1 Summary

1. They model a scene as a collection objects, each having an explicit spatial location and implicit visual features.

2. They train an object-centric forward model using Interaction Network.

2 Object-centric Representation

1. They leverage the fact that multiple distinct objects are present in a typical scene.

2. Each obj can be represented by its location and visual description.

3. Concretely, given an observed image I^t and known location of N objs $\left\{b_n^t\right\}_{n=1}^N$ in the image, they use an obj-level representation $\left\{x_n^t\right\}_{n=1}^N$.

4. Each obj is represented as $x_n^t \equiv \left(b_n^t, f_n^t\right)$ where b_n^t is the observed/predicted location and f_n^t is the implicit visual feature of the obj.

5. b_n^t is the xy-coordinate in image space. f_n^t is the feature extracted from a fixed size window centered in b_n^t, extracted with a NN with resnet-18 as backbone.

3 Object-centric Forward Model

1. The forward model is denoted by $\mathcal{P}$, implemented as an instance of Interaction Network.

2. $\left\{x_n^{t+1}\right\} \equiv \mathcal{P}\left(\left\{x_n^t\right\}, a^{t+1}\right)$.

3. The Interaction Network takes in a graph (V, E), where each node is associated with a vector.

4. The network learns to output a new representation for each node by iterative message passing and aggregation.

5. The action of the gripper is added as an additional node, with the features being a learned embedding of the action a^{t+1}.

6. In addition to the predictor $\mathcal{P}$, we also train a decoder $\mathcal{D}$ to further regularize features to encode meaningful visual information.

7. The training data for $\mathcal{P}$ consists of triplet $\left(I^t, a^{t+1}, I^{t+1}\right)$. They also require the location of obj at every step $\left\{\hat{b}_n^t\right\}$ and the correspondence of those objs over time.

8. They argue that these annotations can be obtained using off-the-shelf visual detector.

9. The loss consists of two terms, a reconstruction loss and a prediction loss.

$$\begin{aligned} L_{recon} &= \left\|\mathcal{D}\left(\left\{\hat{x}_n^t\right\}\right) - I^t\right\|_1 \\ L_{pred} &= \left\|\mathcal{D}\left(\left\{x_n^{t+1}\right\}\right) - I^{t+1}\right\|_1 + \left\|\mathcal{P}\left(\left\{\hat{x}_n^t\right\}, a^{t+1}\right), \left\{\hat{x}_n^{t+1}\right\}\right\|_2^2 \end{aligned}$$

   where $\hat{x}_n^t$ represents features extracted from I^t at the ground truth object locations.

10. They argue that the reconst. loss forces the features to encoding meaningful visual info and prevent trivial solution.

11. They argue that the prediction loss encourages the forward model to predict plausibly in both feature space and pixel space.

4 Planning via a forward model

1. Given the goal $\{x_n^g\}$ and current state $\{x_n^0\}$, they generate an action trajectory $a^{1:T}$.

2. They use CEM to optimize for the sequence of action $a^{1:T}$.

3. CEM at every iteration draws S trajectories of length H.

4. The CEM cost function is

$$C = \sum_{n=1}^{N} \left(\left\| b_n^H, b_n^g \right\|_2^2 + \lambda \left\| f_n^H, f_n^g \right\|_2^2 \right)$$

5. They sample trajectories in continuous velocity space and upper-bound the magnitude.

5 Robust closed loop control via correction modeling

1. Given the current representation $\left\{x_n^t\right\}$ and the desired goal conf. $\{x_n^g\}$, CEM generates a sequence of action $a^{t+1:t+1+H}$.

2. Only a^{t+1} is executed, after which they replan.

3. But since they do not assume access to ground truth object locations at intermediate steps, it is not obvious what the new current rep $\left\{x_n^{t+1}\right\}$ should be.

4. If they use the predicted representation $\left\{x_n^{t+1}\right\} \equiv \mathcal{P}\left(\left\{x_n^t\right\}, a^{t+1}\right)$, this leads to an open-loop controller which does not update the estimates based on the new observed image $\hat{I}^{t+1}$ available after taking action a^{t+1}.

5. Since the forward model is not perfect, the predicted trajectory will drift.

6. To solve this problem, they propose to additionally learn a correction model $\mathcal{C}$ that can update the predicted location based on the new observation image $\hat{I}^{t+1}$.

7. Denote $I[b]$ as the region cropped on image I specified by the location b. Given the initial crop $\hat{I}^0\left[\hat{b}_n^0\right]$ to identify the obj being tracked, and the predicted location cropped on the new observed image $\hat{I}^{t+1}\left[b_n^{t+1}\right]$, $\mathcal{C}$ regresses the residual Δb_n^{t+1} such that $b_n^{t+1} + \Delta b_n^{t+1}$ approximates $\hat{b}_n^{t+1}$ and recenters the cropped region to objs.

8. They train this model using random jitters around the ground truth boxes on the same training data used to learn the forward model.

6 Experiments

1. The experiments seem to involve understanding under what experimental setting the baseline model would fail and try to make their approach works in these setting.

7 Questions

1. What do they mean they train the correction model using random jitters around the ground truth boxes on the same training data used to learn the forward model?

2. It is unclear how the reconstruction loss prevents trivial solution? What do they mean by trivial solution? It seems the reconstruction loss is just training the decoder.

3. How do they find the initial state representation of the initial observation image I^0.

4. Is the choice of the norm in the reconstruction and prediction loss chosen with trial and error? Is there any good reason for this particular choice of norm ?

5. In the loss function for CEM, how do they know the correspondence between the predicted obj location and the goal obj location? If they do not know this, then the CEM obj does not make sense since whether the loss function is small depends on whether the features of the same object is compared between the prediction and goal.

6. What is the black thingy in Figure 5?