1. Appendix I.a. Network Architectures

We denote each fully-connected layer $f_{c}(d)$ by its output dimension $d$, and volumetric convolution layer by $conv3D(k, c, s)$ representing kernel size of $k$, strides of $s$ across three spatial axes, and $c$ channels. 2D convolutional layer is represented as $conv2D(k, c, s)$. The encoder and generator for aligned shapes take as input an $30 \times 30 \times 30 \times 1$ tensor, and consists of 3 convolution layers: $conv3D(4, 32, 2)$, $conv3D(4, 64, 2)$, $conv3D(4, 64, 2)$; two fully-connected layers $f_{c}(200)$ and $f_{c}(200)$, regressing from the last convolution feature to the 200-dimensional mean and variance vectors for style embedding, following [2]. The decoder takes in the 200-dimensional style vector, and consists of one fully-connected layer $f_{c}(8192)$ to connect the input vector to a $4 \times 4 \times 4 \times 128$ convolutional feature; and 3 transpose convolution layers $deconv3D(4, 64, 2)$, $deconv3D(4, 32, 2)$, $deconv3D(4, 1, 2)$ output the reconstructed shape with size $30 \times 30 \times 30 \times 1$. All convolution and transpose convolution layer are batch normalized except the first convolution and last transpose convolution layer. LeakyReLU [3][1] is the rectifier for all layers except the output layer which uses tanh. This architecture is also used for 3D VAE in Section 4.2.

Image to Style/Pose Regressors The two regressors have identical architecture of convolution layers: $conv2D(11, 64, 4)$, $conv2D(5, 128, 2)$, $conv2D(5, 256, 2)$, $conv2D(5, 512, 2)$, $conv2D(3, 200, 2)$. For the style regressor, an $f_{c}(200)$ connects the last convolution layer to the style parameters. For the pose regressor, $f_{c}(5)$ is used instead. All but the first convolution layers are batch normalized, and rectified with LeakyReLU.

2. Appendix I.b. Training Details

p-TL Both the aligned shape autoencoder and the style/pose regressors are trained with Adam optimizer at a learning rate of 0.0003 and batch size of 100.

p-3D-VAE-GAN We follow [4] in training the 3D-VAE-GAN. We replace the L2 voxel-wise reconstruction loss described in [5] with the L2 loss between the last layer convolution features in the discriminator. We use RMSProp with a learning rate of 2e-5 and and batch size of 100 in training the 3D-VAE-GAN. The pose regressor is trained in the same routine as in p-TL.

3. Appendix II. Fine-tuning Details

In fine-tuning, we fine-tuned all the parameters in style/pose regressors, with an tiny learning rate of 1e-12. Each batch is natural images with silhouette annotations, mixed with rendered image-shape pairs. In this case, the loss is composed of two parts with both weight of 1: reprojection loss for natural images, and loss in shape for natural images (for p-VAE, this part is the euclidean loss in style and pose; for p-3D-VAE-GAN, this part is loss of VAE-GAN).

Fig. 1 gives an evaluation of the test 3D AP of pose-aware shapes over the ratio of natural images in a training batch. We may observe a relatively equivalent portion of rendered and natural samples in a fine-tuning batch returns the best AP, while too few natural images helps little in fine-tuning, and too many natural samples easily lead to overfitting.

4. Appendix III. Reprojected Silhouettes as Instance Segmentation

In this section we showcase test results for MS COCO dataset where reprojected silhouettes from our pose-aware shape reconstructions could be used as instance segmentation, similar to the practice in [6]. We list 20 samples for each of the three categories: aeroplane, chair, car in Fig. 2 3 4, respectively.
Figure 1: 3D AP as a function of ratio of natural images in a training batch, averaged over both approaches and all categories.

References


Figure 2: Reprojected silhouette as instance segmentation (aeroplane). For each sample we show the input image (left), ground truth segmentation (middle left), reprojected silhouette (middle right), and reconstructed pose-aware shape (right).
Figure 3: Reprojected silhouette as instance segmentation (chair). For each sample we show the input image (left), ground truth segmentation (middle left), reprojected silhouette (middle right), and reconstructed pose-aware shape (right).
Figure 4: **Reprojected silhouette as instance segmentation (car)**. For each sample we show the input image (left), ground truth segmentation (middle left), reprojected silhouette(middle right), and reconstructed pose-aware shape(right).