Appendix

A. Robotic Experiment Details

All of the robotic experiments were conducted on a PR2 robot. The robot was controlled at 20 Hz via direct effort control,[2] and camera images were recorded using the RGB camera on a PrimeSense Carmine sensor. The images were downsamples to 240×240. The learned policies controlled one 7 DoF arm of the robot. The camera was kept fixed in each experiment. Each episode was 5 seconds in length. For each task, the cost function required reaching the goal state, defined both by visual features and gripper pose. Similar to previous work, the cost was given by the following equation:

$$\ell(\mathbf{x}_t, \mathbf{u}_t) = w_{\ell_2} d_t^2 + w_{\log} \log(d_t^2 + \alpha) + w_{\mathbf{u}} \|\mathbf{u}_t\|^2,$$

where d_t is the distance between three points in the space of the end-effector and learned feature points in 2D and their respective target positions[3], and the weights are set to $w_{\ell_2} = 10^{-3}$, $w_{\log} = 1.0$, and $w_{\mathbf{u}} = 10^{-2}$. The quadratic term in the cost encourages moving towards the target when it is far, while the logarithm term encourages reaching the target state precisely, as discussed in prior work [8]. The rice scoop task used two target states, in sequence, with half of the episode (2.5 seconds) devoted to each target. For each of the tasks, the objects were reset to their starting positions manually between trials during training. We discuss the particular setup for each experiment below:

a) Lego block: The lego block task required the robot to push a lego block 30 cm to the left of its initial position. For this task, we measured and reported the distance between the top corner of the goal block position to the nearest corner of the lego block at the end of the trial. In some of the baseline evaluations, the lego block was flipped over, and the nearest corner was still used to measure distance to the goal.

b) Bag transfer: The bag transfer task required the robot to place a white bag into a bowl, using a spoon. At the start of each trial, the robot was grasping the spoon with the bag in the spoon. A trial was considered successful if the bag was inside the bowl and did not extend outside of the bowl. In practice, the bag was very clearly entirely in the bowl, or entirely outside of the bowl during all evaluations.

c) Rice scoop: The rice scooping task required the robot to use a spatula to lift a small bag of rice off of a table and place it in a bowl. At the start of each trial, the spatula was in the grasp of the robot gripper, and the bag of rice was on the table, about 3 cm from the bowl. As with the bag transfer task, a trial was considered successful if the bag of rice was inside the bowl and did not extend outside of the bowl. In practice, the bag was very clearly in the bowl, or outside of the bowl during all evaluations.

d) Loop hook: The loop hook task required the robot to place a loop of rope onto a metal hook attached to a scale, for different positions of the hook. At training time, the scale was placed at four different starting positions along a metal pole that were equally spaced across 24 cm of the pole. The test positions were the three midpoints between the four training positions. A trial was considered successful if, upon releasing the rope, the loop of rope would hang from the hook. In practice, the failed trials using our approach were often off by only 1-2 mm, whereas the controller with no vision was typically off by several centimeters.

B. Neural Network Architectures for Prior Work Methods

We compare our network with two common neural network architectures. The first baseline architecture is the one used by Lange et al. [1]. The network is composed of 8 encoder layers and 8 decoder layers. To match the original architecture as closely as possible, we converted our 240×240 RGB images into 60×60 grayscale images before passing them through the network. The encoder starts with 3 convolution layers with filter size 7×7, where the last convolution layer has stride 2. The last convolution layer is followed by 6 fully connected layers, the size of which are 288, 144, 72, 36, 18 and 10 respectively. The last fully connected layer forms the bottleneck of the autoencoder. We chose 10 as the dimension of the bottleneck, since the system has roughly 10 degrees of freedom. The decoder consists of 6 mirrored fully connected layers followed by 3 deconvolution layers, finally reconstructing the down sampled 60×60 image. We used ReLU nonlinearities between each layer. Following [1], we pre-train each pair of the encoder-decoder layers for 4000 iterations. Then, we perform fine tuning on the entire network until the validation error plateaus.

We also experimented with a more widely adopted convolutional architecture. The $240 \times 240 \times 3$ image is directly passed to the network. This network starts with 3 convolutional layers. As in our network architecture, conv1 consists of 64 7×7 filters with stride 2, conv2 has 32 5×5 filters with stride 1, and conv3 has 16 5×5 filters with stride 1, each followed by batch normalization and ReLU nonlinearities. Unlike our architecture, this baseline architecture performs max-pooling after each convolution layer in order to decrease the dimensionality of the feature maps. The convolution layers are followed by two fully connected layers with 512 and 32 units respectively, the last of which forms the bottleneck of the network. These layers together form the encoder, and a mirroring architecture, consisting of fully connected layers and deconvolution layers, forms the decoder. We initialize the first convolution layer with weights trained on ImageNet, and train the network until validation error plateaus.

[2] The PR2 robot does not provide for closed loop torque control, but instead supports an effort control interface that directly sets feedforward motor voltages. In practice, these voltages are roughly proportional to feedforward torques, but are also affected by friction and damping.

[3] Three points fully define the pose of the end-effector.