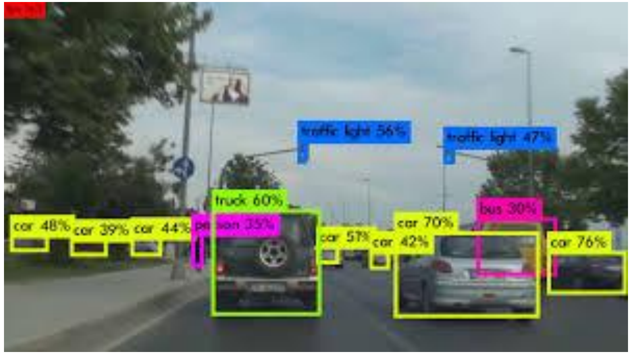
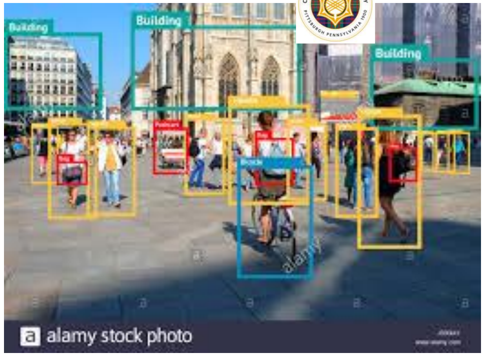
Data Dreaming for Object Detection: Learning Object-Centric State Representations for Visual Imitation

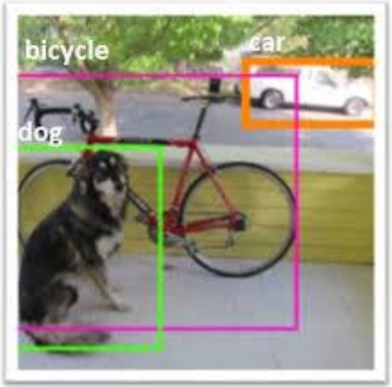
Maximilian Sieb

Katerina Fragkiadaki

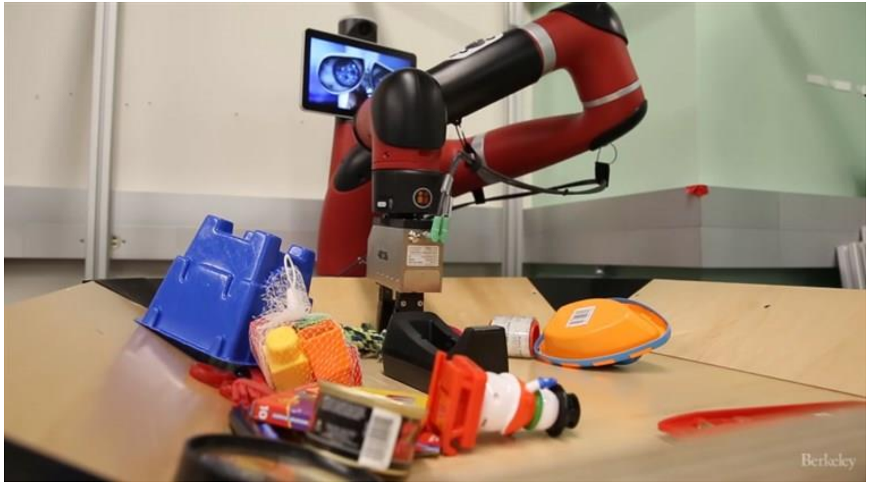
Computer Vision works well for ImageNet and COCO Categories

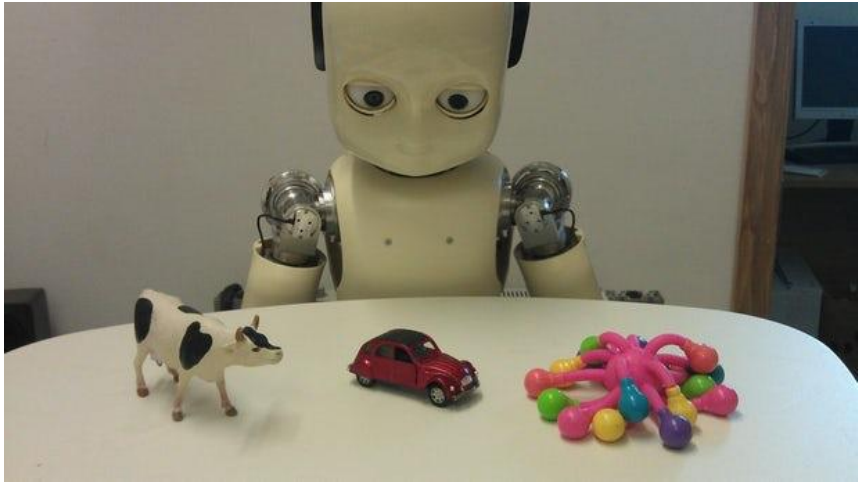






Why we do we not see this kind of performance on robots?





Problem: Object categories are different

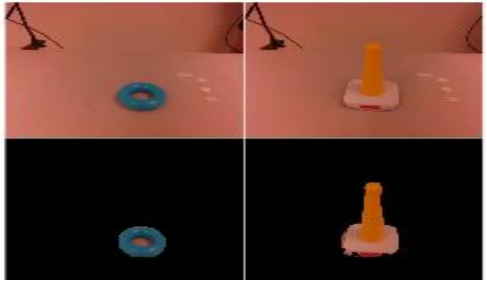
Proposed method: Data Dreaming for Object-Detection

- On-the-fly data augmentation with synthetically generated data
- Robust per-instance object detectors
- May not work on *all* scene variations, but does very well on the particular environment and **its specific variations**

Object Detector Pipeline

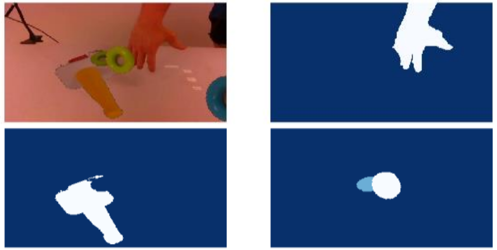
1. Obtaining object masks

- Background subtraction gives ground truth object masks
- Requires single un-occluded image of each relevant object



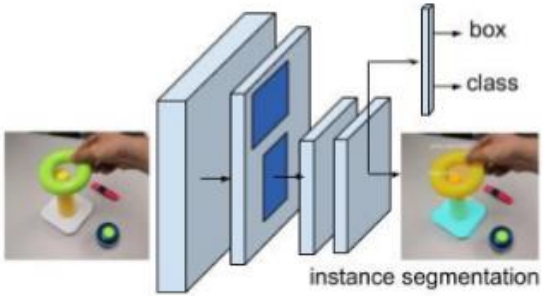
2. Creating synthetic data

- Massive data augmentation of ground truth
- Overlay images with random transformations and occlusions to obtain ground truth occluded masks

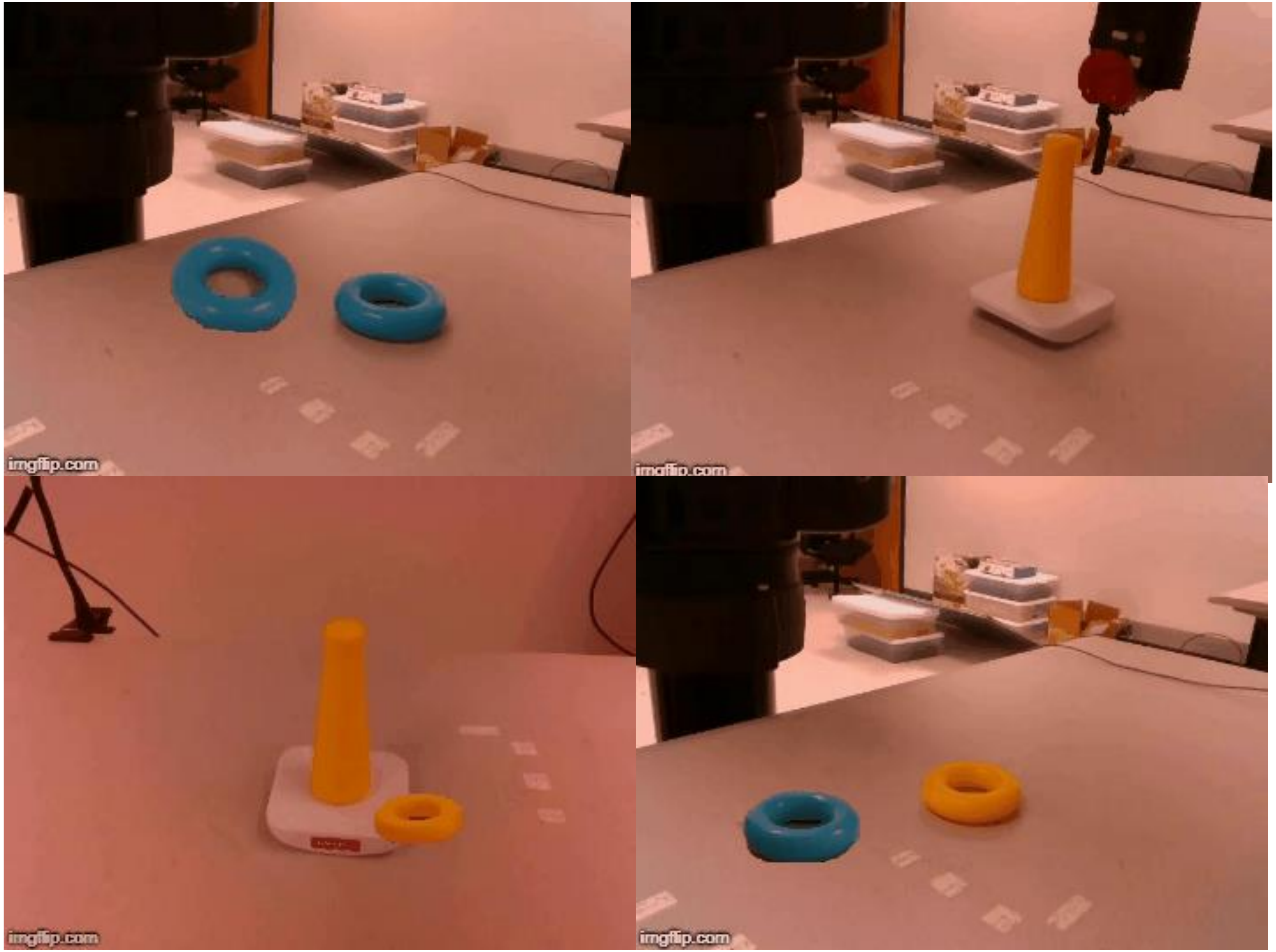


3. Training detectors

- Mask R-CNN* as architecture of choice for instance segmentation



*K. He, G. Gkioxari, P. Doll´ar, and R. B. Girshick. Mask R-CNN. CoRR, abs/1703.06870, 2017.

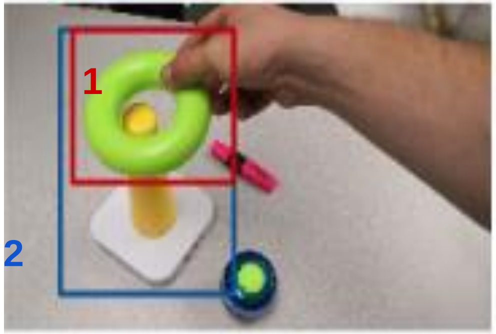

imgflip.com
imgflip.com
imgflip.com
imgflip.com

State Representation of Scene and Objects

Proposed Method:
Use bounding box to focus on relevant objects in the scene

a. Spatial features capture relations between different objects (pairwise difference of box x, y-centroid & depth-value)

b. Visual/Appearance features capture object-specific state



$$\phi(x_d) = \begin{matrix} \phi^{spatial}, \\ \phi_1^{visual}, \\ \phi_2^{visual} \end{matrix}$$

Appearance Feature Encoding

- We use deep features extracted from the RGB channels of each bounding box
- We train these features with unsupervised losses, e.g. time-constrastive loss (see below), multi-view invariance loss etc. to enforce focusing on the relevant objects

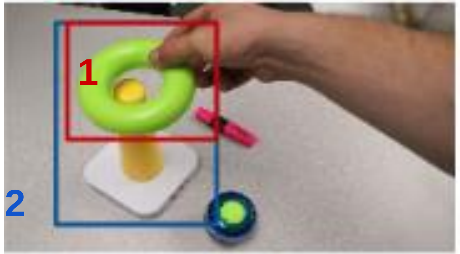
Appearance Feature Encoding – Previous Work

- Previous works: Focus on full-frame encodings

Problem: Large amounts of training data required to learn good features

Our method:

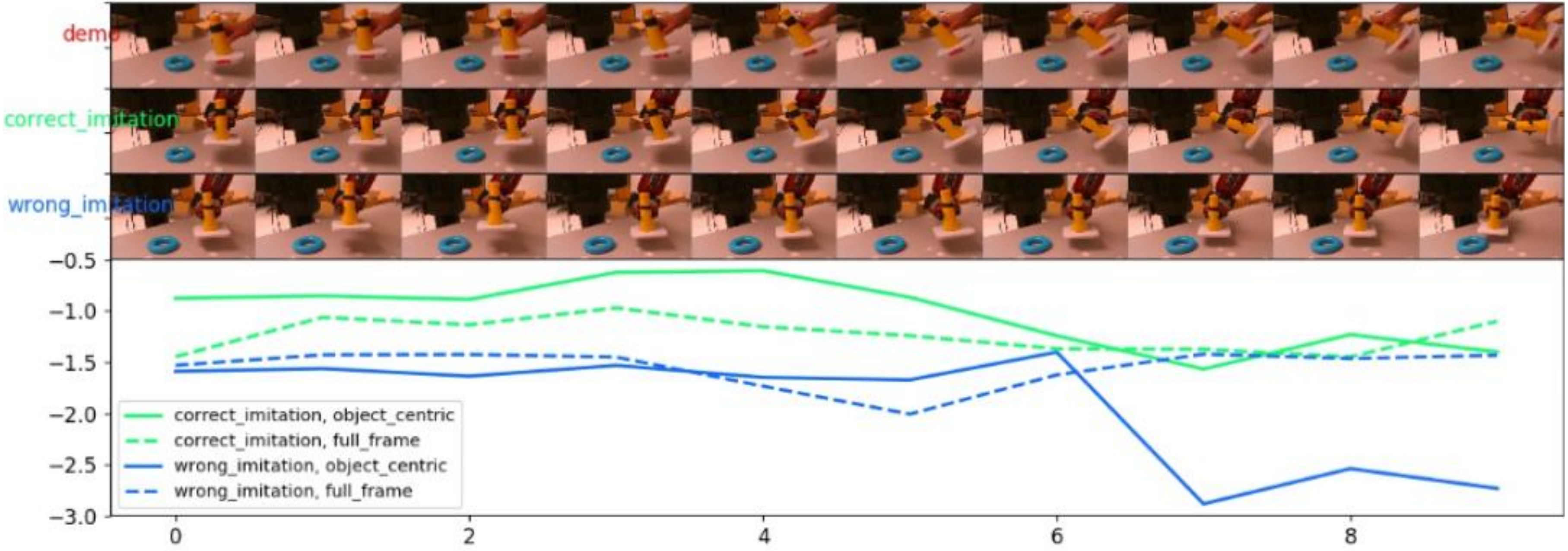
a. Spatial features capture relations between different objects

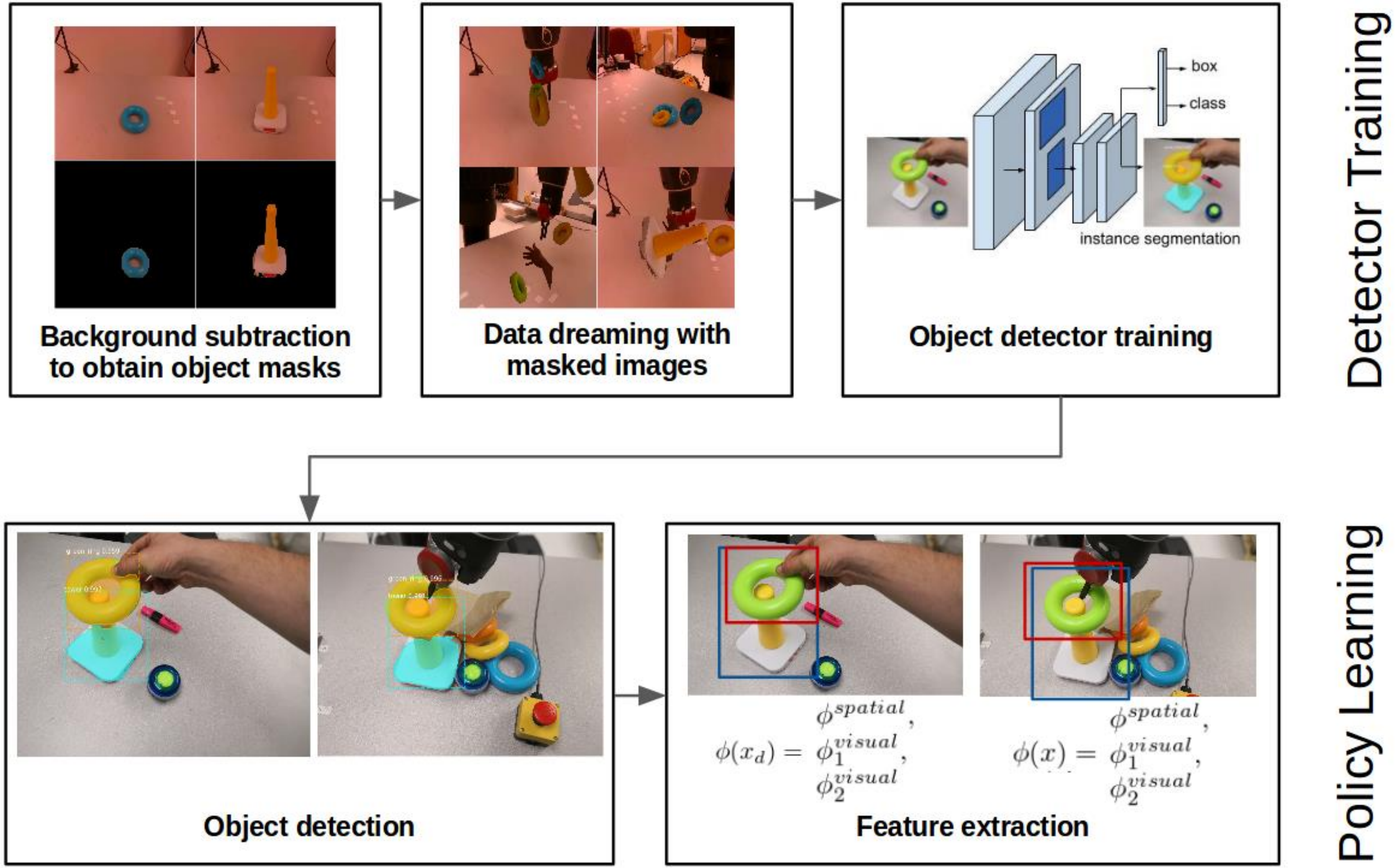


b. Visual/Appearance features capture object-specific state

Discriminative Rewards through Box-Centered Features

- Able to distinguish between correct imitations of target object
- Full-frame encoding does not capture change in scene appropriately




Background subtraction to obtain object masks
Data dreaming with masked images
box
class
instance segmentation
Object detector training
Detector Training
Object detection
$\phi(x_d) = \phi^{spatial}, \phi_1^{visual}, \phi_2^{visual}$
$\phi(x) = \phi^{spatial}, \phi_1^{visual}, \phi_2^{visual}$
Feature extraction
Policy Learning

Model-Based Control for Sample-Efficient Learning

- We use a trajectory optimization method (PILQR*)
- Using visual features as the state representation x, minimize visual dissimilarity between demonstrator and imitator as cost c

Given: $\bar{x}_0,\ \phi(.)$

For $t = 0, 1, 2, ..., T$

- Solve

$$\min_{x,u} \sum_{k=t}^{T} c_k(x_k, u_k)\,, \qquad c_k = \alpha \frac{1}{2} \|\phi(x_k) - \phi(x_k^{demo})\|_2^2$$

$$\text{s.t. } x_{k+1} = f(x_k, u_k), \quad \forall k \in \{t, t+1, ..., T-1\}$$

$$x_t = \bar{x}_t$$

- Execute u_t

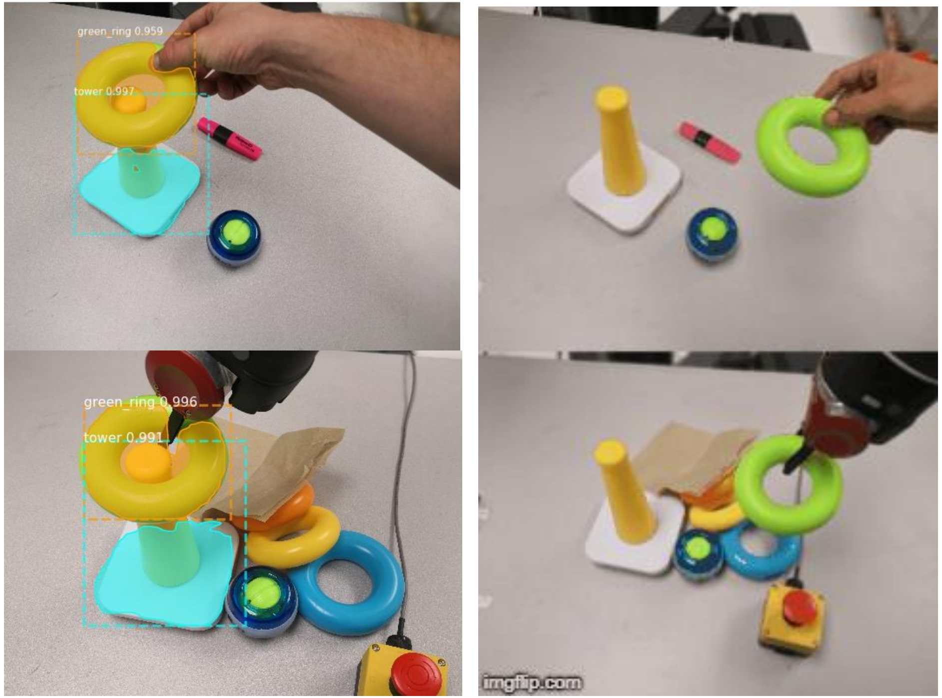
- Observe resulting state $\bar{x}_{t+1}$

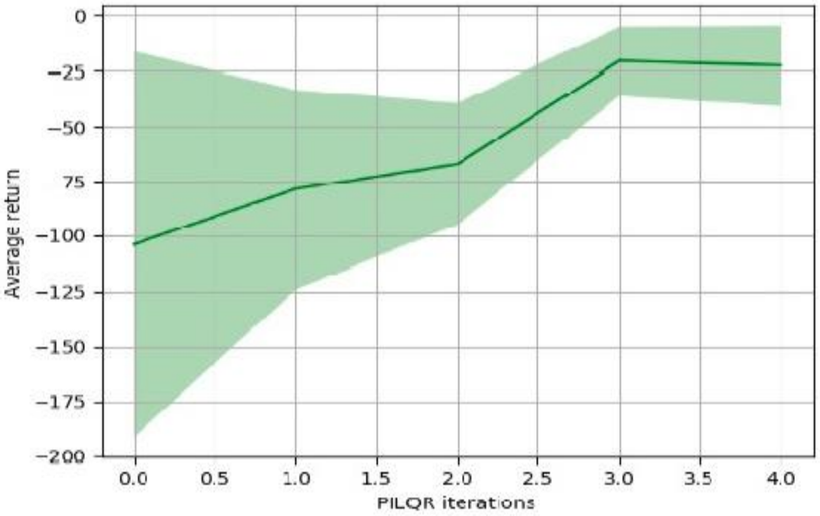
*Y. Chebotar, K. Hausman, M. Zhang, G. Sukhatme,S. Schaal, and S. Levine. Combining Model-Based and Model-Free Updates for Trajectory-Centric Reinforcement Learning. 2017.

Model-Based Control for Sample-Efficient Learning

Demonstration

Learned policy





Conclusion

We can leverage on-the-fly training of object detectors to obtain a structured state representation

Benefits:

- **Visual robustness**: Robust towards visual changes in scene and partial occlusions
- **Graph-like state representation**: Establishes interpretable relations between objects
- **Sample-efficient policy learning**: Hard visual attention allows for learning a policy in only a few iterations, requiring only a few real-world trajectory rollouts

Data Dreaming for Object Detection:
Learning Object-Centric State Representations for Visual Imitation