The Curious Robot: Learning Visual Representations via Physical Interactions

Lerrel Pinto, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, and Abhinav Gupta

Russell Kaplan and Raphael Palefsky-Smith

Right now...

We train ConvNets on millions of static images with semantic labels.

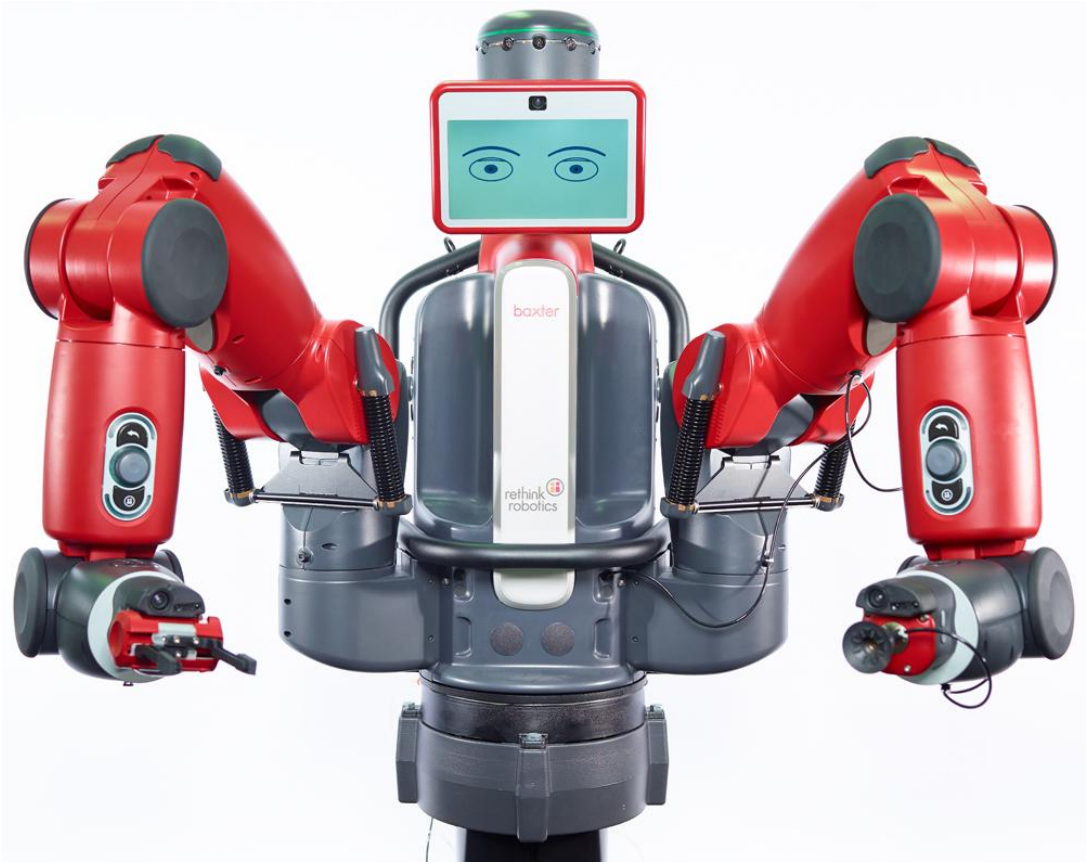
But that’s not how humans learn.

Insight

Humans learn by *interacting* with the physical world.

Example: babies push, poke, and chew objects to learn more about them.

Why not let our ConvNets do the same?



Why not let our ConvNets do the same?

Goal

Learn **good visual representations** from many physical interactions, without semantic labels.

Related Work

Two threads.

1. How do you learn a representation in an unsupervised way?
2. How do you interact with the world for learning?

Related Work: unsupervised learning

Generative

- Deep Belief Networks [Hinton et al., 2006]
- Autoencoders, VAEs [Kingma & Welling, 2013]
- GANs [Goodfellow et al., 2014]

Discriminative

- Supervision from context [Doersch et al., 2014]
- Ego-motion [Irani et al., 1994]

Related Work: robotic tasks

Executing robotic tasks

- Grasping [1, 2]
- Pushing [3, 4]
- Tactile sensing [5]
- Identity vision [6, 7]

Vision and deep learning for robotics

- Grasp regression [8]
- Task policy learning [9]

Citations in appendix

What's different here?

1. Learning from **active manipulations of the world** (v.s. passive unsupervised learning with labeled images)
2. **Reversal of traditional robotics** research**:** instead of using *vision* to do *robotic tasks* better, authors use *robotic tasks* to learn *better visual representations*
    a. This is the first time robotic tasks have been used specifically to learn better visual representations, according to authors

Setup

- Use a Baxter Robot to carry out tasks and record data
- Train ConvNet on that data, with supervisory signal from four physical interaction tasks:
  - Grasp
  - Push
  - Poke
  - Identity (pose-invariance)
- Analyze quality of the ConvNet’s learned representations

Task: Grasp

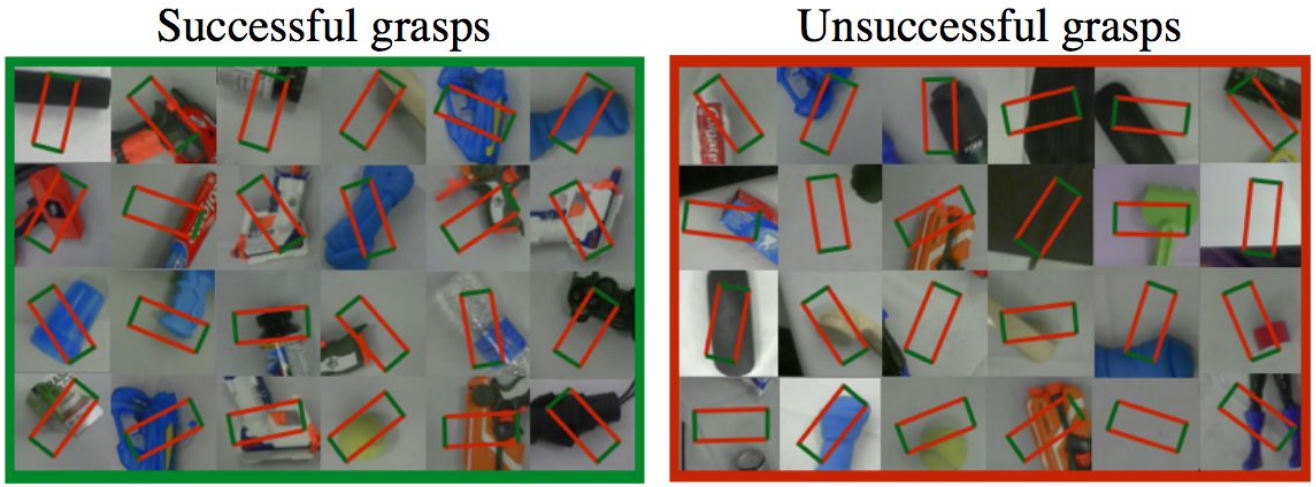
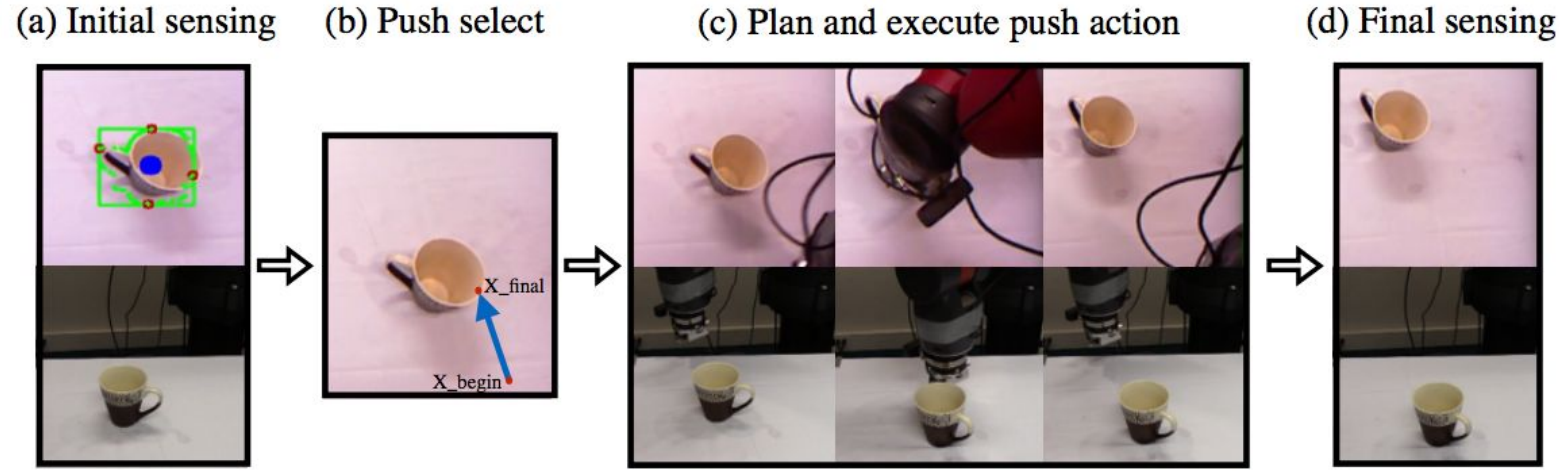


Fig. 2. Examples of successful (left) and unsuccesful grasps (right). We use a patch based representation: given an input patch we predict 18-dim vector which represents whether the center location of the patch is graspable at 0°, 10°, ... 170°.

Task: Planar push

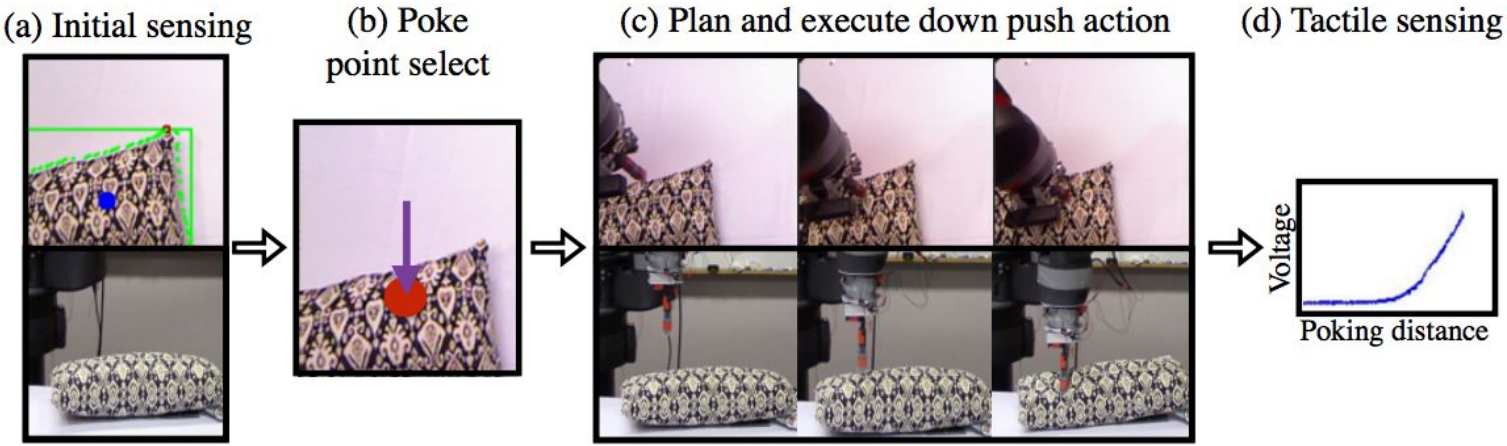


$X_{begin} = (x_{begin}, y_{begin}, z_{begin})$

$X_{end} = (x_{end}, y_{end}, z_{end})$

Given I_{begin}, I_{end}, predict regression: $\{X_{begin}, X_{end}\}$

Task: Poke



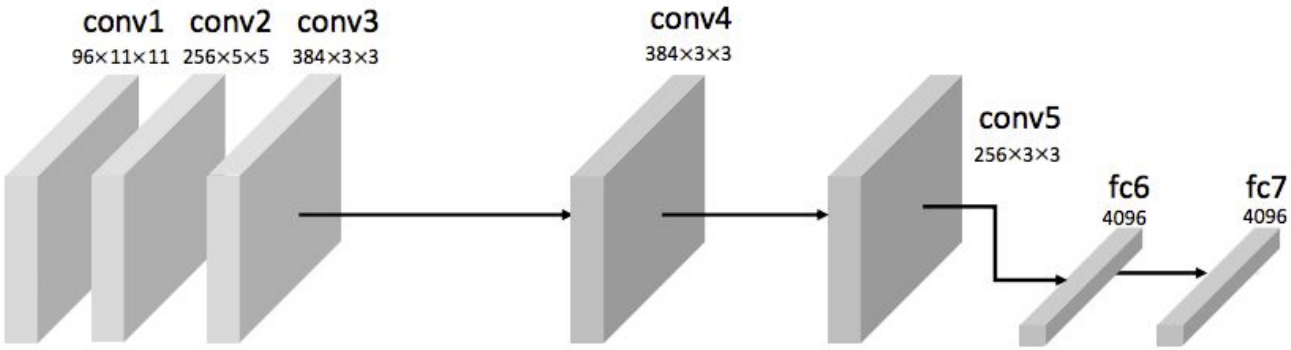
Use a tactile sensor to measure pressure over the poke action. Try to predict pressure v.s. poking distance curve.

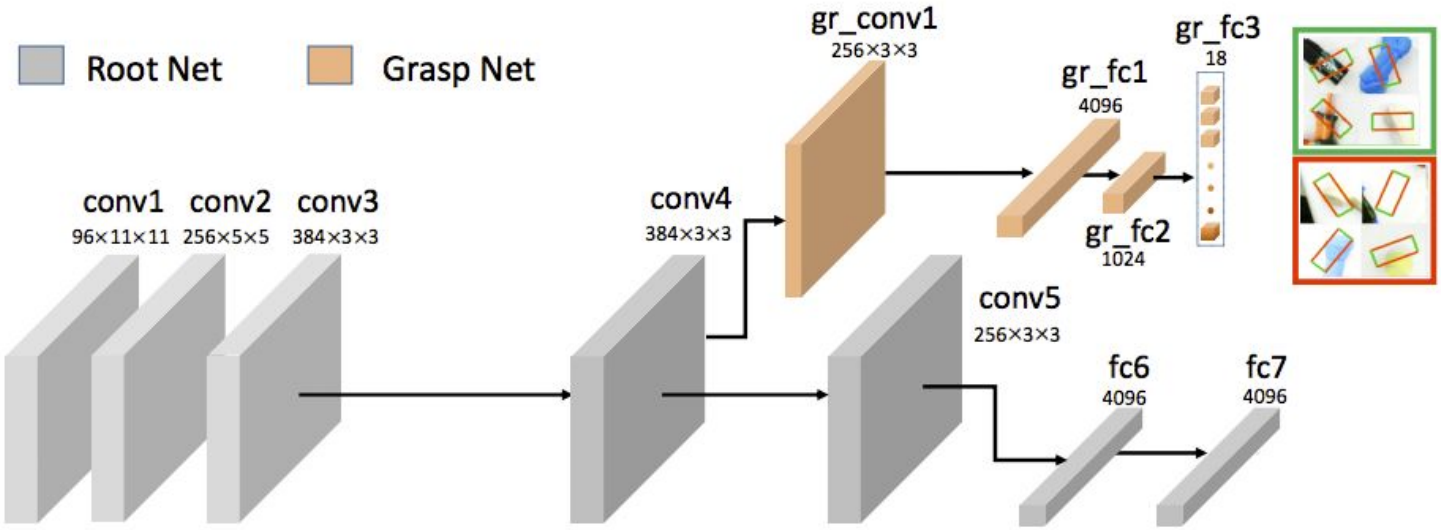
Task: Identity (Pose Invariance)

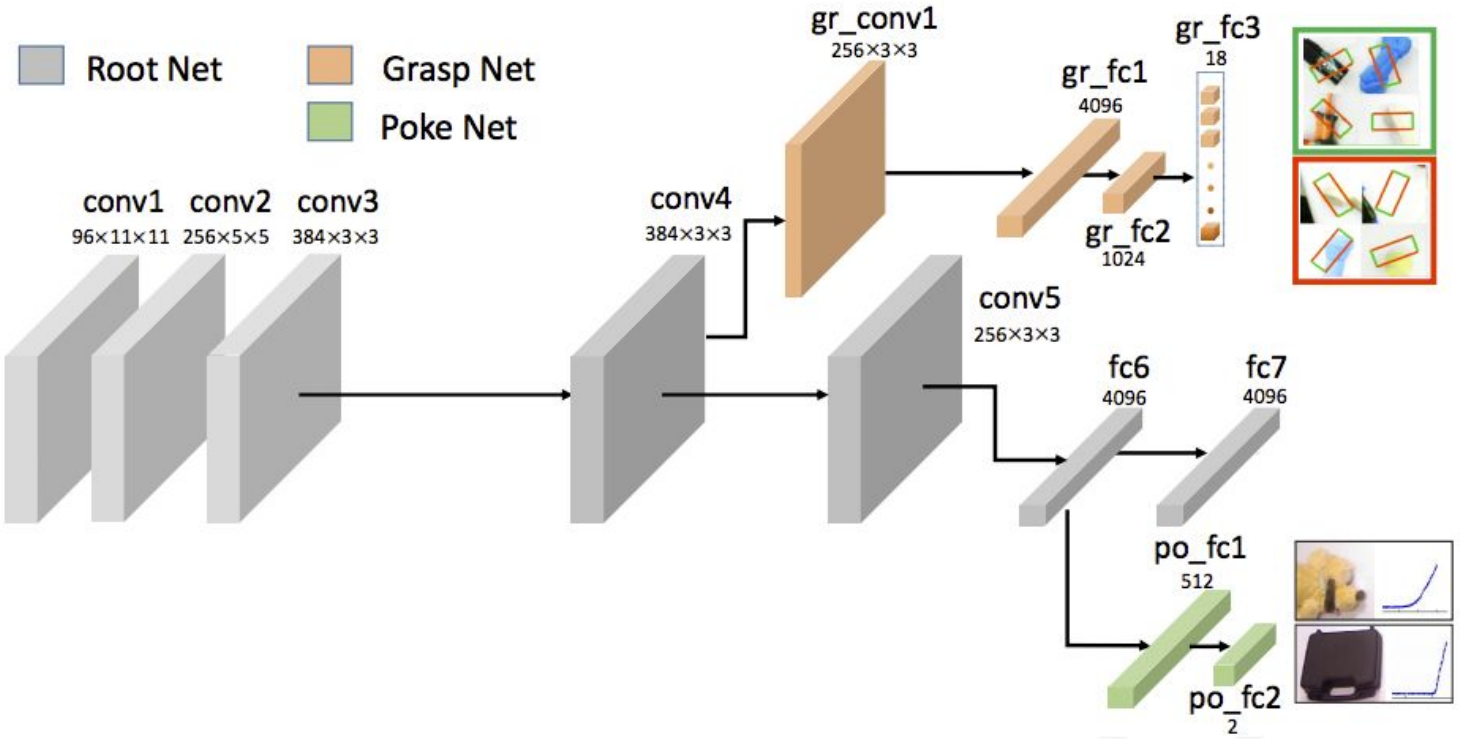
Force FC-7 feature embeddings to be similar if images are from the same task.

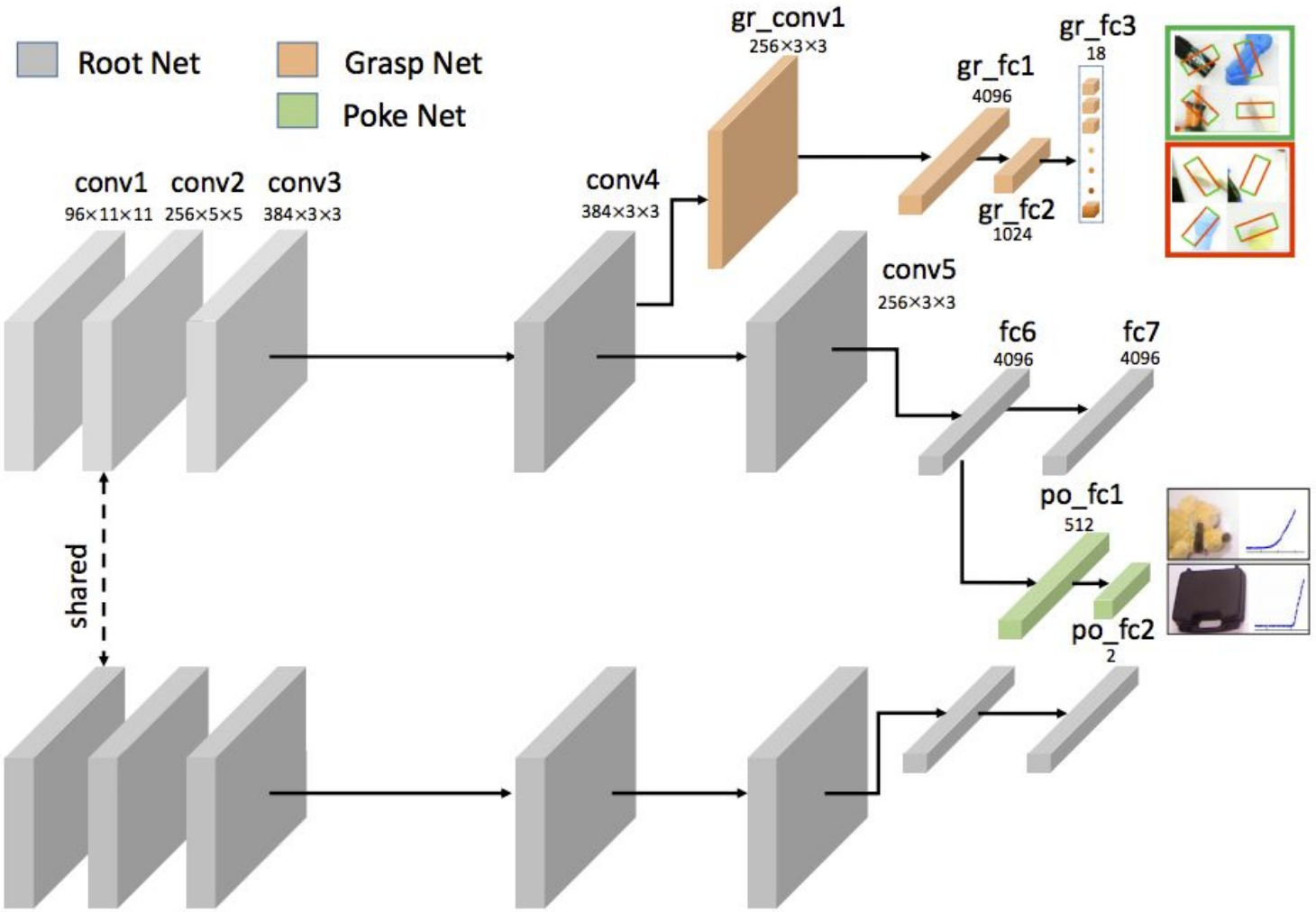
Network Architecture

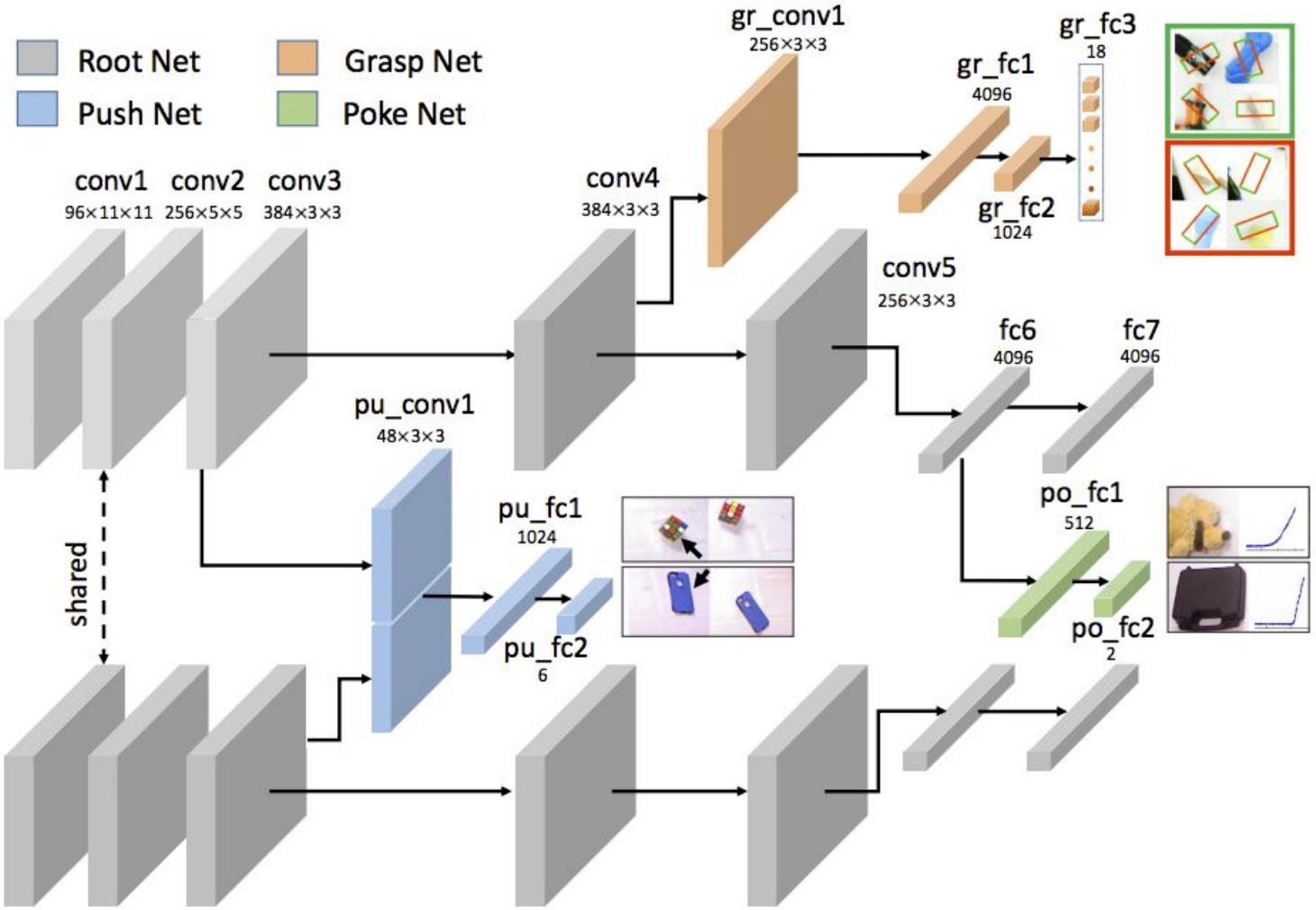
Root Net

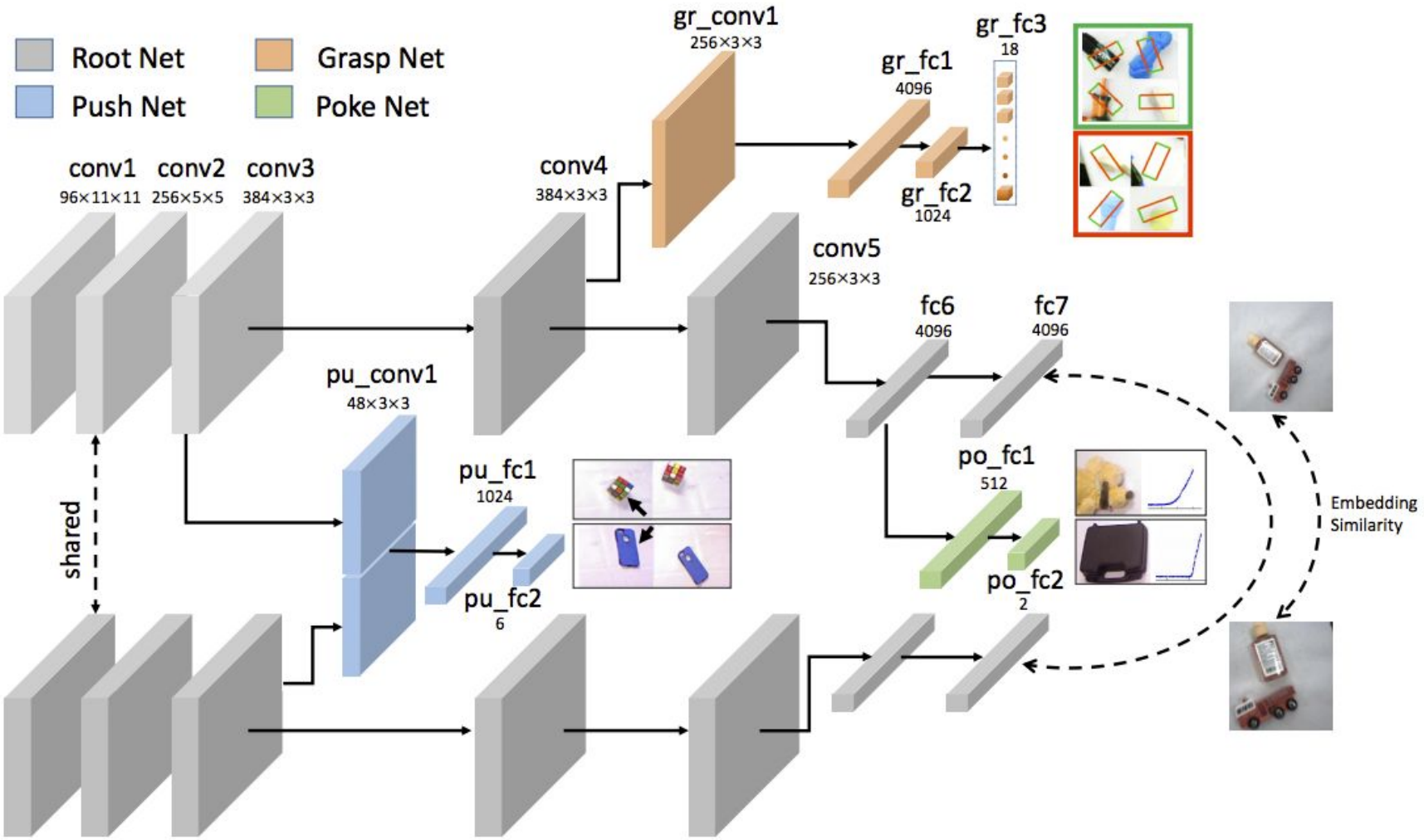

conv1
96×11×11
conv2
256×5×5
conv3
384×3×3
conv4
384×3×3
conv5
256×3×3
fc6
4096
fc7
4096


Root Net
Grasp Net
conv1
96×11×11
conv2
256×5×5
conv3
384×3×3
conv4
384×3×3
gr_conv1
256×3×3
gr_fc1
4096
gr_fc2
1024
gr_fc3
18
conv5
256×3×3
fc6
4096
fc7
4096


Root Net
Grasp Net
Poke Net
conv1
96×11×11
conv2
256×5×5
conv3
384×3×3
conv4
384×3×3
gr_conv1
256×3×3
gr_fc1
4096
gr_fc2
1024
gr_fc3
18
conv5
256×3×3
fc6
4096
fc7
4096
po_fc1
512
po_fc2
2


Root Net
Grasp Net
Poke Net
conv1
96×11×11
conv2
256×5×5
conv3
384×3×3
conv4
384×3×3
gr_conv1
256×3×3
gr_fc1
4096
gr_fc2
1024
gr_fc3
18
conv5
256×3×3
fc6
4096
fc7
4096
po_fc1
512
po_fc2
2
shared


Root Net
Grasp Net
Push Net
Poke Net
conv1
96×11×11
conv2
256×5×5
conv3
384×3×3
conv4
384×3×3
gr_conv1
256×3×3
gr_fc1
4096
gr_fc2
1024
gr_fc3
18
conv5
256×3×3
fc6
4096
fc7
4096
pu_conv1
48×3×3
pu_fc1
1024
pu_fc2
6
po_fc1
512
po_fc2
2
shared

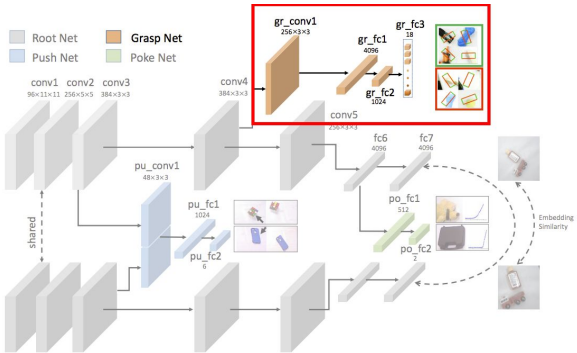

Root Net
Grasp Net
Push Net
Poke Net
conv1
96×11×11
conv2
256×5×5
conv3
384×3×3
conv4
384×3×3
gr_conv1
256×3×3
gr_fc1
4096
gr_fc2
1024
gr_fc3
18
conv5
256×3×3
fc6
4096
fc7
4096
pu_conv1
48×3×3
pu_fc1
1024
pu_fc2
6
po_fc1
512
po_fc2
2
shared
Embedding
Similarity

Training Procedure

- Stage 1: Grasp Only
  - Initialize root network (Gaussian) up to conv4
  - Train for 20k iterations on only the Grasp task
- Stage 2: All Tasks
  - Minibatches of 128 examples for each of the four tasks
  - Weights for task-specific networks updated immediately
  - Gradients for root network are averaged between the four tasks, and applied after all four have run

Grasp Loss (40k examples)

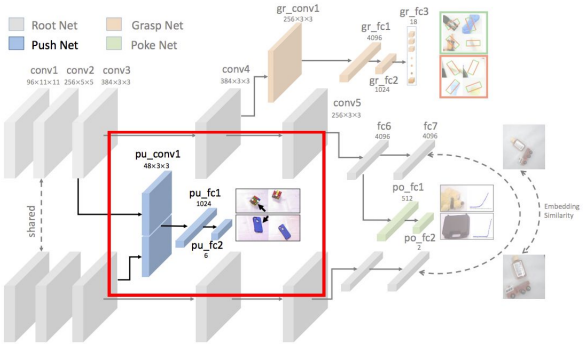
- Binary classification problem: will a centered grasp with angle θ = 0º, 10º, …, 170º be successful?
- Input: image of object
- Predict: 18 binary classifications for 10˚ bins from 0-180˚

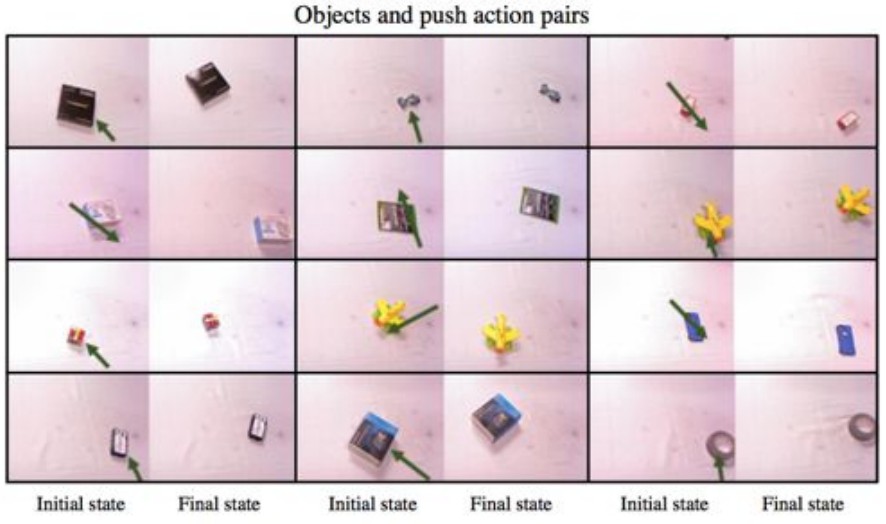


$$L = \sum_{i=1}^{B} \sum_{j=1}^{N=18} \delta(j, \theta_i) \cdot \text{softmax}(A_{ji}, l_i)$$

Push Loss (5k examples)

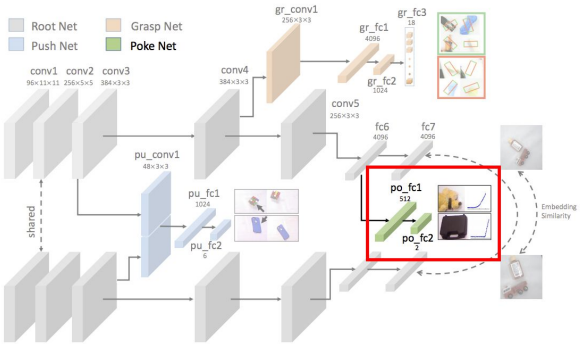
- Given a starting image and an ending image, predict the push parameters needed to perform that action
- Input: two images, before and after push, run through Siamese network
- Predict: push parameters (x_{start}, y_{start}, x_{final}, y_{final}, z)
- Loss: Mean-Squared Error

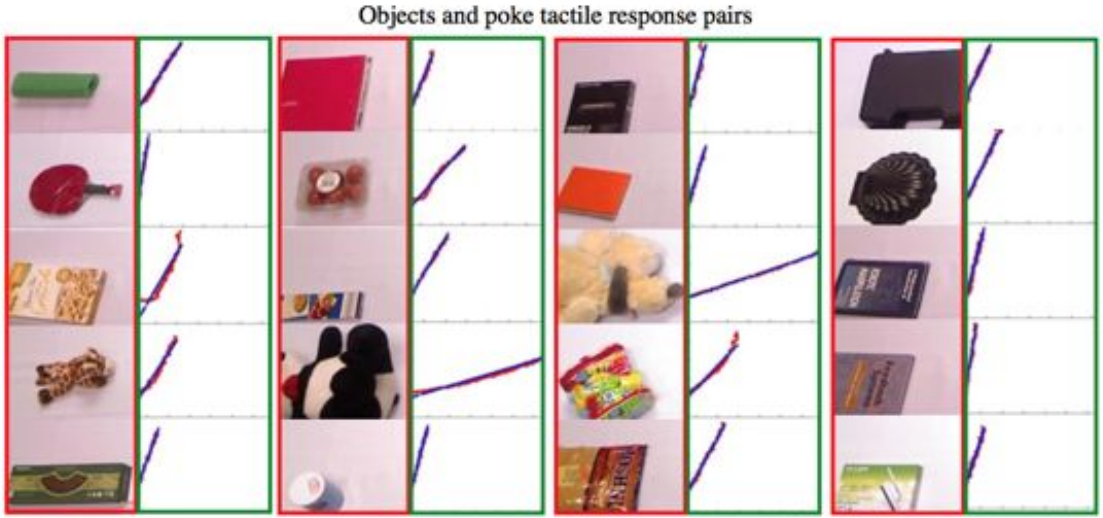




Poke Loss (1k examples)

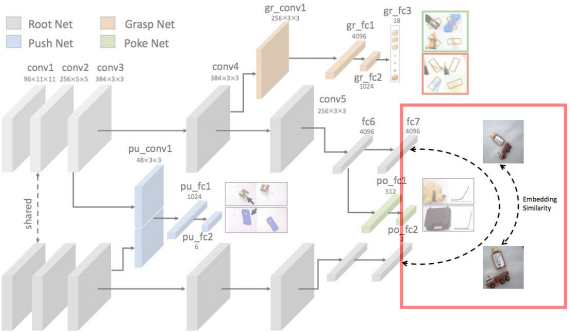
- Tactile force response appears linear, predict the y-intercept and slope
- Input: image of object
- Predict: y-intercept and slope of force curve
- Loss: Mean-Squared Error

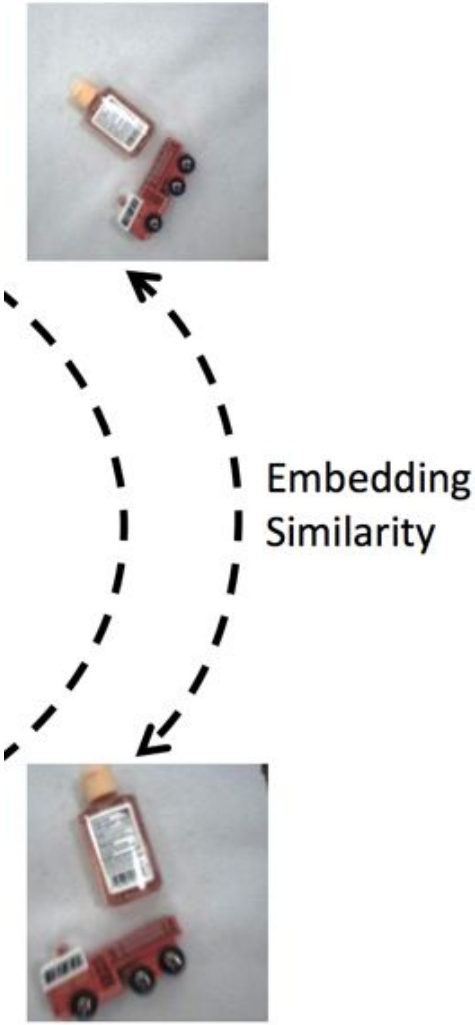




Identity Loss (84k pairs)

- Enforce embedding similarity in fc7 between all images of the same object, leading to pose invariance (Siamese nets)
- Input: images of the same object in different poses
- Output: 4096-dimensional activations
- Loss: Cosine Similarity



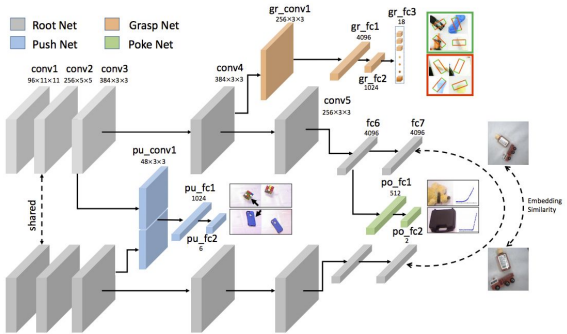


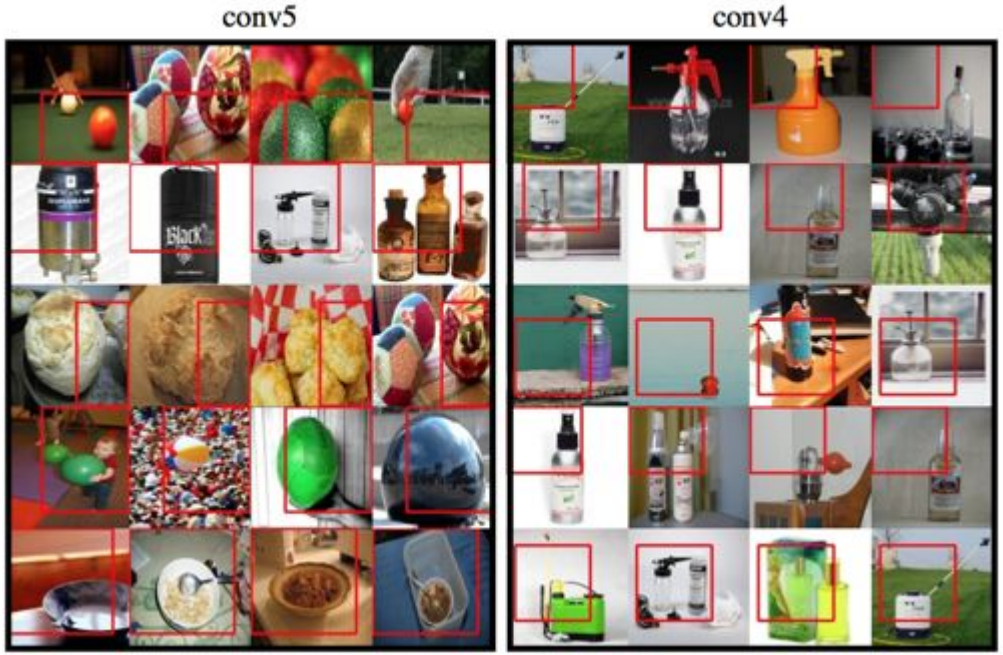
Key Experiments

- Maximally-Activating Images
- Nearest Neighbor
- Image Classification
- Image Retrieval
- Task Ablation Analysis

Maximally-Activating Images

- From 2500 ImageNet examples, find the images that maximally activate conv4 and conv5
- Shape-based





Nearest Neighbor

- Seems to also rely largely on shape attributes

Image Classification

- Dataset: 25 household object classes from ImageNet, 2500 total images
- Additional datasets: UW RGBD and Caltech-256
- Comparison between…
  - Robot task network **finetuned** with class labels
  - Root network trained only on class labels
  - AlexNet trained on entire ImageNet
  - Autoencoder trained on all robot images

Image Classification (cont'd)

- 10.4% relative boost in accuracy by using robot task training data!

Table 1. Classification accuracy on ImageNet Household, UW RGBD and Caltech-256

	Household	UW RGBD	Caltech-256
Root network trained on robot tasks (**ours**)	0.354	0.693	0.317
Root network trained on identity data	0.315	0.660	0.252

Image Classification (cont'd)

Table 1. Classification accuracy on ImageNet Household, UW RGBD and Caltech-256

	Household	UW RGBD	Caltech-256
Root network with random init.	0.250	0.468	0.242
Root network trained on robot tasks (**ours**)	0.354	0.693	0.317
AlexNet trained on ImageNet	0.625	0.820	0.656
Root network trained on identity data	0.315	0.660	0.252
Auto-encoder trained on all robot data	0.296	0.657	0.280

Image Retrieval

- Use fc7 features to perform retrieval on UW RGBD dataset
- recall@1 is 3% better than ImageNet

Table 2. Image Retrieval with Recall@k metric

	Instance level				Category level			
	k=1	k=5	k=10	k=20	k=1	k=5	k=10	k=20
Random Network	0.062	0.219	0.331	0.475	0.150	0.466	0.652	0.800
Our Network	0.720	0.831	0.875	0.909	0.833	0.918	0.946	0.966
AlexNet	0.686	0.857	0.903	0.941	0.854	0.953	0.969	0.982

Task Ablation Analysis

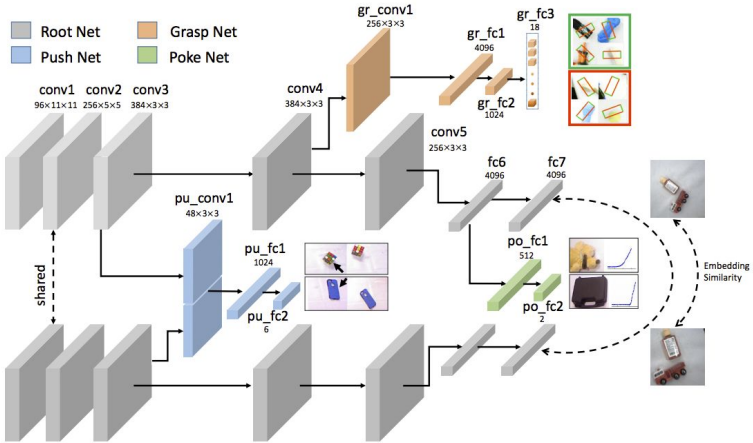
- Re-train the network 4 times, excluding each of the tasks one by one
- Excluding Grasp task leads to largest drop in performance

Table 3. Task ablation analysis on classification tasks

	Household	UW RGB-D	Caltech-256
All robot tasks	0.354	0.693	0.317
Except Grasp	0.309	0.632	0.263
Except Push	0.356	0.710	0.279
Except Poke	0.342	0.684	0.289
Except Identity	0.324	0.711	0.297

Takeaways

- Networks can learn from **active interaction** with the world, rather than just passive labels
- Visual representations learned from physical interactions can **generalize** to other tasks like classification and retrieval
- Despite learning features automatically, this system involves a heavily **handcrafted architecture**



What's Next: Levine et al (Google)

- Reinforcement learning on grasp task
- Key difference: CNN actually controls the robot!
- 14 robots in parallel, shared CNN

Thank You