

Table 5: The architecture details. “FC.” denotes fully connected layer, “conv.” denotes convolutional layer, “deconv” denotes transposed convolution layer. c is the dimension of color channel.

Encoder	Decoder
4×4 conv. 32 stride 2	FC.256
4×4 conv. 32 stride 2	FC. $4 \times 4 \times 64$
4×4 conv. 64 stride 2	4×4 deconv. 64 stride 2
4×4 conv. 64 stride 2	4×4 deconv. 32 stride 2
FC. 256	4×4 deconv. 32 stride 2
FC. 10	4×4 deconv. c stride 2

A APPENDIX

A.1 ARCHITECTURE

The details of architectures are listed in Table 5.

A.2 DISENTANGLEMENT-INVARIANT REPRESENTATIONS

In this section, we prove the proposed disentanglement-invariant transformation. Consider that we have a new representation by multiplying a diagonal matrix: $\mathbf{z}' = \mathbf{w}\mathbf{z}$, \mathbf{w} . We can calculate the Covariance between any two latent variables:

$$\begin{aligned}
 \text{Cov}(\mathbf{w}_i \mathbf{z}_i, \mathbf{w}_j \mathbf{z}_j) &= \mathbb{E}[(\mathbf{w}_i \mathbf{z}_i - \mathbb{E}[\mathbf{w}_i \mathbf{z}_i])(\mathbf{w}_j \mathbf{z}_j - \mathbb{E}[\mathbf{w}_j \mathbf{z}_j])] \\
 &= \mathbf{w}_i \mathbf{w}_j (\mathbb{E}[\mathbf{z}_j] - \mathbb{E}[\mathbf{z}_i] \mathbb{E}[\mathbf{z}_j]) \\
 &= \mathbf{w}_i \mathbf{w}_j \text{Cov}(\mathbf{z}_i, \mathbf{z}_j),
 \end{aligned} \tag{9}$$

where the subscript denotes the index of latent variables. Note that we use a different notion in this section to simplify the formula.

Then we can get the correlation coefficient by

$$\begin{aligned}
 \rho(\mathbf{w}_i \mathbf{z}_i, \mathbf{w}_j \mathbf{z}_j) &= \frac{\text{Cov}(\mathbf{w}_i \mathbf{z}_i, \mathbf{w}_j \mathbf{z}_j)}{\sqrt{\text{Var}[\mathbf{w}_i \mathbf{z}_i] \text{Var}[\mathbf{w}_j \mathbf{z}_j]}} \\
 &= \rho(\mathbf{z}_i, \mathbf{z}_j).
 \end{aligned} \tag{10}$$

Therefore, the correlation matrix will not change by multiplying a diagonal matrix \mathbf{w} , $\mathbf{w} \neq 0$. We could create a disentanglement-invariant representation by multiplying a diagonal matrix.

A.3 ESTIMATION OF $I(\mathbf{z}_j; \mathbf{c}_i)$

Given an inference network $q(\mathbf{z}|\mathbf{x})$, we use the Markov chain Monte Carlo (MCMC) method to get samples from $q(\mathbf{z})$ by the formula $q(\mathbf{z}) = q(\mathbf{z}|\mathbf{x})p(\mathbf{x})$. We use 10, 000 points to estimate $q(\mathbf{z})$. Then, we discretize these latent variables by a histogram with 20 bins. After discretizing one latent variable, we call a discrete mutual information estimation algorithm to calculate $I(\mathbf{w}_j \mathbf{z}_j; \mathbf{c}_i)$ by a 2D histogram.

A.4 LATENT TRAVERSALS

We compare DeVAE to others with latent traversals on Shapes3D and dSprites. Each column denotes the generated images by traversing one latent variable from -2 to 2. We also interpret the extracted factor at the bottom. From Figure 6 and Figure 7, we can see that DeVAE has a lower entanglement level. Note that only DeVAE disentangles object size isolated on Shapes3D.

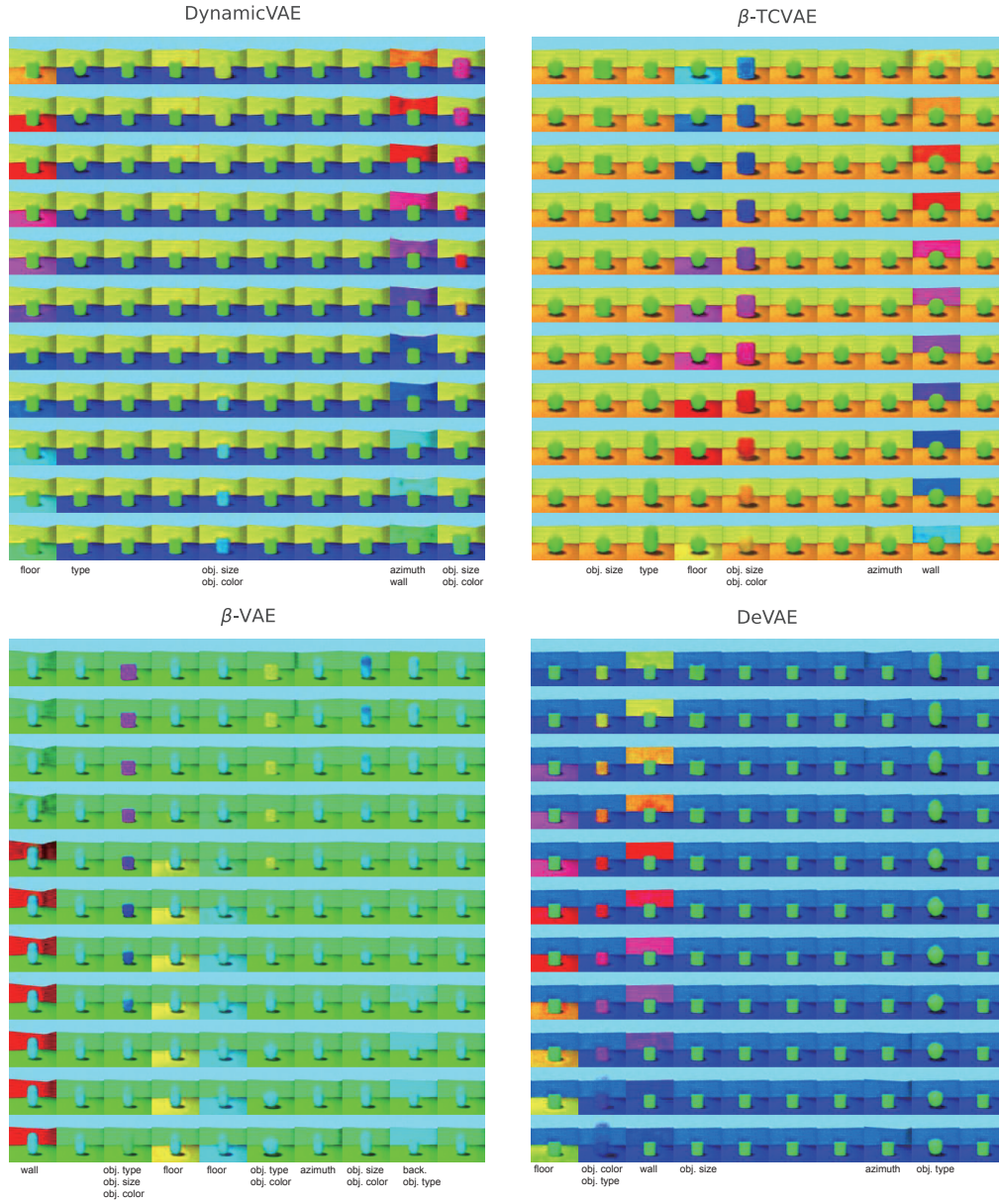


Figure 6: Latent traversal on Shapes3D. "back," denotes background color, "floor" denotes floor color, "obj." denotes object, and "wall" denotes wall color.

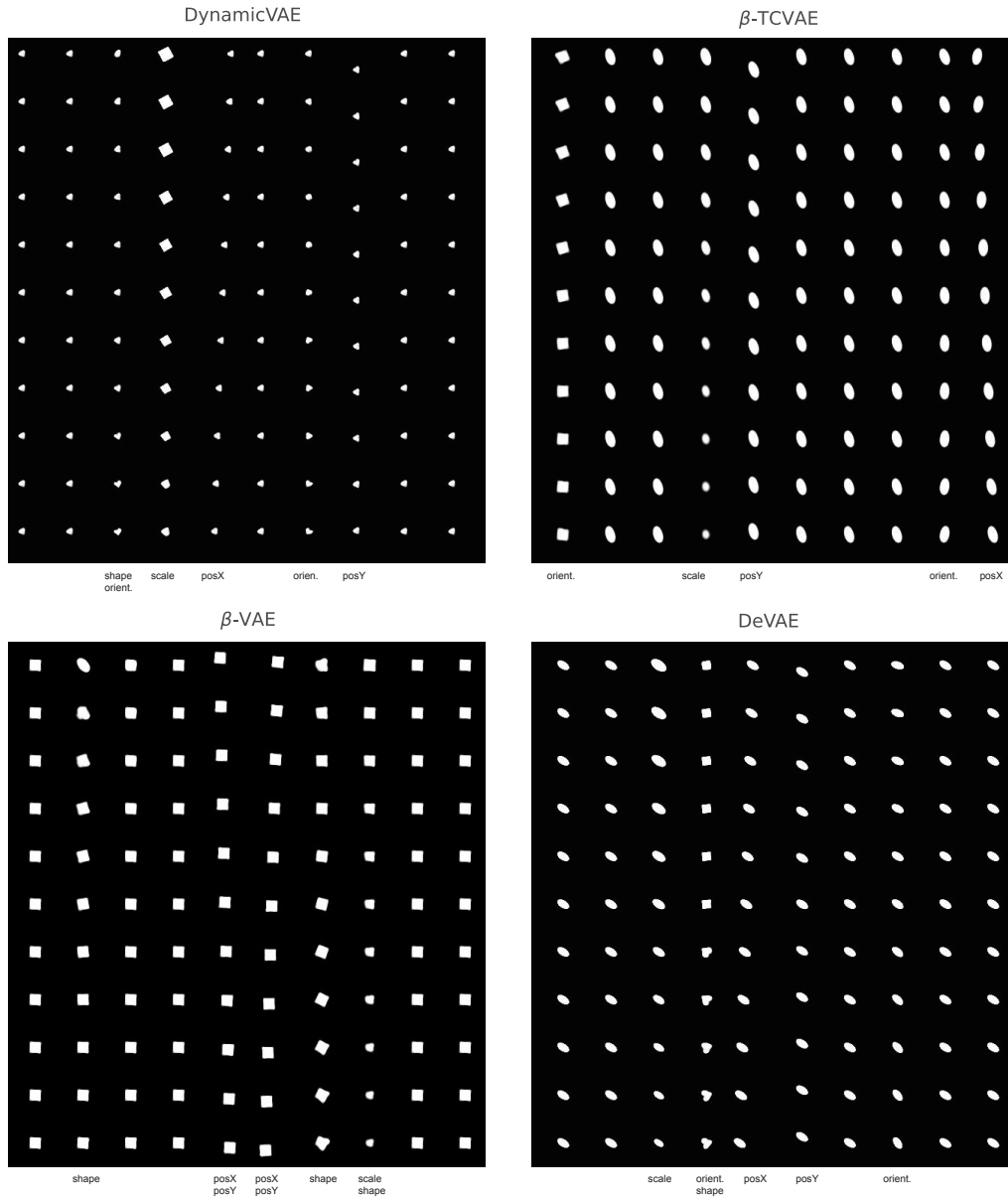


Figure 7: Latent traversal on dSprites.