Supplementary Material for Disentangling Latent Hands for Image Synthesis and Pose Estimation

Linlin Yang
University of Bonn, Germany
yangl@cs.uni-bonn.de

Angela Yao
National University of Singapore, Singapore
ayao@comp.nus.edu.sg

In the following supplementary material, we present more details for our proposed method. Section A provides the derivations for Section 3, the Methodology, in the main manuscripts; Section B provides the details of our network architectures. Ablation experiments and more results are presented in Section C and Section D respectively. Section E shows the numerical evaluation of disentanglement. Note that all the notation and abbreviations here are consistent with the main manuscript.

A. Methodology

In this section, we provide full derivations of Eq. 2 and Eq. 4 in the main manuscript and get the joint objective (*i.e.* Eq. 5 in the main manuscript) of dVAE. Remember that we assume $\mathbf{z}$ can be fully specified by $\mathbf{z}_{\mathbf{y}_1}$ and $\mathbf{z}_{\mathbf{y}_2}$ with observed $\mathbf{y}_1$ and $\mathbf{y}_2$, and define a disentangled $\mathbf{z} = [\mathbf{z}_{\mathbf{y}_1}, \mathbf{z}_{\mathbf{y}_2}]$ based on $\mathbf{y}_1$, $\mathbf{y}_2$ and $\mathbf{x}$. In the ***disentangling step***, we factorize the joint distribution $\log p(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ into $\log p(\mathbf{y}_1, \mathbf{y}_2)$ and $\log p(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2)$. We define the distributions via the latent variable $\mathbf{z}$ and then we have:

$$\begin{aligned}
&\log p(\mathbf{y}_1, \mathbf{y}_2) \\
&= \int_{\mathbf{z}} q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) \log \frac{q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2)}{p_\theta(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2)} d\mathbf{z} \\
&+ \int_{\mathbf{z}} q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) \log \frac{p(\mathbf{z}) p_\theta(\mathbf{y}_1, \mathbf{y}_2|\mathbf{z})}{q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2)} d\mathbf{z} \\
&= D_{KL}(q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) || p_\theta(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2)) \\
&+ E_{\mathbf{z} \sim q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}} \log p_\theta(\mathbf{y}_1, \mathbf{y}_2|\mathbf{z}) \\
&- D_{KL}(q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) || p(\mathbf{z})),
\end{aligned} \tag{1}$$

and

$$\begin{aligned}
&\log p(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2) \\
&= \int_{\mathbf{z}} q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) \log \frac{q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2)}{p_\theta(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2, \mathbf{x})} d\mathbf{z} \\
&+ \int_{\mathbf{z}} q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) \log \frac{p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y}_1, \mathbf{y}_2) p_\theta(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2)}{q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2)} d\mathbf{z} \\
&= D_{KL}(q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) || p_\theta(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2, \mathbf{x})) \\
&+ E_{\mathbf{z} \sim q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}} \log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y}_1, \mathbf{y}_2) \\
&- D_{KL}(q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) || p_\theta(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2)).
\end{aligned} \tag{2}$$

Combining the two log probabilities, we get the joint distribution between $\mathbf{x}$, $\mathbf{y}_1$ and $\mathbf{y}_2$ as:

$$\begin{aligned}
&\log p(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2) \\
&= \log p(\mathbf{x}|\mathbf{y}_1, \mathbf{y}_2) + \log p(\mathbf{y}_1, \mathbf{y}_2) \\
&= D_{KL}(q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) || p_\theta(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2, \mathbf{x})) \\
&+ E_{\mathbf{z} \sim q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}} \log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y}_1, \mathbf{y}_2) \\
&+ E_{\mathbf{z} \sim q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}} \log p_\theta(\mathbf{y}_1, \mathbf{y}_2|\mathbf{z}) \\
&- D_{KL}(q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) || p(\mathbf{z})) \\
&= D_{KL}(q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) || p_\theta(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2, \mathbf{x})) \\
&+ E_{\mathbf{z} \sim q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}} \log p_\theta(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2|\mathbf{z}) \\
&- D_{KL}(q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) || p(\mathbf{z})).
\end{aligned} \tag{3}$$

Since $D_{KL}(\cdot) \geq 0$ for any distribution and $\mathbf{z}$ is a disentangled representation, the evidence lower bound is obtained below:

$$\begin{aligned}
&\log p(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2) \\
&\geq \text{ELBO}_{\text{dis}}(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \phi_{\mathbf{y}_1}, \phi_{\mathbf{y}_2}, \theta_{\mathbf{y}_1}, \theta_{\mathbf{y}_2}, \theta_{\mathbf{x}}) \\
&= E_{\mathbf{z} \sim q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}} \log p_\theta(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2|\mathbf{z}) \\
&- D_{KL}(q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) || p(\mathbf{z})) \\
&= E_{\mathbf{z} \sim q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}} \log p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z}) \\
&+ E_{\mathbf{z}_{\mathbf{y}_1} \sim q_{\phi_{\mathbf{y}_1}}} \log p_{\theta_{\mathbf{y}_1}}(\mathbf{y}_1|\mathbf{z}_{\mathbf{y}_1}) \\
&+ E_{\mathbf{z}_{\mathbf{y}_2} \sim q_{\phi_{\mathbf{y}_2}}} \log p_{\theta_{\mathbf{y}_2}}(\mathbf{y}_2|\mathbf{z}_{\mathbf{y}_2}) \\
&- D_{KL}(q_{\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2}}(\mathbf{z}|\mathbf{y}_1, \mathbf{y}_2) || p(\mathbf{z})).
\end{aligned} \tag{4}$$

To train the unpaired encoder $q_{\phi_{\mathbf{x}}}(\mathbf{z}|\mathbf{x})$ ***in the embedding step*** [5], we fix the decoders and maximize the following:

$$\begin{aligned}
&\mathcal{L}(\phi_{\mathbf{x}}|\theta_{\mathbf{y}_1},\theta_{\mathbf{y}_2},\theta_{\mathbf{x}})\\
&= -D_{KL}(q_{\phi_{\mathbf{x}}}(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x},\mathbf{y}_1,\mathbf{y}_2))\\
&= \text{ELBO}_{\text{emb}}(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\phi_{\mathbf{x}}) - \log p(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)\\
&= E_{\mathbf{z}\sim q_{\phi_{\mathbf{x}}}} \log p_\theta(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2|\mathbf{z})\\
&- D_{KL}(q_{\phi_{\mathbf{x}}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - \log p(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)\\
&= E_{\mathbf{z}\sim q_{\phi_{\mathbf{x}}}} \log p_{\theta_{\mathbf{x}}}(\mathbf{x}|\mathbf{z}) + E_{\mathbf{z}_{\mathbf{y}_1}\sim q_{\phi_{\mathbf{x}}}} \log p_{\theta_{\mathbf{y}_1}}(\mathbf{y}_1|\mathbf{z}_{\mathbf{y}_1})\\
&+ E_{\mathbf{z}_{\mathbf{y}_2}\sim q_{\phi_{\mathbf{x}}}} \log p_{\theta_{\mathbf{y}_2}}(\mathbf{y}_2|\mathbf{z}_{\mathbf{y}_2}) - D_{KL}(q_{\phi_{\mathbf{x}}}(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))\\
&- \log p(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2).
\end{aligned} \tag{5}$$

Here $\log p(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)$ is constant with respect to $\phi_{\mathbf{x}}$ and θ's and hence can be dropped. So we get the final objective by combining the disentangling and embedding evidence lower bounds:

$$\begin{aligned}
&\mathcal{L}(\phi_{\mathbf{x}},\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2},\theta_{\mathbf{x}},\theta_{\mathbf{y}_1},\theta_{\mathbf{y}_2})\\
&= \text{ELBO}_{\text{dis}}(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\phi_{\mathbf{y}_1},\phi_{\mathbf{y}_2},\theta_{\mathbf{x}},\theta_{\mathbf{y}_1},\theta_{\mathbf{y}_2})\\
&+ \text{ELBO}_{\text{emb}}(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\phi_{\mathbf{x}}).
\end{aligned} \tag{6}$$

B. Model Architectures

Here we specify the design of our networks. The encoder and decoder architectures for RGB images and background content are shown in Table 1. For encoding images, we use ResNet-18 [2]; for decoding images, we follow the decoder architecture DCGAN [4]. Here, we use 256 × 256 RGB images as the input of ResNet-18. Also the encoder and decoder architectures for CPose, viewpoint or 3DPose are detailed in Table 2. Especially, for additional $\mathbf{z}_{\mathbf{u}}$, we design shared layers for the decoder of RGB images and 3DPose as shown in Table 3. Abbreviations: N for number of kernels or neurons, FC stands for fully connected layers, TCONV stands for transposed convolutional layers with 5 × 5 kernels, stride of size 2 and padding of size 1, BN stands for batch normalization layers. For example, FC-(N512) refers to a fully connected layer with 512 neurons.

Encoder	Decoder
ResNet-18	FC-(N8192) Reshape(8,8,128), BN TCONV-(N64), BN, Relu TCONV-(N32), BN, Relu TCONV-(N16), BN, Relu TCONV-(N8), BN, Relu TCONV-(N3), Tanh Reshape(256,256,3)

Table 1: Encoder and Decoder architectures for hand images or background content.

Encoder	Decoder
Flatten	FC-(N512), Relu
FC-(N512), Relu	FC-(N512), Relu
FC-(N512), Relu	FC-(N512), Relu
FC-(N512), Relu	FC-(N512), Relu
FC-(N512), Relu	FC-(N512), Relu
FC-(N512), Relu	FC-(N9) or (N63)
FC-(N512)	Reshape (3,3,1) or (21,3,1)

Table 2: Encoder and Decoder architectures for viewpoint, CPose or 3DPose. The final reshape is (3,3,1) for the viewpoint and (21,3,1) for CPose and 3DPose.

Decoder	
FC-(N8192) Reshape(8,8,128), BN TCONV-(N64), BN, Relu TCONV-(N32), BN, Relu TCONV-(N16), BN, Relu TCONV-(N8), BN, Relu	

TCONV-(N3), Tanh
Reshape(256,256,3)
	FC-(N512), Relu
	FC-(N512), Relu
	FC-(N63)
	Reshape(21,3,1)

Table 3: Decoder architectures for images and 3DPose with shared layers. The final reshape is (256,256,3) for the images and (21,3,1) for 3DPose.

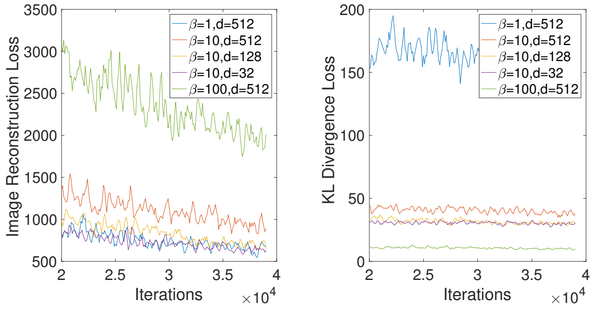

Figure 1: Ablative study on different parameters β and d. Left: Image reconstruction loss. Right: KL divergence loss.

C. Ablation Experiments

We test different β and d for image reconstruction on STB. For convenience, we set $d_{\mathbf{z}_{\mathbf{y}_1}} = d_{\mathbf{z}_{\mathbf{y}_2}} = 0.5d$. Fig. 1 shows the image reconstruction loss (*i.e.* mean squared reconstruction error) and the KL divergence loss (*i.e.* $D_{KL}(q_\phi(\mathbf{z}|\cdot)||p(\mathbf{z}))$). We can see that as β increases,

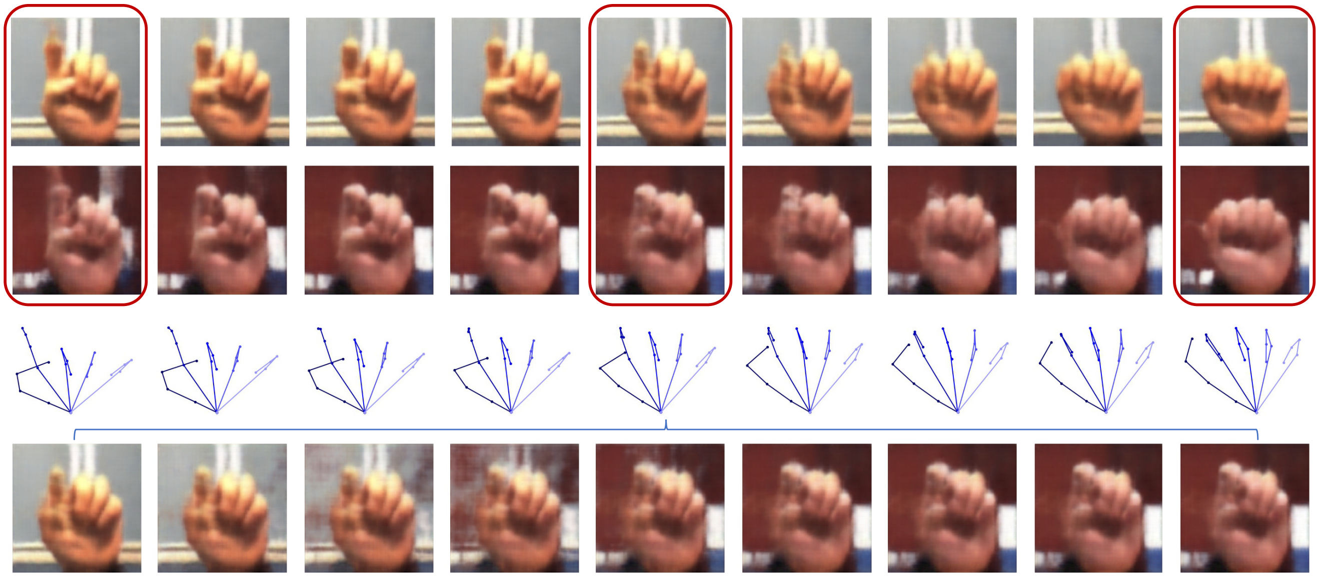

Figure 2: Latent space walk. The images in the red boxes are provided inputs. The first two rows show synthesized images when interpolating on the latent 3DPose space; the third row shows skeletons of the reconstructed 3DPose. The fourth row shows synthesized images when the pose is fixed (to the fifth column) when interpolating in the content latent space.

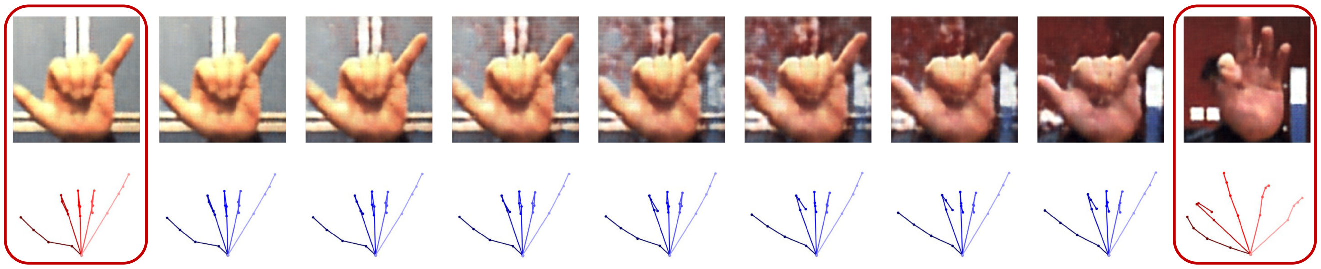

Figure 3: Latent space walk, interpolating $\mathbf{z_u}$ *representing image background content. The images along with groundtruth 3DPose (red) in the red box are the input points; the first row shows generated images and the second row corresponding reconstructed 3DPose (blue). Note that because we are interpolating only on the background content, the pose stays well-fixed.*

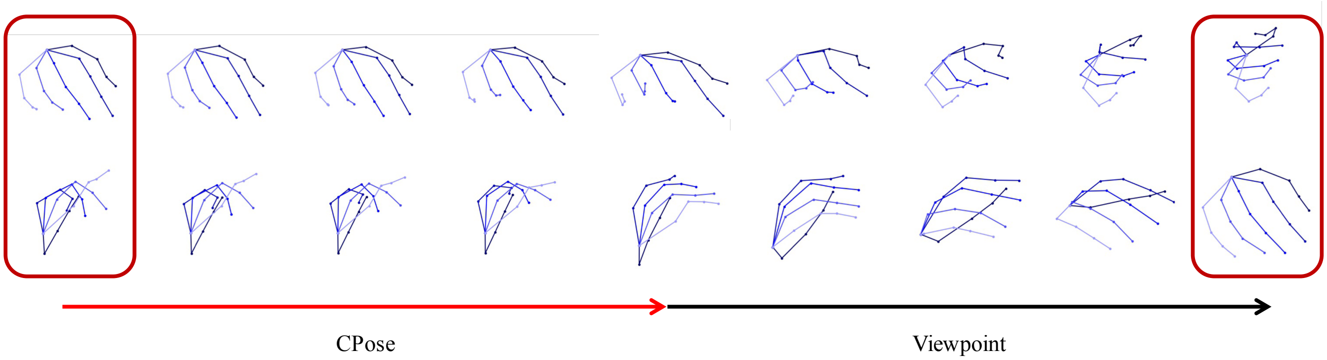

Figure 4: Latent space walk. The 3DPose in the red boxes are provided inputs. The left most five rows show synthesized 3DPose when the viewpoint is fixed and interpolating in the CPose latent space; the right most five rows show synthesized 3DPose when the CPose is fixed and interpolating in the viewpoint latent space.

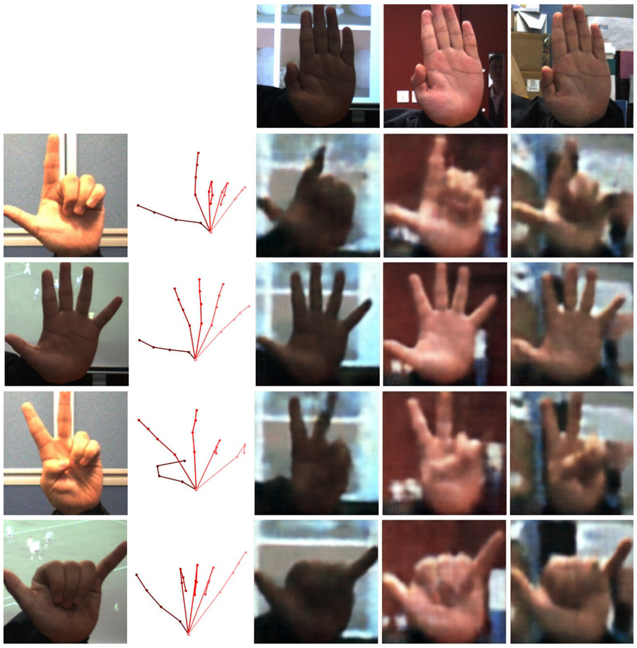

Figure 5: Pose transfer. The first column corresponds to images from which we extract the 3DPose (ground truth pose in second column); the first row corresponds to tag images columns we extract the latent content; the 2-5 rows, 3-5 columns are pose transferred images.

Figure 6: Random sampling.

the image reconstruction loss increases but the KL divergence loss decreases. β here trades off between latent space capacity and reconstruction accuracy. The use of β has been discussed in the work [3]. In the experiments, we empirically set $\beta = 100$ for image synthesis and $\beta = 0.01$ for pose estimation. As d increases, both the image reconstruction loss and the KL divergence loss increase. It is probably because the latent space will be over-completed when d is large enough. So we prefer to choose $d = 64$ in the experiments.

D. Additional Results

This section shows more results. In Fig. 2 and 5, we provide additional latent space random walk and pose transfer results with fully specified $\mathbf{z}$. In Fig. 3, we provide additional illustrations of random walk on $\mathbf{z_u}$. In Fig. 4, we show random walk on the CPose part and the viewpoint part during hand pose estimation. We show the synthesized 3DPose when we interpolate the CPose while keeping the viewpoint fixed and when we interpolate viewpoint while keeping the CPose fixed. In both random walks, the reconstructed poses demonstrate a smoothness and consistency of the latent space. In addition, we randomly sample 10 points of $\mathbf{z}$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and show the corresponding synthesize images in Fig. 6.

	$\mathbf{z}_{\mathbf{y}_1}$ $\sim q_{\phi_\mathbf{x}}$	$\mathbf{z}_{\mathbf{y}_1}$ $\sim q_{\phi_{\mathbf{y}_1}}$	$\mathbf{z}_{\mathbf{y}_2}$ $\sim q_{\phi_\mathbf{x}}$	$\mathbf{z}_{\mathbf{y}_2}$ $\sim q_{\phi_{\mathbf{y}_2}}$	$\mathbf{z}_{cross}$ $\sim q_{\phi_\mathbf{x}}$	Noise $\sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{y_1}$	0.773	0.779	0.561	0.562	0.782	0.555
$\mathbf{y_2}$	1.84	1.99	1.26	1.27	1.09	2.15

Table 4: AUC for CPose $\mathbf{y}_1$ (higher is better) and MSE for viewpoint $\mathbf{y}_2$ (lower is better).

E. Evaluation of Disentanglement

To evaluate the disentanglement of fully specified latent $\mathbf{z}$, we fix the dimensionality of d of $\mathbf{z}$ to 8 and use the disentanglement metric score from β-VAE [3]. We generate 1000 samples of $\mathbf{z}^{32}_{\text{diff}}$ according to [3]. 500 samples are used for training and others for testing. We receive a score of 100%, implying that our representation is completely disentangled. We believe that our (very high) score is due to the fact that we use labelled factors of variations during training and only disentangle the latent space into two parts.

The metric from [3] however is not applicable for multiple modalities, so in addition, we generate 300 samples of $\mathbf{z}$ and evaluate our multiple modalities case according to [1]. This approach regresses $\mathbf{y}_1, \mathbf{y}_2$ (CPose and viewpoint respectively) from sub-latent variables $\mathbf{z}_{\mathbf{y}_1}, \mathbf{z}_{\mathbf{y}_2}$. The quality of the disentanglement is high if $\mathbf{z}_{\mathbf{y}_1}$ is informative about $\mathbf{y}_1$ but not $\mathbf{z}_{\mathbf{y}_2}$ and vice-versa. For comparison, we look at values drawn from noise and from $\mathbf{z}_{\text{cross}}$ which serve as lower and upper bounds. Results in Tab. 4 show that our regressed variables fall close to these bounds, indicating that we are able to disentangle CPose very well, though viewpoint could be improved.

References

[1] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *AAAI*, 2018. 4

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[3] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2016. 4

[4] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *arXiv preprint arXiv:1511.06434*, 2015. 2

[5] Ramakrishna Vedantam, Ian Fischer, Jonathan Huang, and Kevin Murphy. Generative models of visually grounded imagination. In *ICLR*, 2018. 2