAE	-66.48	-68.12	0.30	-141	-144	0.679	-8.62×10^3	-8.56×10^3	0.651
DIP-VAE-II	-46.32	-48.92	0.12	-140	-155	0.733	-8.62×10^3	-8.54×10^3	0.743
U-VITAE	-55.25	-57.29	0.22	-142	-143	0.782	-8.62×10^3	-8.55×10^3	0.673
C-VITAE	-68.26	-70.49	0.38	-139	-141	0.884	-8.62×10^3	-8.52×10^3	0.943

Table 1: Quantitative results on three datasets. For each dataset we report the ELBO, test set log likelihood and disentanglement score D_{score} . Bold marks best results.

by the β -TCVAE model in terms of disentanglement. The experiments clearly shows the effect on performance of the improved inference structure of C-VITAE compared to U-VITAE. It can be shown that the conditional architecture of C-VITAE, minimizes the mutual information between z_A and z_P , leading to better disentanglement of the two latent spaces. To get the U-VITAE architecture to work similarly would require a auxiliary loss term added to the ELBO.

4.2 Disentanglement of MNIST images

Secondly, we test our model on the MNIST dataset [LeCun et al., 1998]. To make the task more difficult, we artificially augment the dataset by first randomly rotating each image by an angle uniformly chosen in the interval $[-20^\circ, 20^\circ]$ and secondly translating the images by $t = [x, y]$, where x, y is uniformly chosen from the interval $[-3, 3]$. For VITAE, we model the perspective with an affine diffiomorphic transformation (Eq. 11).

The quantitative results can be seen in Table 1. We clearly see that C-VITAE outperforms the alternatives on all measures. We overall observes that better disentanglement, seems to give better distribution fitting. Qualitatively, Fig. 5 shows the effect of manipulating the latent codes alongside test reconstructions for VAE, β -VAE and C-VITAE. Due to space constraints, the results from β -TCVAE and DIP-VAE-II can found in the supplementary material. The plots were generated by following the protocol from Higgins et al. [2017]: one latent factor is linearly increased from -3 to 3, while the rest is kept fixed. In the VAE (Fig. 5(a)), this changes both the appearance (going from a 7 to a 1) and the perspective (going from rotated slightly left to rotated right). We see no meaningful disentanglement of latent factors. In the β -VAE model (Fig. 5(b)), we observe some disentanglement, since only the appearance changes with the latent factor. However this disentanglement comes at the cost of poor reconstructions. This trade-off is directly linked to the emphasized regularization in the β -VAE. We note that the value $\beta = 4.0$ proposed in the original paper [Higgins et al., 2017] is insufficiently low for our experiments to observe any disentanglement, and we use $\beta = 8.0$ based on qualitative evaluation of results. For β -TCVAE and DIP-VAE-II we observe nearly the same amount of qualitative disentanglement as β -VAE, however these models achieve less blurred samples and reconstructions. This is probably due to the two models decomposition of the KL-term, only increasing the parts that actually contributes to disentanglement. Finally, for our developed VITAE model (Fig. 5(c)), we clearly see that when we change the latent code in the appearance space (top row), we only change the content of the generated images, while manipulating the latent code in the perspective space (bottom row) only changes the perspective *i.e.* image orientation.

Interestingly, we observe that there exists more than one prototype of a 1 in the appearance space of VITAE, going from slightly bent to straightened out. By our definition of disentanglement, that *everything left* after transforming the image is appearance, there is nothing wrong with this. This is simply a consequence of using an affine transformation that cannot model this kind of local deformation. Choosing a more flexible transformation class could factor out this kind of perspective. The supplements contain generated samples from the different models.

4.3 Disentanglement of body shape and pose

We now consider synthetic image data of human bodies generated by the *Skinned Multi-Person Linear Model (SMPL)* [Loper et al., 2015] which are explicitly factored into *shape* and *pose*. We generate 10,000 bodies (8,000 for training, 2,000 for testing), by first continuously sampling body shape (going from thin to thick) and then uniformly sampling a body pose from four categories ((arms up, tight),

(arms up, wide), (arms down, tight), (arms down, wide)). Fig. 2 shows examples of generated images. Since change in body shape approximately amounts to a local shape deformation, we model the perspective factors using the aforementioned "black-box" diffeomorphic CPAB transformations (Sec. 3.2). The remaining appearance factor should then reflect body pose.

Quantitative evaluation. We again refer to Table 1 that shows ELBO, test set log-likelihood and disentanglement score for all models. As before, C-VITAE is both better at modelling the data distribution and achieves a higher disentanglement score. The explanation is that for a standard VAE model (or β -VAE and its variants for that sake) to learn a complex body shape deformation model, it requires a high capacity network. However, the VITAE architecture gives the autoencoder a short-cut to learning these transformations that only requires optimizing a few parameters. We are not guaranteed that the model will learn anything meaningful or that it actually uses this short-cut, but experimental evidence points in that direction. A similar argument holds in the case of MNIST, where a standard MLP may struggle to learn rotation of digits, but the ST-layer in the VITAE architecture provides a short-cut. Furthermore, we found the training of VITAE to be more stable than other models.

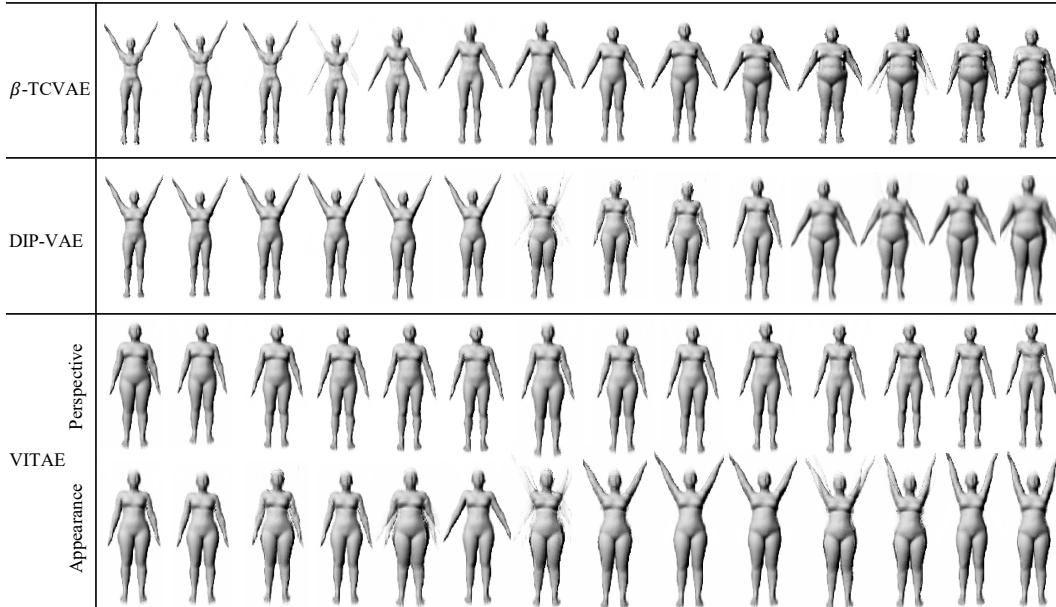


Figure 6: Disentanglement of body shape and body pose on SMPL-generated bodies for three different models. The images are generated by varying one latent dimension, while keeping the rest fixed. For the C-VITAE model we have shown this for both the appearance and perspective spaces, since this is the only model where we quantitatively observe disentanglement.

Qualitative evaluation. Again, we manipulate the latent codes to visualize their effect (Fig. 14). This time, we here show the result for β -TCVAE, DIP-VAE-II and VITAE. The results from standard VAE and β -VAE can be found in supplementary material. For both β -TCVAE and DIP-VAE-II we do not observe disentanglement of body pose and shape, since the decoded images both change arm position (from up to down) and body shape. We note that for both β -VAE, β -TCVAE and DIP-VAE-II we did a grid search for their respective hyper parameters. For these three models, we observe that the choice of hyper parameters (scaling of KL term) can have detrimental impact of reconstructions and generated samples. Due to lack of space, test set reconstructions and generated samples can be found in the supplementary material. For VITAE we observe some disentanglement of body pose and shape, as variation in appearance space mostly changes the positions of the arms, while the variations in the perspective space mostly changes body shape. The fact that we cannot achieve full disentanglement of this SMPL dataset indicates the difficulty of the task.

4.4 Disentanglement on CelebA

Finally, we qualitatively evaluated our proposed model on the CelebA dataset [Liu et al., 2015]. Since this is a "real life" dataset we do not have access to generative factors and we can therefore only qualitatively evaluate the model. We again model the perspective factors using the aforementioned CPAB transformations, which we assume can model the facial shape. The results can be seen in Fig. 7, which shows latent traversals of both the perspective and appearance factors, and how they influence the generated images. We do observe some interpolation artifacts that are common for architectures using spatial transformers.

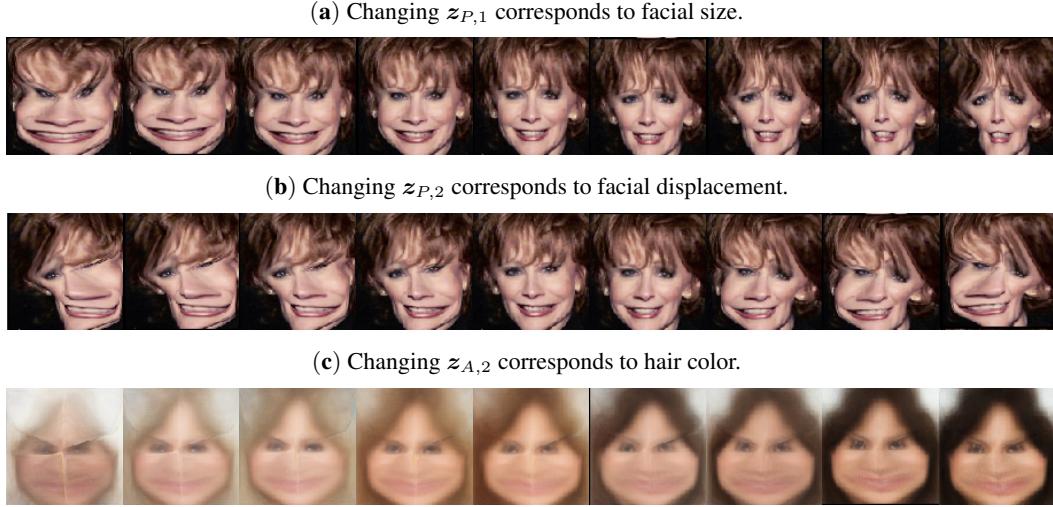


Figure 7: Traversal in latent space shows, that our model can disentangle complex factors such as facial size, facial position and hair color.

5 Summary

In this paper, we have shown how to explicitly disentangle *appearance* from *perspective* in a variational autoencoder [Kingma and Welling, 2013, Rezende et al., 2014]. This is achieved by incorporating a spatial transformer layer [Jaderberg et al., 2015] into both encoder and decoder in a coupled manner. The concepts of appearance and perspective are broad as is evident from our experimental results in human body images, where they correspond to *pose* and *shape*, respectively. By choosing the class of transformations in accordance with prior knowledge it becomes an effective tool for controlling the inductive bias needed for disentangled representation learning. On both MNIST and body images our method quantitatively and qualitatively outperforms general purpose disentanglement models [Higgins et al., 2017, Chen et al., 2018, Kumar et al., 2017]. We find it unsurprisingly that in situations where some prior knowledge about the generative factors is known, encoding these in the into the model give better result than ignoring such information.

Our results support the hypothesis [Higgins et al., 2018] that inductive biases are necessary for learning disentangled representations, and our model is a step in the direction of getting fully disentangled generative models. We envision that the VITAE model should be combined with other models, by first using the VITAE model to separate appearance and perspective, and then training a second model only on the appearance. This will factor out one latent factor at a time, leaving a hierarchy of disentangled factors.

Acknowledgements. This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement n° 757360). NSD and SH were supported in part by a research grant (15334) from VILLUM FONDEN. We gratefully acknowledge the support of NVIDIA Corporation with the donation of GPU hardware used for this research.

References

- R. Annunziata, C. Sagonas, and J. Calì. Destnet: Densely fused spatial transformer networks. *CoRR*, abs/1807.04050, 2018.
- G. Arvanitidis, L. K. Hansen, and S. Hauberg. Latent Space Oddity: on the Curvature of Deep Generative Models. *arXiv e-prints*, art. arXiv:1710.11379, Oct. 2017.
- H. S. Baird. Document image defect models. In *Structured Document Image Analysis*, pages 546–556. Springer, 1992.
- A. J. Bell and T. J. Sejnowski. The “independent components” of natural scenes are edge filters. *Vision research*, 37(23):3327–3338, 1997.
- Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. *arXiv e-prints*, art. arXiv:1206.5538, June 2012.
- Y. Burda, R. B. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *CoRR*, abs/1509.00519, 2015.
- R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud. Isolating Sources of Disentanglement in Variational Autoencoders. feb 2018. URL <http://arxiv.org/abs/1802.04942>.
- X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016.
- P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287 – 314, 1994. ISSN 0165-1684. Higher Order Statistics.
- D’Arcy Thompson. On growth and form. *On growth and form.*, 1917.
- N. S. Detlefsen, O. Freifeld, and S. Hauberg. Deep diffeomorphic transformer networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- J. Duistermaat and J. Kolk. Lie groups and lie algebras. In *Lie Groups*, pages 1–92. Springer Berlin Heidelberg, 2000.
- C. Eastwood and C. K. I. Williams. A Framework for the Quantitative Evaluation of Disentangled Representations, feb 2019. URL <https://openreview.net/forum?id=By-7dz-AZ>.
- S. M. A. Eslami, N. Heess, T. Weber, Y. Tassa, D. Szepesvari, K. Kavukcuoglu, and G. E. Hinton. Attend, Infer, Repeat: Fast Scene Understanding with Generative Models. mar 2016. doi: 10.1038/nature14236.
- O. Freifeld, S. Hauberg, K. Batmanghelich, and J. W. F. III. Highly-expressive spaces of well-behaved transformations: Keeping it simple. In *ICCV*, 2015.
- 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.
- S. Hauberg, O. Freifeld, A. B. L. Larsen, J. W. F. III, and L. K. Hansen. Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation. In *Proceedings of the 19th international Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 41, 2016.
- I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. β -vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.
- I. Higgins, D. Amos, D. Pfau, S. Racaniere, L. Matthey, D. Rezende, and A. Lerchner. Towards a Definition of Disentangled Representations. *arXiv e-prints*, art. arXiv:1812.02230, Dec. 2018.
- A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13 (4-5):411–430, 2000.
- A. Hyvärinen, H. Sasaki, and R. E. Turner. Nonlinear ICA Using Auxiliary Variables and Generalized Contrastive Learning. *arXiv e-prints*, art. arXiv:1805.08651, May 2018.
- M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. *CoRR*, abs/1506.02025, 2015.

- C. Kaae Sønderby, T. Raiko, L. Maaløe, S. Kaae Sønderby, and O. Winther. Ladder Variational Autoencoders. *ArXiv e-prints*, Feb. 2016.
- D. G. Kendall. A survey of the statistical theory of shape. *Statistical Science*, pages 87–99, 1989.
- D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- D. P. Kingma and M. Welling. Auto-Encoding Variational Bayes. *ArXiv e-prints*, Dec. 2013.
- D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improving Variational Inference with Inverse Autoregressive Flow. *arXiv e-prints*, art. arXiv:1606.04934, June 2016.
- T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum. Deep convolutional inverse graphics network. *CoRR*, abs/1503.03167, 2015.
- A. Kumar, P. Sattigeri, and A. Balakrishnan. Variational Inference of Disentangled Latent Concepts from Unlabeled Observations. nov 2017. URL <http://arxiv.org/abs/1711.00848>.
- B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *CoRR*, abs/1604.00289, 2016.
- E. G. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(2):236–250, Feb. 2006. ISSN 0162-8828.
- Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, pages 86(11):2278–2324, nov 1998.
- D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401 (6755):788, 1999.
- M. S. Lewicki. Efficient coding of natural sounds. *Nature neuroscience*, 5(4):356, 2002.
- C. Lin and S. Lucey. Inverse compositional spatial transformer networks. *CoRR*, abs/1612.03897, 2016.
- C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey. ST-GAN: Spatial Transformer Generative Adversarial Networks for Image Compositing. *ArXiv e-prints*, Mar. 2018.
- Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- F. Locatello, S. Bauer, M. Lucic, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *Proceedings of the 36th International Conference on Machine Learning*, 2019.
- M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.
- D. Lorenz, L. Bereska, T. Milbich, and B. Ommer. Unsupervised Part-Based Disentangling of Object Shape and Appearance. *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, Mar 2019.
- L. Matthey, I. Higgins, D. Hassabis, and A. Lerchner. dsprites: Disentanglement testing sprites dataset. <https://github.com/deepmind/dsprites-dataset/>, 2017.
- B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.
- A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.
- D. Rezende, S. Mohamed, and D. Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. *ArXiv e-prints*, Jan. 2014.
- T. Shakunaga and K. Shigenari. Decomposed eigenface for face recognition under various lighting conditions. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.
- J. B. Tenenbaum and W. T. Freeman. Separating style and content with bilinear models. *Neural Comput.*, 12(6):1247–1283, June 2000. ISSN 0899-7667.
- M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

J. H. van Hateren and D. L. Ruderman. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proceedings of the Royal Society of London B: Biological Sciences*, 265(1412):2315–2320, 1998.

X. Xing, R. Gao, T. Han, S.-C. Zhu, and Y. Nian Wu. Deformable Generator Network: Unsupervised Disentanglement of Appearance and Geometry. *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, Jun 2019.

A Derivation of the ELBO for C-VITAE and U-VITAE

We will here focus on deriving the ELBO for the C-VITAE, because as we will see the ELBO for the U-VITAE can easily be identified from this. For both models it hold that the generative model is given by

$$p(\mathbf{x}) = \iint p(\mathbf{x}|\mathbf{z}_A, \mathbf{z}_P) p(\mathbf{z}_A) p(\mathbf{z}_P) d\mathbf{z}_A d\mathbf{z}_P.$$

We know assume that the inference of appearance now becomes dependent on the perspective factors \mathbf{z}_P *i.e.*

$$p(\mathbf{z}_P|\mathbf{x}) \approx q_P(\mathbf{z}_P|\mathbf{x}) \text{ and } p(\mathbf{z}_A|\mathbf{x}) \approx q_A(\mathbf{z}_A|\mathbf{x}, \mathbf{z}_P).$$

as in the C-VITAE model. The log-posterior is then given by:

$$\begin{aligned} \log p(\mathbf{x}) &= \log \left(\iint p(\mathbf{x}|\mathbf{z}_A, \mathbf{z}_P) p(\mathbf{z}_A) p(\mathbf{z}_P) d\mathbf{z}_A d\mathbf{z}_P \right) \\ &= \log \left(\iint p(\mathbf{x}|\mathbf{z}_A, \mathbf{z}_P) p(\mathbf{z}_A) p(\mathbf{z}_P) \frac{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})}{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} \frac{q_P(\mathbf{z}_P|\mathbf{x})}{q_P(\mathbf{z}_P|\mathbf{x})} d\mathbf{z}_A d\mathbf{z}_P \right) \\ &= \log \left(\int \mathbb{E}_{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} \left[\frac{p(\mathbf{x}|\mathbf{z}_P, \mathbf{z}_A)p(\mathbf{z}_A)}{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} \right] p(\mathbf{z}_P) \frac{q_P(\mathbf{z}_P|\mathbf{x})}{q_P(\mathbf{z}_P|\mathbf{x})} d\mathbf{z}_P \right) \\ &= \log \mathbb{E}_{q_P(\mathbf{z}_P|\mathbf{x})} \left[\mathbb{E}_{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} \left[\frac{p(\mathbf{x}|\mathbf{z}_P, \mathbf{z}_A)p(\mathbf{z}_A)}{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} \right] \frac{p(\mathbf{z}_P)}{q_P(\mathbf{z}_P|\mathbf{x})} \right] \end{aligned}$$

By using Jensen's inequality once to exchange the outer expectation with the log gives us

$$\begin{aligned} \log p(\mathbf{x}) &\geq \mathbb{E}_{q_P(\mathbf{z}_P|\mathbf{x})} \left[\log \left(\mathbb{E}_{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} \left[\frac{p(\mathbf{x}|\mathbf{z}_P, \mathbf{z}_A)p(\mathbf{z}_A)}{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} \right] \right) + \log \left(\frac{p(\mathbf{z}_P)}{q_P(\mathbf{z}_P|\mathbf{x})} \right) \right] \\ &= \mathbb{E}_{q_P(\mathbf{z}_P|\mathbf{x})} \left[\log \left(\mathbb{E}_{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} \left[\frac{p(\mathbf{x}|\mathbf{z}_P, \mathbf{z}_A)p(\mathbf{z}_A)}{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} \right] \right) \right] - D_{KL}(q_P(\mathbf{z}_P|\mathbf{x})||p(\mathbf{z}_P)) \end{aligned}$$

Then, by using Jensen's inequality once more to exchange the log and inner expectation we get

$$\begin{aligned} \log p(\mathbf{x}) &\geq \mathbb{E}_{q_P(\mathbf{z}_P|\mathbf{x})} \left[\underbrace{\mathbb{E}_{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}_P, \mathbf{z}_A) + \log \left(\frac{p(\mathbf{z}_A)}{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} \right) \right]}_{\text{term 1}} \right] - D_{KL}(q_P(\mathbf{z}_P|\mathbf{x})||p(\mathbf{z}_P)) \\ &= \underbrace{\mathbb{E}_{q_P(\mathbf{z}_P|\mathbf{x})} [\mathbb{E}_{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} [\log p(\mathbf{x}|\mathbf{z}_P, \mathbf{z}_A)]]}_{\text{term 1}} - \underbrace{\mathbb{E}_{q_P(\mathbf{z}_P|\mathbf{x})} [D_{KL}(q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})||p(\mathbf{z}_A))]}_{\text{term 2}} - \underbrace{D_{KL}(q_P(\mathbf{z}_P|\mathbf{x})||p(\mathbf{z}_P))}_{\text{term 3}} \end{aligned}$$

Here term 1 is reconstruction term between \mathbf{x} and $p(\mathbf{x}|\mathbf{z}_A, \mathbf{z}_P)$, is the term 2 is the KL divergence for the appearance space $q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})$ and its prior $p(\mathbf{z}_A)$ and term 3 is the KL divergence for the perspective space $q_P(\mathbf{z}_P|\mathbf{x})$ and its prior $p(\mathbf{z}_P)$. Similar to how gradients are calculate in VAE's, the outer expectation in term 2 is calculated with respect to a single sample, but can also be computed with respect to multiple samples similar to the work of Burda et al. [2015].

To get the ELBO of the U-VITAE model, we make the the inference of the latent spaces independent of each other *i.e.* $q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x}) = q_A(\mathbf{z}_A|\mathbf{x})$. This get rid of the expectation in term 2 and we are left with

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q_P(\mathbf{z}_P|\mathbf{x})} [\mathbb{E}_{q_A(\mathbf{z}_A|\mathbf{z}_P, \mathbf{x})} [\log p(\mathbf{x}|\mathbf{z}_P, \mathbf{z}_A)]] - D_{KL}(q_A(\mathbf{z}_A|\mathbf{x})||p(\mathbf{z}_A)) - D_{KL}(q_P(\mathbf{z}_P|\mathbf{x})||p(\mathbf{z}_P)),$$

which is the ELBO for the U-VITAE model. The intuition behind this equation is that the U-VITAE model is just a standard VAE, where the latent space \mathbf{z} has been split into two smaller latent spaces $\mathbf{z}_P, \mathbf{z}_A$, thus this is reflected in ELBO where the KL-term is similar split into two terms.

B Implementation details for the experiments

Below we describe the network architectures in details. All models were trained using the Adam optimizer [Kingma and Ba, 2014] with fixed learning rate of 10^{-4} . For the MNIST experiments we used a batch size of 512 and trained for a 2000 epochs and for SMLP and CelebA experiments we used a batch size of 256 and trained for 5000 epochs. No early stopping was used. Similar to Kaae Sønderby et al. [2016], we use annealing/warmup for the KL-divergence by scaling the term(s) by $w = \min\left(\frac{\text{epoch}}{\text{warmup}}, 1\right)$, where the warmup parameter was set to half the number of epochs.

Details for MNIST experiments. Pixel values of the images are scaled to the interval [0,1]. Each pixel is assumed to be Bernoulli distributed. For the encoders and decoders we use multilayer perceptron networks. For the VAE, β -VAE [Higgins et al., 2017], β -TCVAE [Chen et al., 2018] and DIP-VAE [Kumar et al., 2017], we use the settings listed below. For both VITAE models, we model both encoders and both decoders with approximately half the neurons, for a fair comparison. In practice we found that the encoders/decoders of the appearance factors benefits from having a bit higher capacity than the encoders/decoders of the perspective factors.

	Layer 1	Layer 2	Layer 3
$\mu_{encoder}$	128, (LeakyReLU)	64, (LeakyReLU)	d, (Linear)
$\sigma^2_{encoder}$	128, (LeakyReLU)	64, (LeakyReLU)	d, (softplus)
$\mu_{decoder}$	64, (LeakyReLU)	64, (LeakyReLU)	D, (Sigmoid)

Table 2: Model architecture for the MNIST experiments.

Here $D = 784$ and $d = 4$ for VAE based models and $d = 2$ for VITAE based models. The numbers corresponds to the size of the layer and the parenthesis is the used activation function. For the LeakyRelu activation function we use hyper parameter $\alpha = 0.1$. We only parametrize a mean function in the decoder because we assume the output pixels are Bernoulli distributed.

Details for SMPL experiments. Images was generated using the SMPL library³. The parameters for generating the body shape was drawn from a $\mathcal{N}(0, 1.25^2)$ distribution. The parameters that controls the body pose was uniformly sampled from one out of 4 pre-specified pose configurations, see Table 3.

	Pose 1	Pose 2	Pose 3	Pose 4
Left shoulder	$-\pi/8$	$-\pi/16$	$\pi/16$	$\pi/8$
Right shoulder	$\pi/8$	$\pi/16$	$-\pi/16$	$-\pi/8$
Left arm	$-\pi/3.5$	$-\pi/3.5$	$\pi/3.5$	$\pi/3.5$
Right arm	$\pi/3.5$	$\pi/3.5$	$-\pi/3.5$	$-\pi/3.5$

Table 3: When generating synthetic bodies, we uniformly sample one of the above settings for the pose.

The resolution of each image was scaled down to (400, 200). Each pixel is assumed to be Normal distributed. For the VAE based models, we use the settings listed below. For the VITAE models we used approximately half the neurons for the encoders/decoders.

³<http://smpl.is.tue.mpg.de/>

	Layer 1	Layer 2	Layer 3
$\mu_{encoder}$	256, (LeakyReLU)	128, (LeakyReLU)	d, (Linear)
$\sigma^2_{encoder}$	256, (LeakyReLU)	128, (LeakyReLU)	d, (softplus)
$\mu_{decoder}$	128, (LeakyReLU)	256, (LeakyReLU)	D, (Linear)

Table 4: Model architecture for the SMPL experiments.

Here $D = 80.000$ and $d = 4$ for VAE based models and $d = 2$ for VITAE based models. The numbers corresponds to the size of the layer and the parenthesis is the used activation function. For the LeakyRelu activation function we use hyper parameter $\alpha = 0.1$. We only parametrize a mean in the decoder because the variance function is in general very hard to train and completely arbitrarily outside the latent manifold [Arvanitidis et al., 2017]. It was therefore fixed for all pixels in all images to $\sigma^2_{decoder} = 0.1$. For the CPAB transformations [Freifeld et al., 2015] we ran the experiments with tessellation parameters [2, 4] with zero boundary constrains and no volume preservation constrains. With these settings, we are generating perspective transformations of size 30 *i.e.* $\dim(\boldsymbol{\theta}) = 30$.

Details for CelebA experiments. We use the align and cropped version of the dataset, downloaded from the homepage⁴. Each image was then down sampled to size 128×128 , to decrease computational time. Each pixel is assumed to be Normal distributed. For this task we use a convolutional-VAE. Below is listed the configuration of the network:

	Layer 1	Layer 2	Layer 3	Layer 4
$\mu_{encoder}$	Conv(10, 5, 2, LeakyReLU)	Conv(20, 5, 2, LeakyReLU)	Conv(40, 3, 2, LeakyReLU)	Dense(2, Linear)
$\sigma^2_{encoder}$	Conv(10, 5, 2, LeakyReLU)	Conv(20, 5, 2, LeakyReLU)	Conv(40, 3, 2, LeakyReLU)	Dense(2, Softplus)
$\mu_{decoder}$	DeConv(40, 3, 2, LeakyReLU)	DeConv(20, 3, 2, LeakyReLU)	DeConv(10, 5, 2, LeakyReLU)	DeConv(3, 5, 2, Sigmoid)

Table 5: Model architecture for the CelebA experiments. Conv denotes a convolutional layer and DeConv denotes de-/transposed convolutional layers. The parameters are respective number of filters, filter size, stride and activation function.

For the CPAB transformation [Freifeld et al., 2015] we ran the experiments with tessellation parameters [4, 4] with zero boundary constrains and no volume preservation constrains. With these settings, we are generating perspective transformations of size 62.

Computational requirements. Even though VITAE has a more complicated architecture than VAE (comparing Fig. (3a) vs. (3c) in main paper) both forward and backward passes in the models have roughly the same complexity when we use affine transformations (see Table 6). Using the more complex CPAB transformations adds some penalty to the computational time.

	Forward	Backward
VAE & β -VAE	0.0016s	0.014s
β -TCVAE	0.0020s	0.016s
DIP-VAE-II	0.0025s	0.018s
C-VITAE	Affine	0.0092s
	CPAB	0.1s
		0.86s

Table 6: Forward and backwards timings for the different architectures. The experiments was conducted with an Intel Xeon E5-2620v4 CPU and Nvidia GTX TITAN X GPU.

C Stability results

In the main paper we discuss multiple ways to parameterize an affine transformation. If we choose \mathcal{T}_γ with a diffiomorphic parameterization, we have found that this also has positive positive optimization properties. Fig. 8 shows the ELBO as a function of the learning rate λ for the three different choices of affine parametrization discussed in the main paper, using our C-VITAE architecture. We clearly see that the diffeomorphic affine parametrization archives a tighter bound, and can run for much higher learning rates (faster convergence) before the network begins to diverge. These findings are similar to those of Detlefsen et al. [2018] in the supervised context.

⁴<http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>

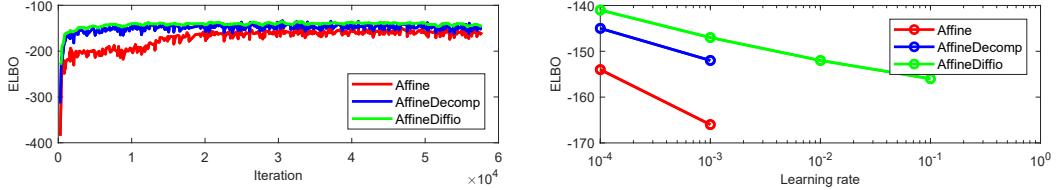


Figure 8: Top: Stability towards choice of learning rate for three different parametrizations of affine transformations. Missing values indicates that the network diverged. Bottom: Learning curves also show that the diffeomorphic affine parametrization converges faster and is more stable in its training.

These experiments was conducted on the MNIST dataset. For all three experiments we use the C-VITAE architecture with a neural network structure as Table 2. A batch size of 512 was used. The results where generated by changing the parametrization of the affine spatial transformer between

$$\text{Affine} \quad \mathcal{T}_\gamma(\mathbf{x}) = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{14} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (12)$$

$$\text{AffineDecomp} \quad \mathcal{T}_\gamma(\mathbf{x}) = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (13)$$

$$\text{AffineDiffio} \quad \mathcal{T}_\gamma(\mathbf{x}) = \text{expm} \left(\begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{14} \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (14)$$

and by varying the learning rate $\lambda = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The lower subplot of Figure 4, was generated using a learning rate of $\lambda = 10^{-4}$ to make sure that all transformer types would converge.

D Additional results

D.1 MNIST experiments

In Fig. 9 reconstructions from the different models can be seen. In Fig. 10 generated sampler from the different models can be seen. In Fig. 11 latent manipulations can be seen.

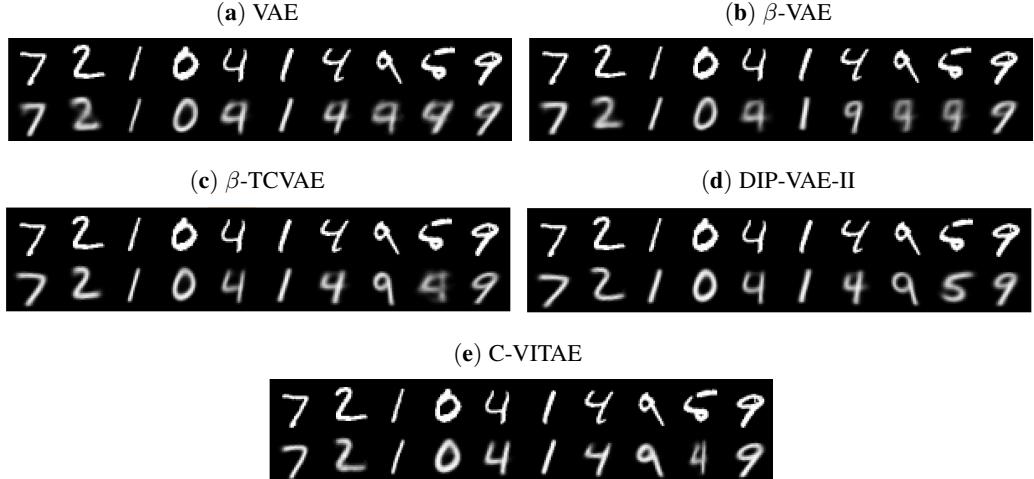


Figure 9: Samples from the test set (top rows) and the corresponding reconstructions (bottom rows) for all models. We clearly observe that the additional weight on the KL term in β -VAE, β -TCVAE and DIP-VAE-II makes the reconstructions worse.

(a) VAE	(b) β -VAE	(c) β -TCVAE
(d) DIP-VAE-II	(e) C-VITAE	

Figure 10: Samples from the prior distribution.

E SMPL experiment

In Fig. 12 reconstructions from the different models can be seen. In Fig. 13 generated sampler from the different models can be seen. In Fig. 11 latent manipulations can be seen.

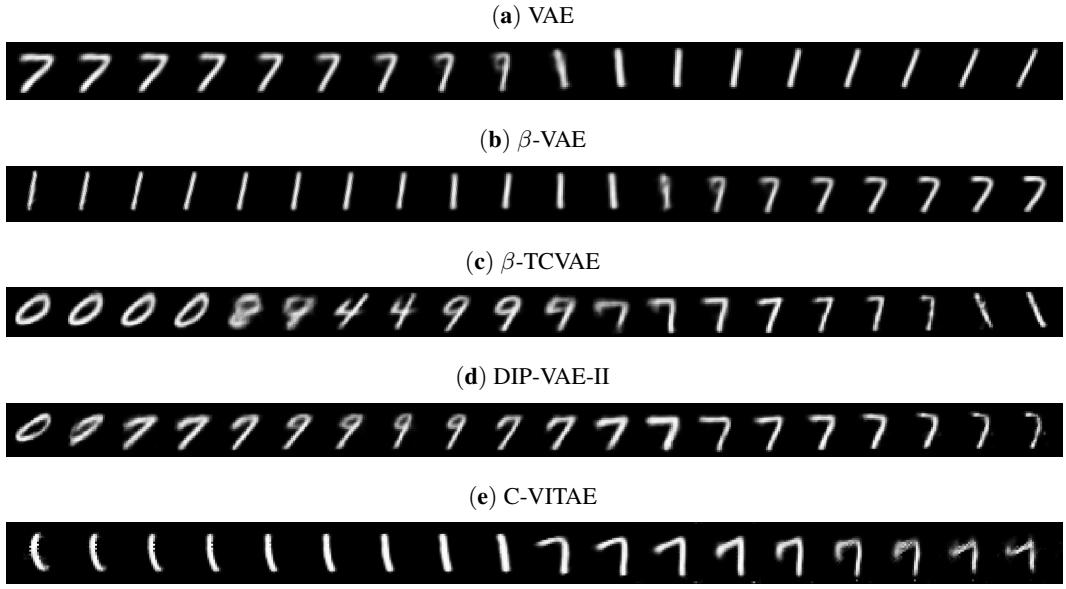


Figure 11: Latent manipulation. The images were generated by varying one latent dimension, while keeping the rest fixed. We choose the latent variable that qualitatively gave the best results.

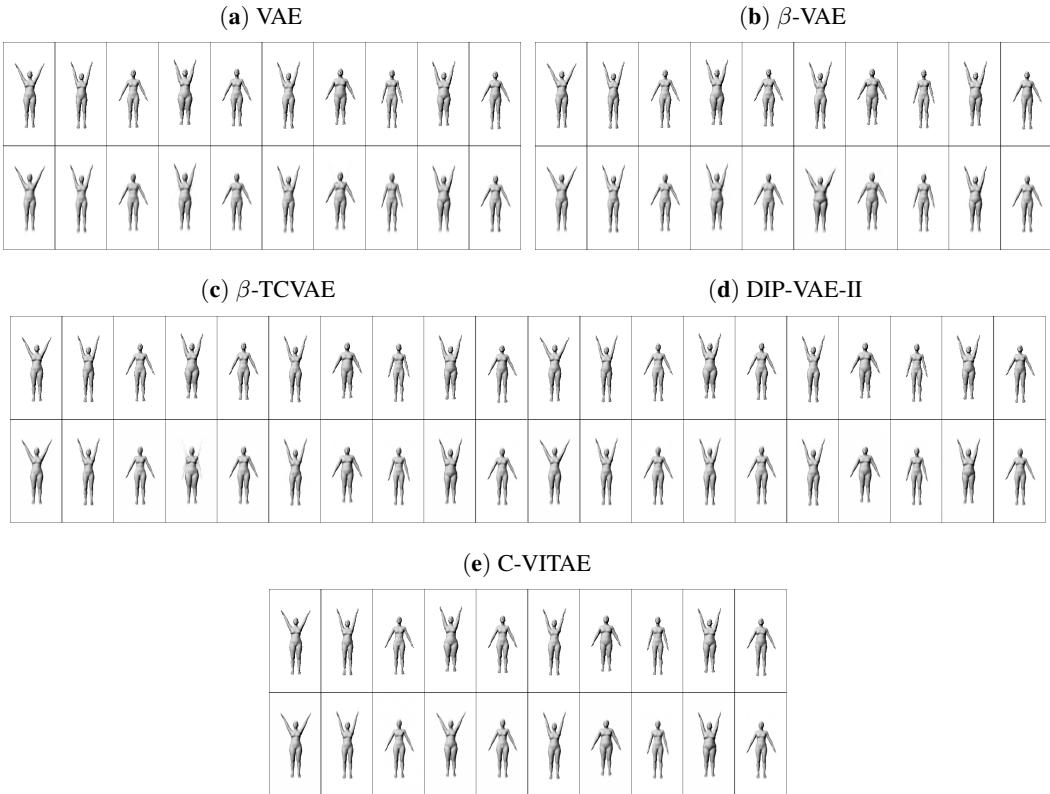


Figure 12: Test set reconstructions on SMPL dataset.

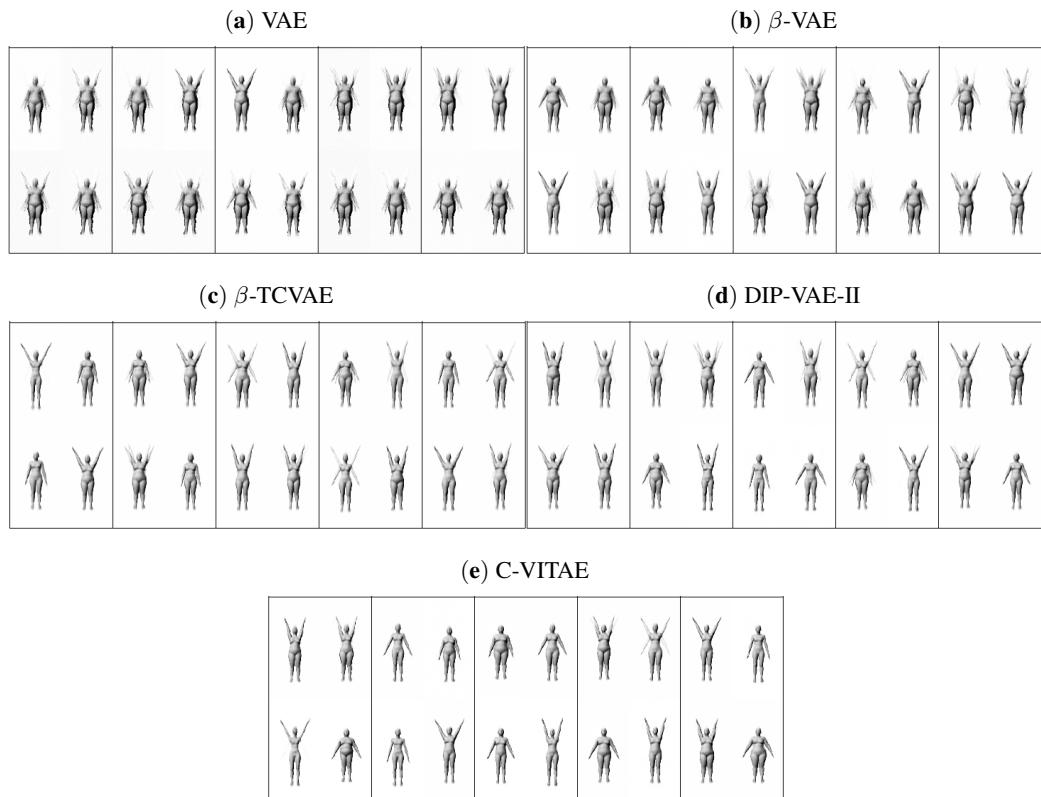


Figure 13: Samples from the prior distribution.

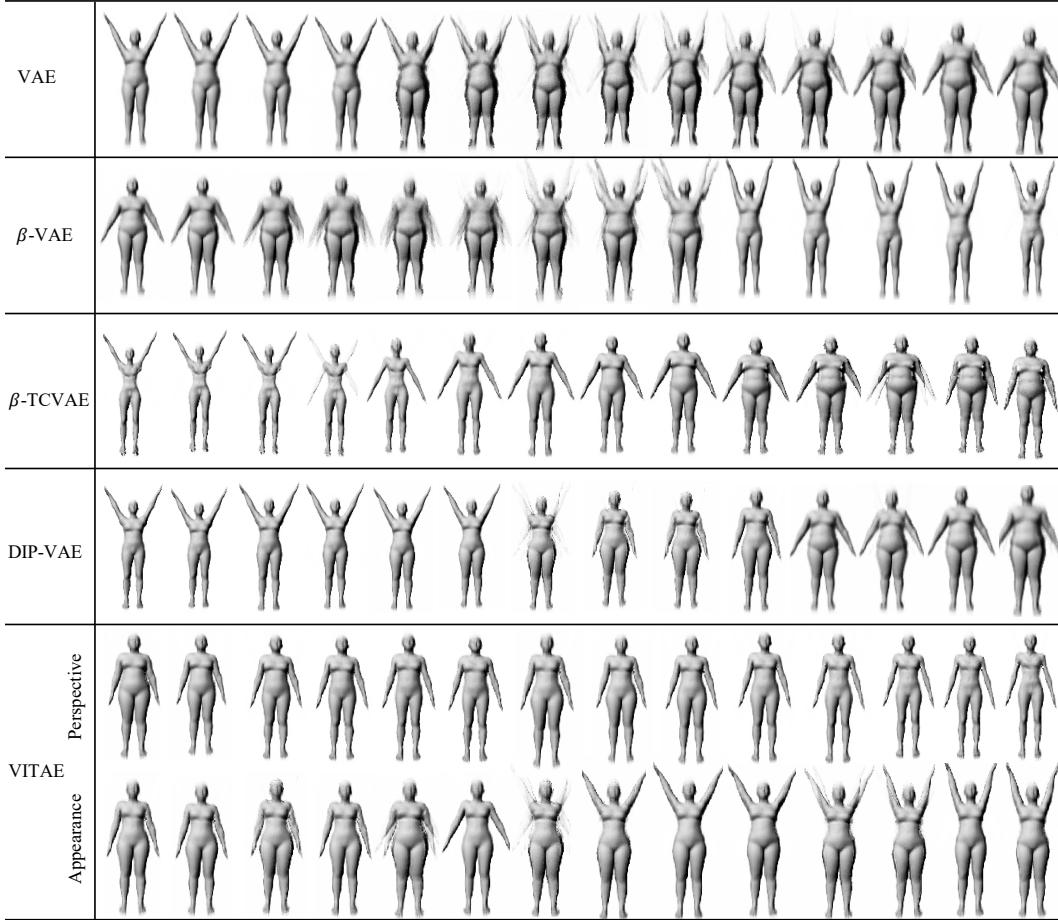


Figure 14: Disentanglement of body shape and body pose on SMPL-generated bodies for all models. The images are generated by varying one latent dimension, while keeping the rest fixed. For the C-VITAE model we have shown this for both the appearance and perspective spaces, since this is the only model where we quantitatively observe disentanglement.