Supplementary for Compositional Plan Vectors

Anonymous Author(s)
Affiliation
Address
email

1 Network Architectures

1.1 2D environment

The observation is an RGB image of 33x30 pixels. The architecture for g concatenates the first and last image of the reference trajectory along the channel dimension, to obtain an input size of 33x30x6. This is followed by 4 convolutions with 16, 32,64, and 64 channels, respectively, with ReLU activations. The 3x3x64 output is flattened and a fully connected layer reduces this to the desired embedding dimension. The same architecture is used for the TECNet encoder. For the policy, the observation is passed through a convolutional network with the same architecture as above and the output is concatenated with the subtraction of embeddings as defined in the paper's method. This concatenation is passed through a 4 layer fully connected network with 64 hidden units per layer and ReLU activations. The output is softmaxed to produce a distribution over the 6 actions. The TECNet uses the same architecture, but the reference trajectory embeddings are normalized there is no subtraction; instead, the initial image of the current trajectory is concatenated with the observation. The naive model uses the same architecture but all four input images are concatenated for the initial convolutional network and there is no concatenation at the embedding level.

1.2 3D environment

For the object centric model, see Figure 1

2 Hyperparameters

We compared all models across embedding dimension sizes of [64,128,256, and 512]. In the 2D crafting environment, the 512 size was best for all methods. In the grasping environment, the 128 size was best for all methods. For TECNets, we tested $\lambda_{\text{ctr}} = 1$ and 0.1, and found that 0.1 was best. All models are trained on either k-80 GPUs or Titan X GPUs.

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

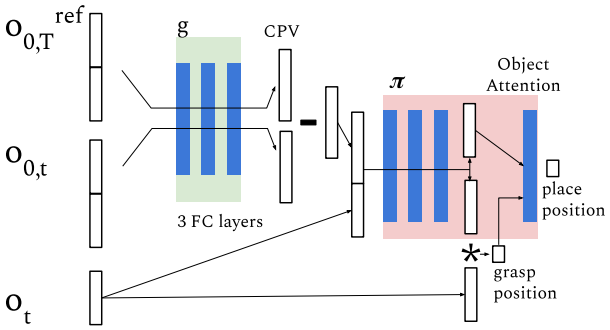

Figure 1: The object-centric network architecture we use for the 3D grasping environment. Because the observations include the concatenated positions of the objects in the scene, the policy chooses a grasp position by predicting a discrete classification over the objects grasping at the weighted sum of the object positions. The classification logits are passed back to the network to output the position at which to place the object.

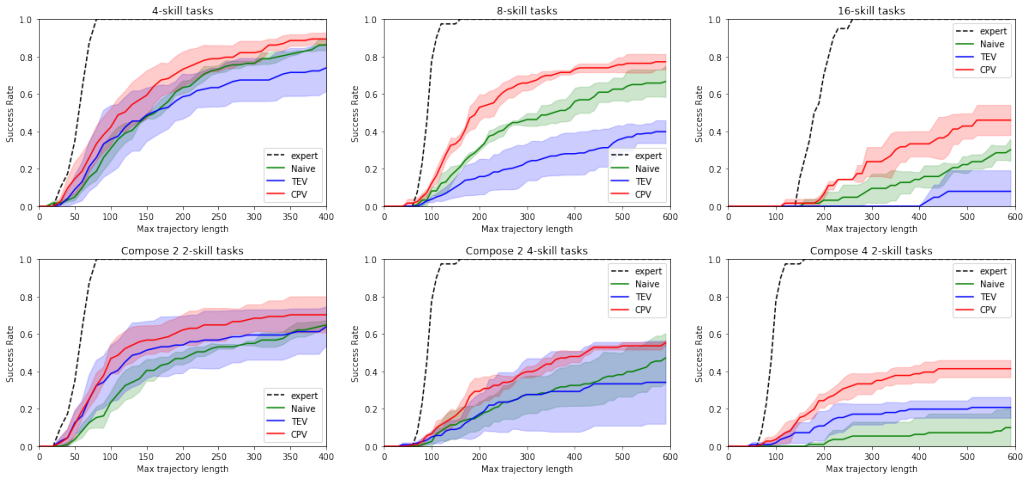

Figure 2: ROC curves for each evaluation task. Each plot show success rates for different max path lengths; a greater area under the curve indicates that policy accomplishes the tasks in fewer time steps. The top row shows success rates for policies conditioned on reference tasks using 4, 8, and 16 skills, respectively. The bottom row shows models conditioned on the average of embeddings from multiple reference trajectories, as described in each plot's title.