# 1 Summary

1. 3 separate components, perception, physics and rendering are jointly trained.
2. Assume availability of the segmentation mask for object in a scene.
3. The trained module can successfully be used to predict action sequence for block stacking.

# 2 Perception Module

1. Given an image, they assume access to a segmentation of the input $\mathbf{S} = \{s_k\}\, k = 1 \ldots N$.
2. Each segment is supposed to contain one single object.
3. An encoder is applied to each object to obtain object representation vectors $\mathbf{O} = \{o_k\}_{k=1 \ldots N}$.

# 3 Physics Module

1. The physics module contain 2 learnt subcomponent: a unary $f_{\text{trans}}$ and a binary fun $f_{\text{interact}}$.
2. The output fo the physics predictor for object $o_k$ is denoted $\bar{o}_k = f_{\text{trans}}\left(o_k\right) + \sum_{j \neq k} f_{\text{interact}}\left(o_k, o_j\right) + o_k$.
3. Let $\overline{\mathbf{O}} = \{\bar{o}_k\}_{k=1 \ldots N}$.
4. Let $\overline{\mathbf{O}} = f_{\text{physics}}(\mathbf{O})$ denotes the forward pass of the physics module.
5. They only predict the steady-state conf. of objs as a result of simulating physics indefinitely.

# 4 Rendering Engine

1. They argue that a challenge is in designing a fun which constructs a single image from an entire collection of objs.
2. The learned renderers consist of 2 networks.
3. The first preds an image independently for each obj vector rep.
4. The second preds a single-channel heatmap for each obj.
5. The composite scene image is a weighted average of all of the obj-specific renderings, where the weights are from the negative of the predicted heatmaps.

# 5 Learning obj representation

1. The perception, physics and rendering modules are jointly trained on an image reconstruction and prediction task.
2. Their training data consists of image pairs $(I_0, I_1)$ depicting a collection of objs on a platform before and after a new obj has been dropped. ($I_0$ shows one obj in mid-air, as if being held in place before being released.)
3. Given the observed segmented image $\mathbf{S}_0$, they predict obj representations using the perception module $\mathbf{O} = f_{\text{percept}}\left(\mathbf{S_0}\right)$ and their time-evolution using the physics module $\overline{\mathbf{O}} = f_{\text{physics}}(\mathbf{O})$.
4. The rendering module then predicts an image from each of the obj representations: $\hat{I}_0 = f_{\text{render}}(\mathbf{O}), \hat{I}_1 = f_{\text{render}}(\overline{\mathbf{O}})$.
5. $\mathcal{L}_{\text{percept}}(\cdot) = \mathcal{L}_2\left(\hat{I}_0, I_0\right) + \mathcal{L}_{\text{VGG}}\left(\hat{I}_0, I_0\right), \mathcal{L}_{\text{physics}}(\cdot) = \mathcal{L}_2\left(\hat{I}_1, I_1\right) + \mathcal{L}_{\text{VGG}}\left(\hat{I}_1, I_1\right)$
6. $\mathcal{L}_{\text{render}}(\cdot) = \mathcal{L}_{\text{percept}}(\cdot) + \mathcal{L}_{\text{physics}}(\cdot)$

# 6 Planning with learned module

1. The planning procedure does not use the rendering function and only uses errors in the learned obj representation space.

2. The planning step is greedy, actions are selected sequentially.

3. Each action is selected to find a placement of an obj such that the obj is closest to one of the obj in the goal img.

# 7 Experiments

1. Their approach outperforms planning based on method that uses the oracle physics simulator, but operates directly in pixel space.

2. The section on experiments on real robot seemed rushed, some of the explanation is not clear and also use additional algorithmic improvements to make it work.