Combining Self-Supervised Learning and Imitation for Vision-Based Rope Manipulation

Presenter: Salih Hasan Siddiqi
Advisor: Artemij Amiranashvili

Outline

- **Introduction**
- Robot Specification
- Neural Network Training
- Evaluation

Deformable Objects

- Such as ropes and cloths.
- Manipulation skills requires high degree of dexterity.

<u>Examples:</u>

1. Children need help learning to tie their shoes.
2. Folding paper into an origami flower needs practice.

Deformable Objects

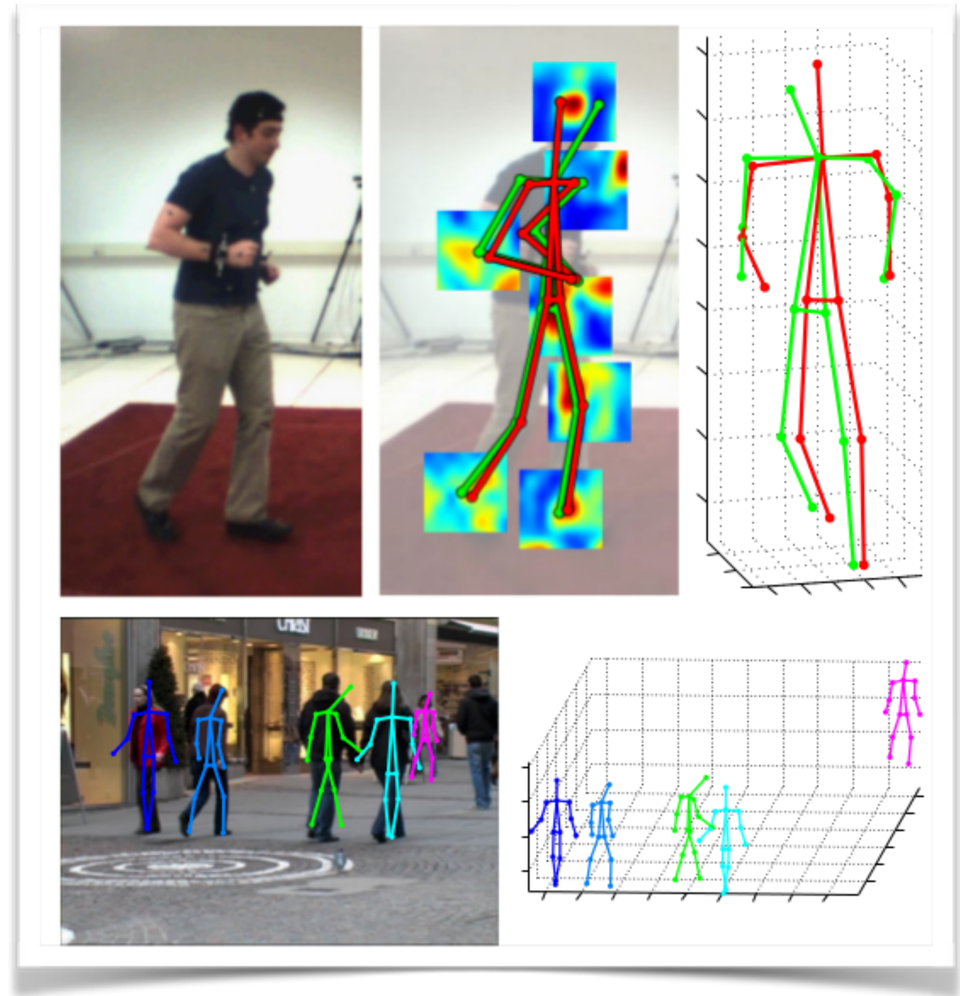
Problems

- Material can shift in unpredictable ways.

- Pose estimation difficult

Reasons:

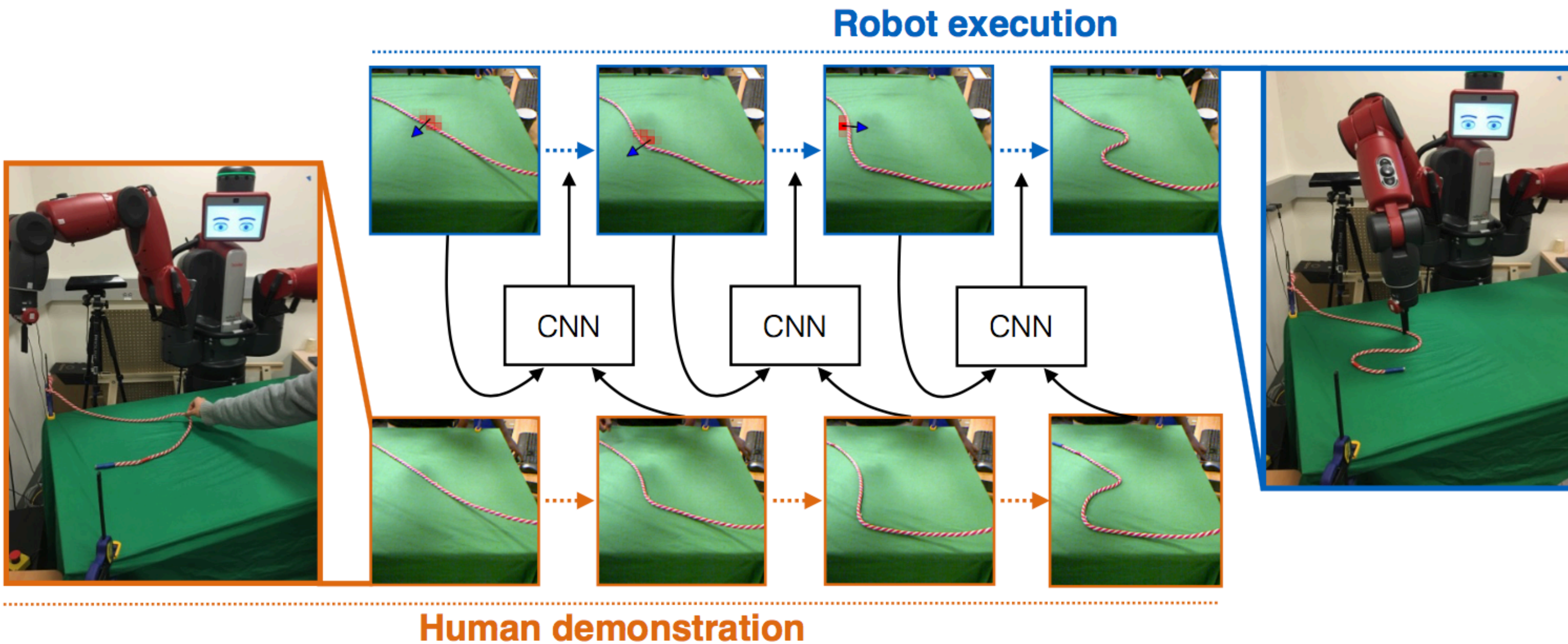
1. Difficult to define degrees of freedom.
2. Lack of enough suitable training data.



Key Idea

- Inspired by how we as humans learn to manipulate deformable objects.
- Humans can learn to “tie a tie” by looking at the picture on the right.
- We know how to perform small manipulations based on our past experience.
- A robot could learn in a similar way.

Key Idea



Key Idea

- Self supervised learning of robot to learn small deformations.
- Human provided demonstration as a high level plan.

Self supervised Robot Learning

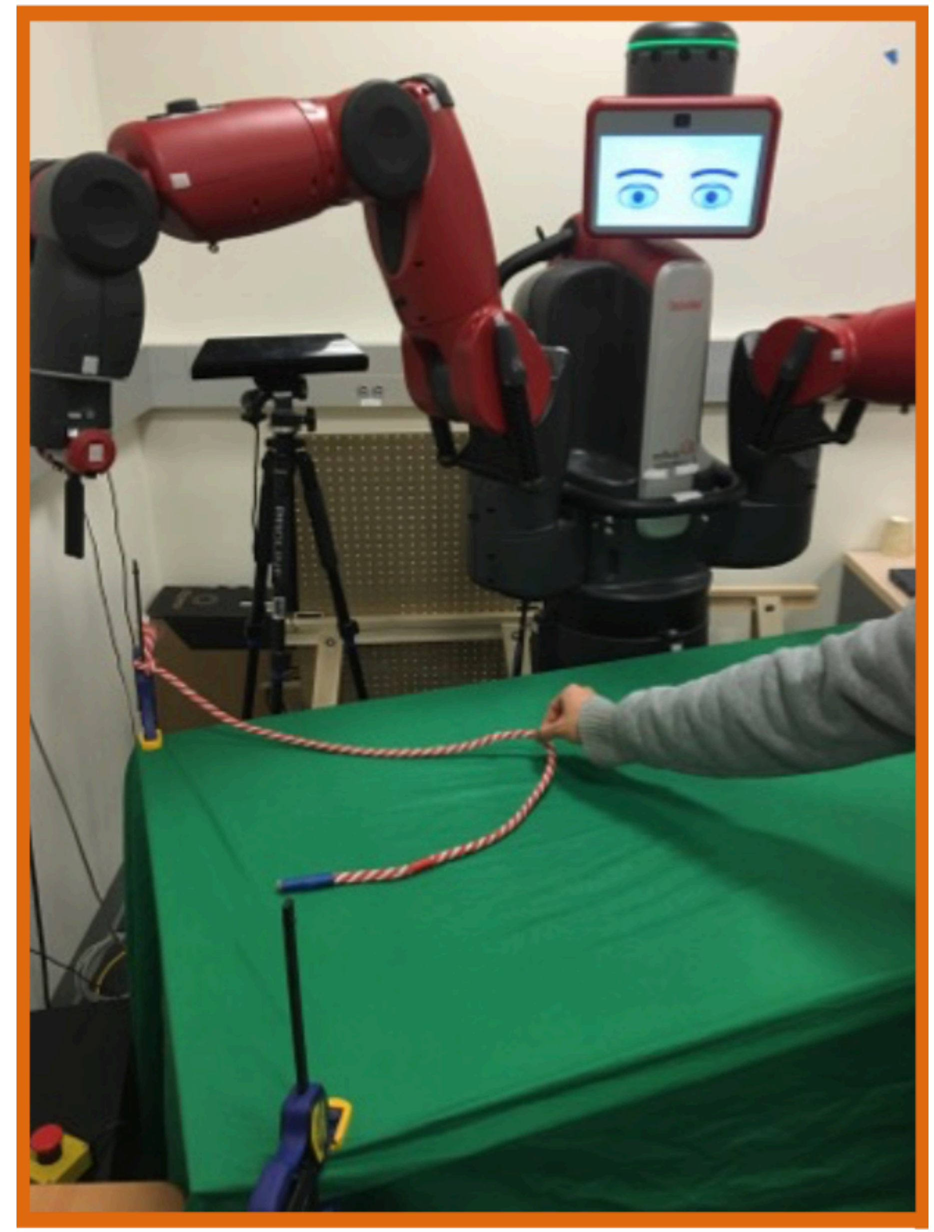
How to Do it

What to do

Human Provided Demonstration

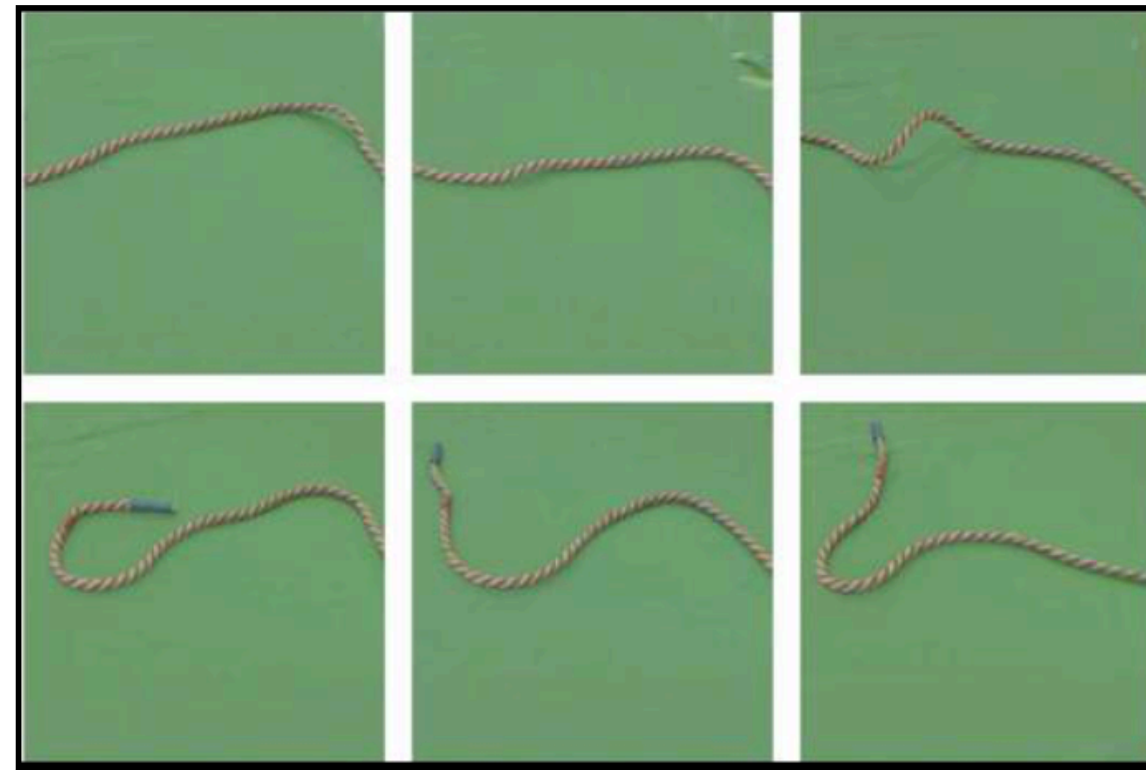
Problems With Previous Approaches

- Modelling for deformable objects is challenging
- Kinematic models fail to capture full variability of the deformable object.
- Direct human Imitation fails in conditions that deviate from those in demonstrations.



Difference with Previous Approaches

- Model free approach with Deep Convolutional Neural Networks.
- Behaviour driven approach.
- Use raw images of rope which provides representational flexibility.
- Generalizable to other deformable objects.

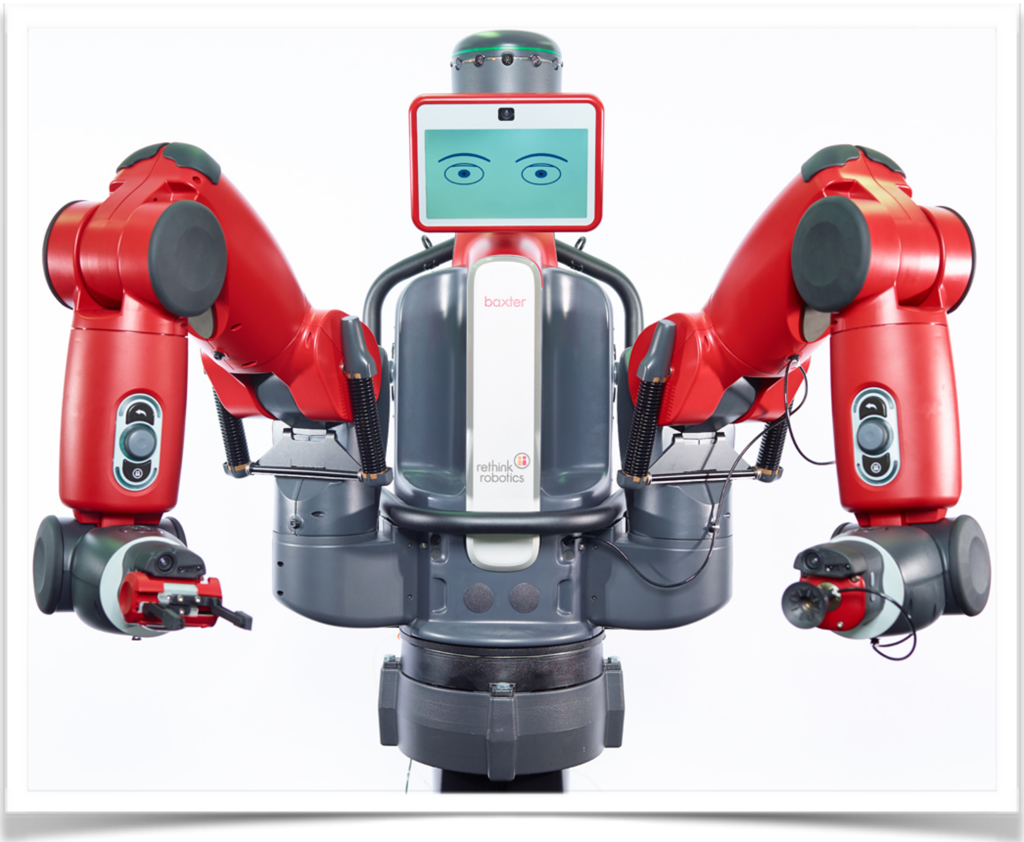


Outline

- Introduction
- **Robot Specification**
- Neural Network Training
- Evaluation

Robot Specification

- **Baxter robot.**



Robot Specification

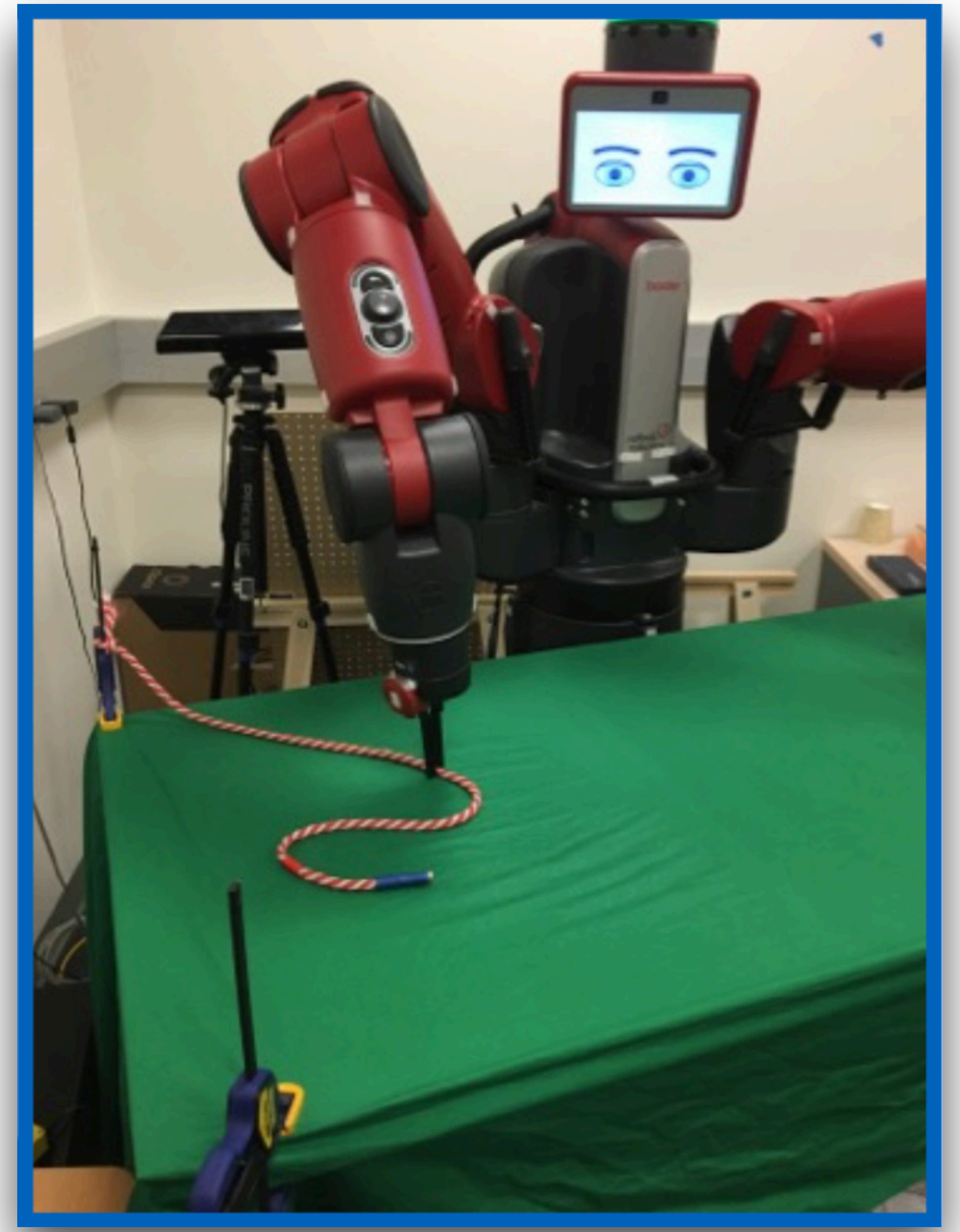
- Baxter robot.

- **One hand with parallel jaw gripper.**

<u>Actions available:</u>
1) Rotate
2) Open
3) Close

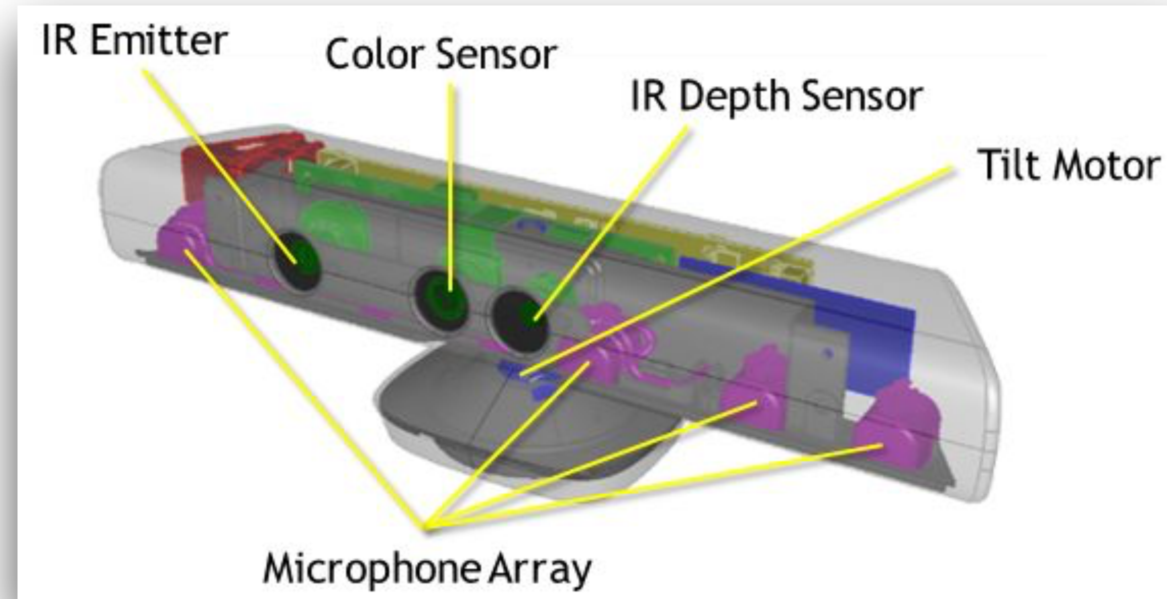
Robot Specification

- Baxter robot.
- One hand with Parallel jaw gripper.
- **Single action primitive**
  1. Pick the rope at (x1, y1)
  2. Drop the rope at (x2, y2)



Robot Specification

- Baxter robot.
- One hand with parallel jaw gripper.
- Single action primitive
  1. Pick the rope at (x1, y1)
  2. Drop the rope at (x2, y2)
- **Visual inputs via RGB channels of Kinect camera.**



Outline

- Introduction
- Robot Specification
- **Neural Network Training**
- Evaluation

Inverse Dynamics Model

- A model that predicts the action **u** that relates a pair of input states is called an inverse dynamics model

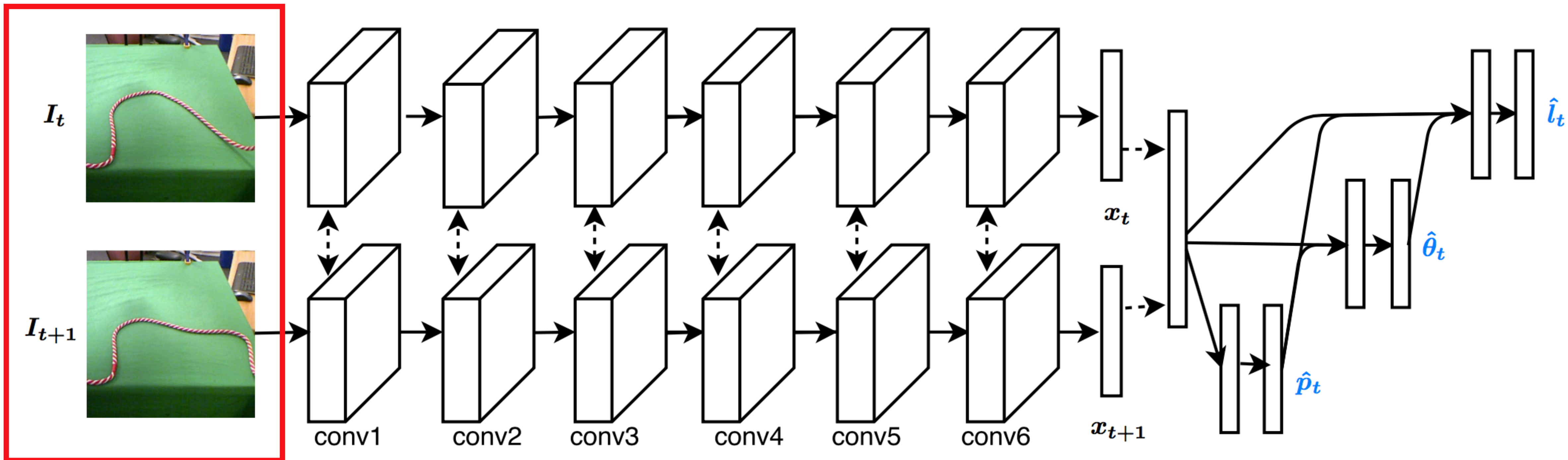
$$\mathbf{u_t=F(I_t,I_{t+1})}$$

- Data autonomously collected by robot is used to train Inverse dynamics model.
- Human provides demonstration in the form of a sequence of images

$$\mathbf{V = \{I_t | t = 1..T\}}$$

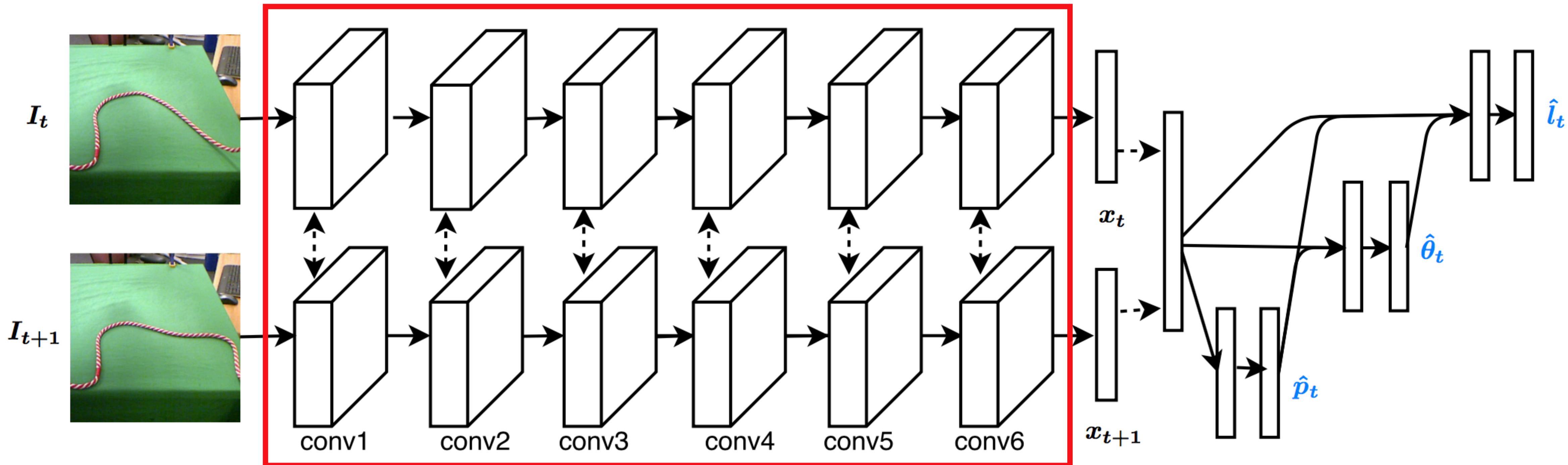
- The robot executes a series of actions to transform I_1 into I_2, then I_2 into I_3, and so on until the end of the sequence

Neural Network Architecture



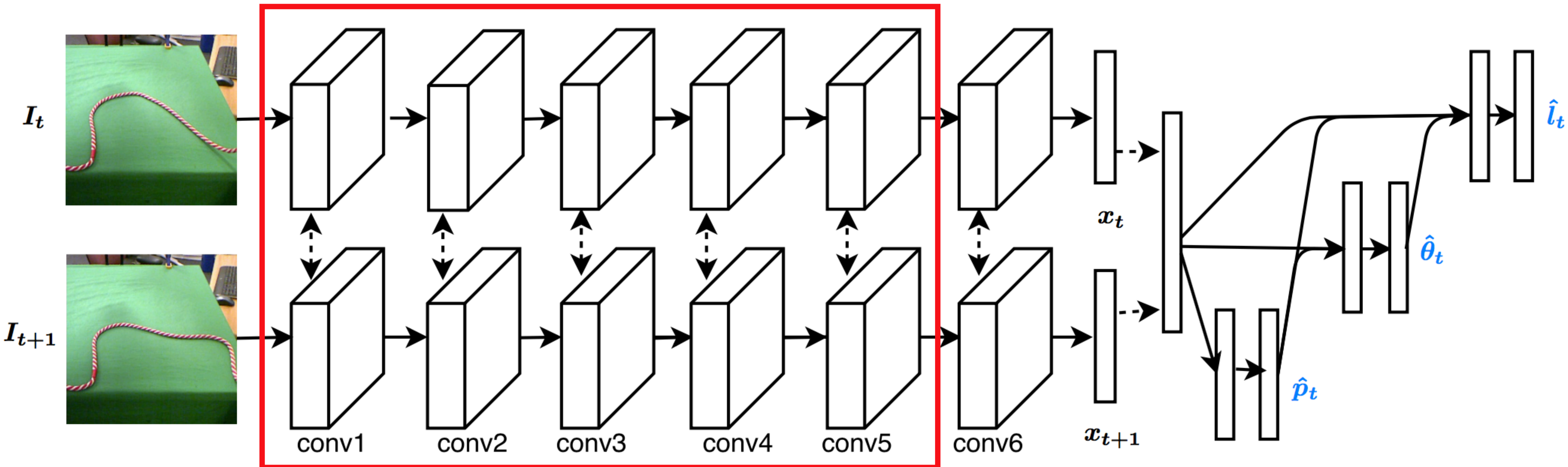
- **Pair of Input Images I_t and I_{t+1}.**
- **Network predicts the action that turn I_t into I_{t+1}.**

Neural Network Architecture

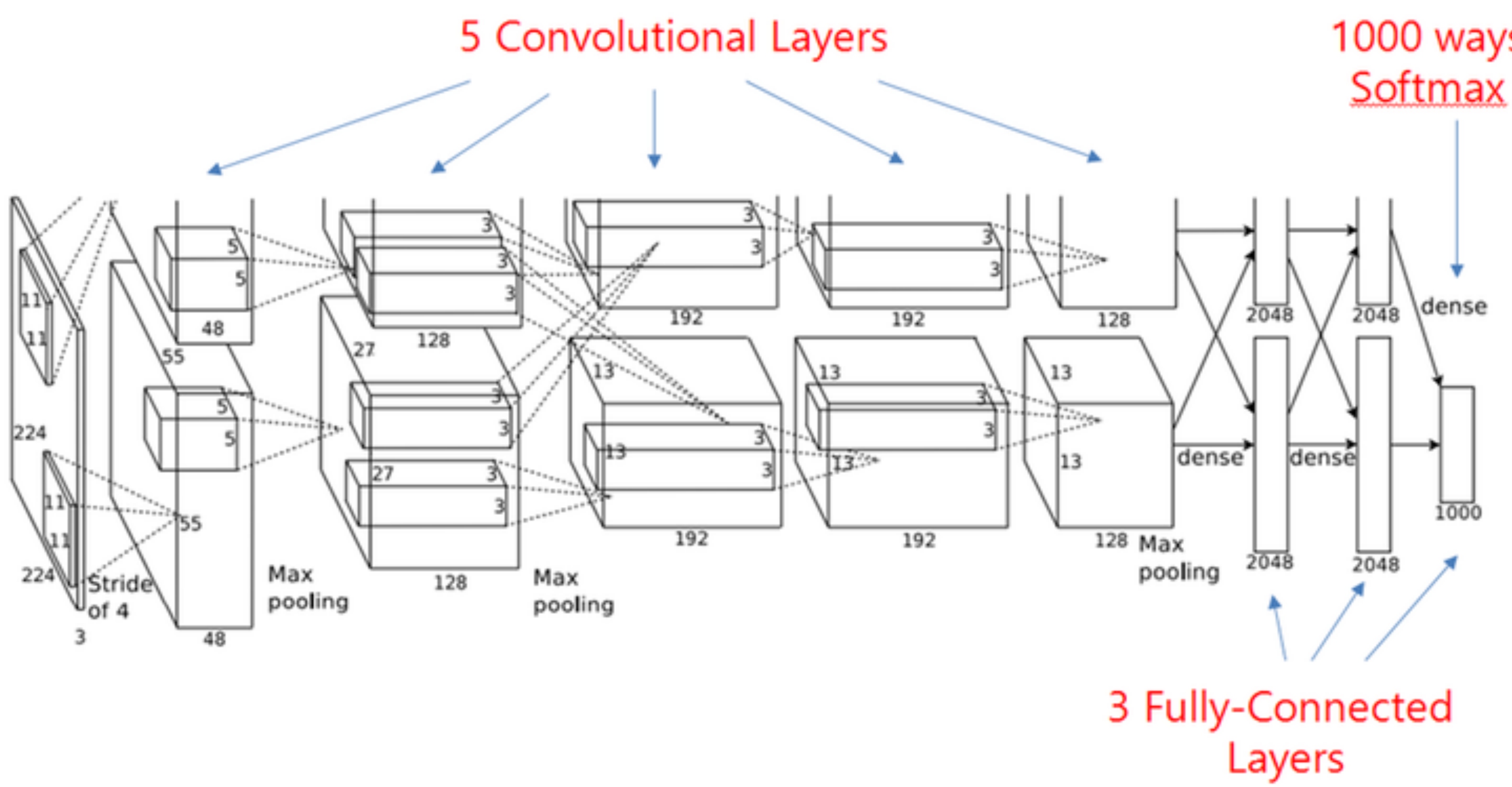


- **Two input streams.**
- **Network weights of both input streams are tied.**
- **ELU non-linearity**

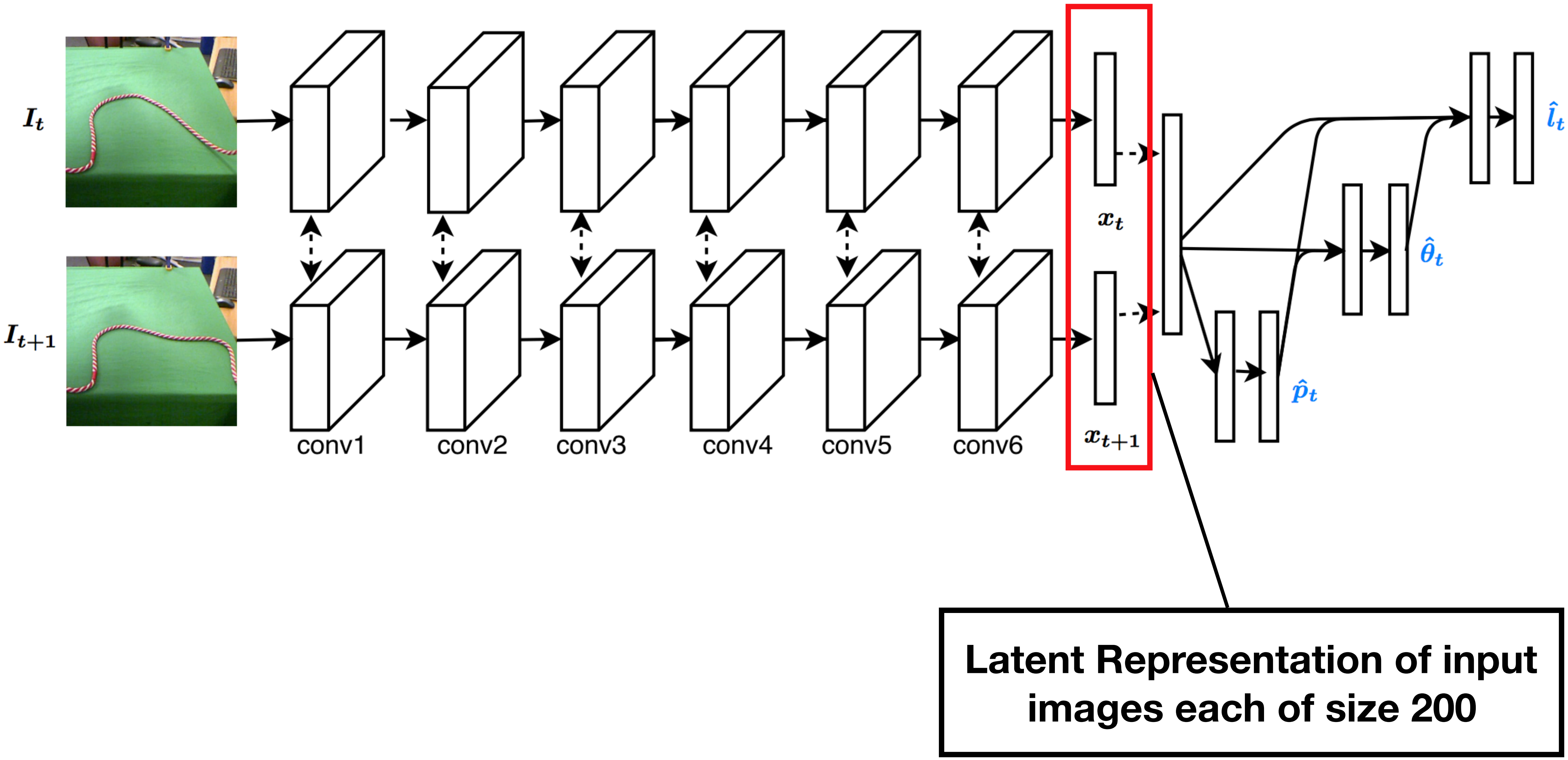
Neural Network Architecture



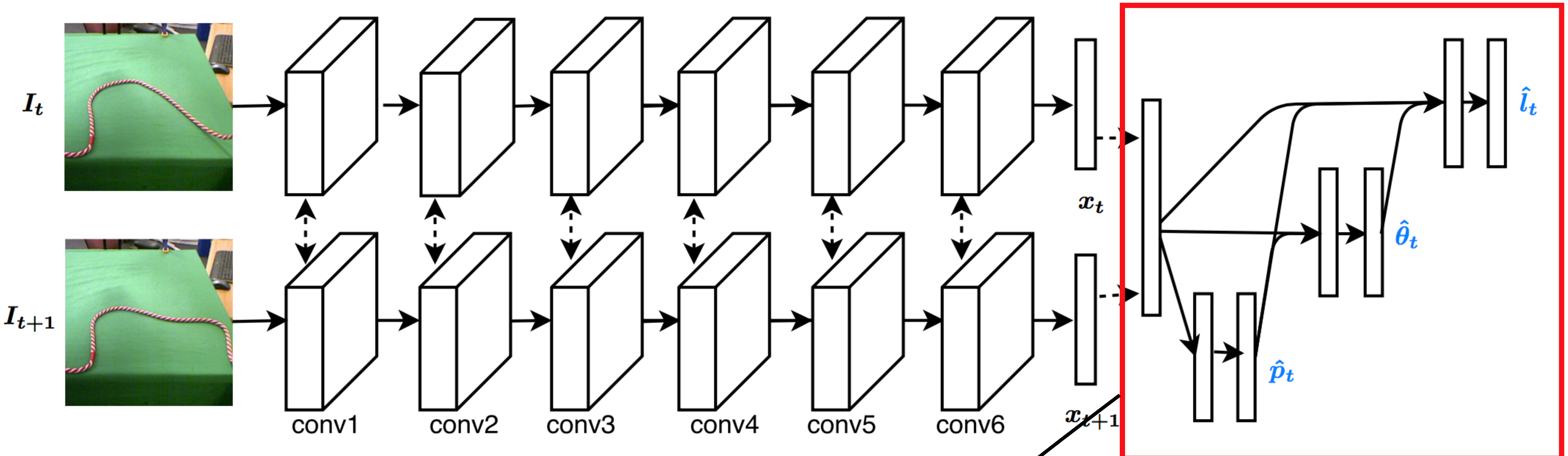
Same architecture as AlexNet



Neural Network Architecture

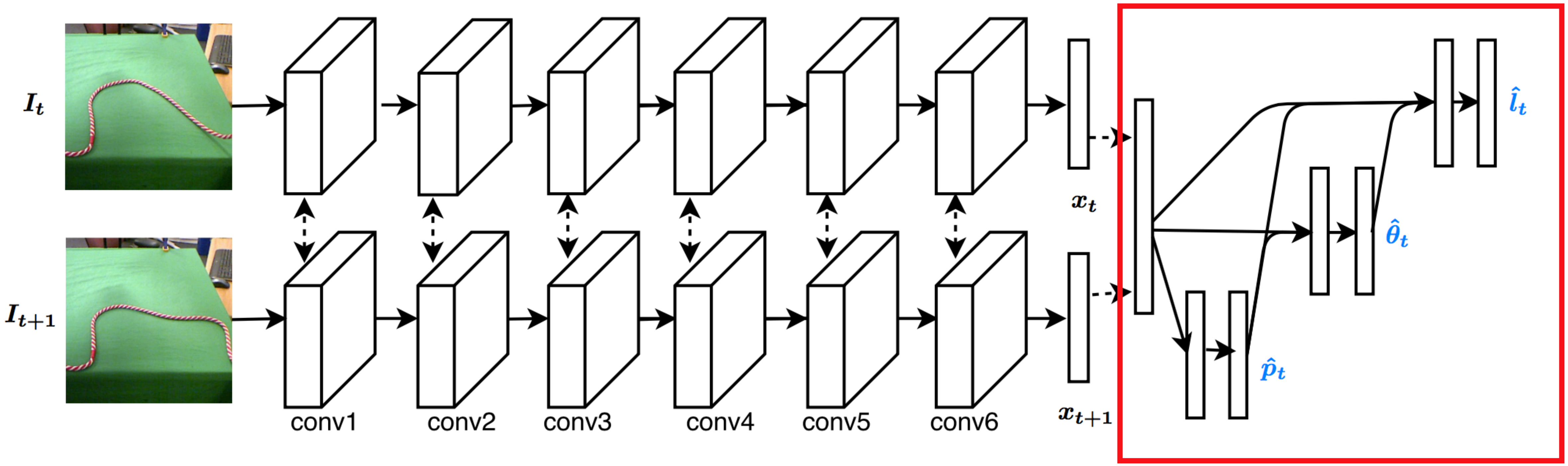


Neural Network Architecture



- **Provides joint non-linear feature representation of the I_t and I_{t+1}**
- **Representation is used to predict the action.**

Neural Network Architecture



- For training, turn action prediction into a classification problem.

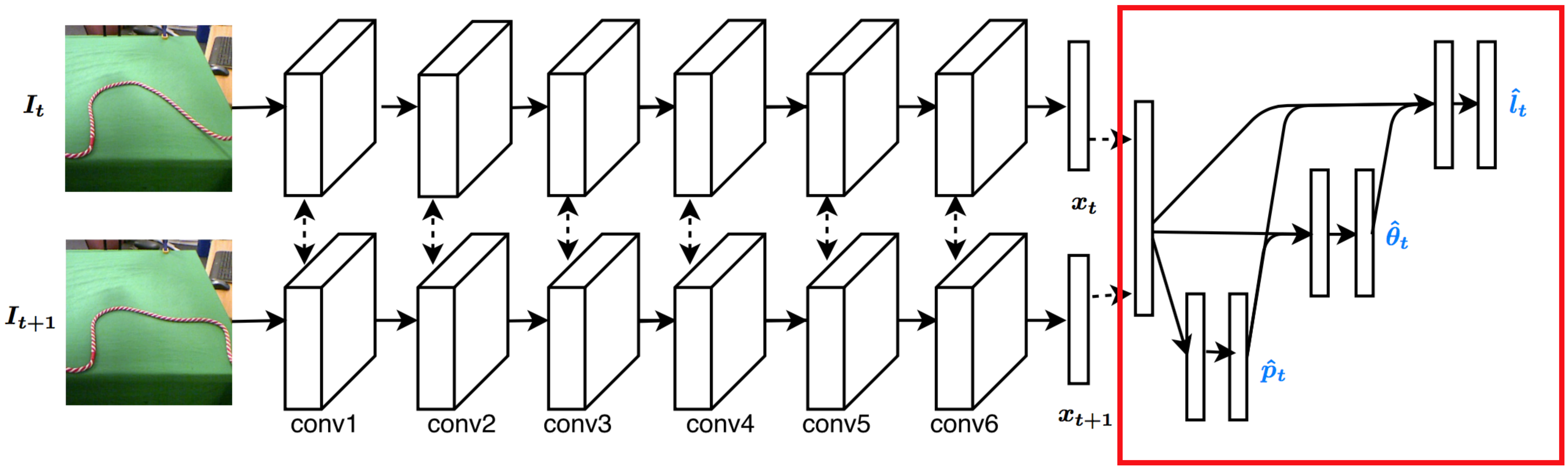
- Discretize Action space (p_t, θ_t, l_t)

  p_t (action location) -> Discretised onto a 20*20 spatial grid

  θ_t (direction) -> Discretised over 36 bins

  l_t (length) -> Discretised over 10 bins

Neural Network Architecture



- Classify action elements independently

- Decompose joint discrete Distribution of action

$$P\,(p_t, \theta_t, l_t) = P\,(p_t)P\,(\theta_t|p_t)P\,(l_t|\theta_t, p_t)$$

$$\boxed{P\,(p_t)} \xrightarrow{(x_t\ ,\ x_{t+1}),\ (\hat{p}_t)} \boxed{P\,(\theta_t|p_t)} \xrightarrow{(x_t\ ,\ x_{t+1}),\ (\hat{p}_t)\ ,\ (\hat{\theta}_t)} \boxed{P\,(l_t|\theta_t, p_t)}$$

Learning Through Self Exploration

1. Segment the rope using point cloud from Kinect camera.

2. Pick uniformly at random a pick point from this segment.

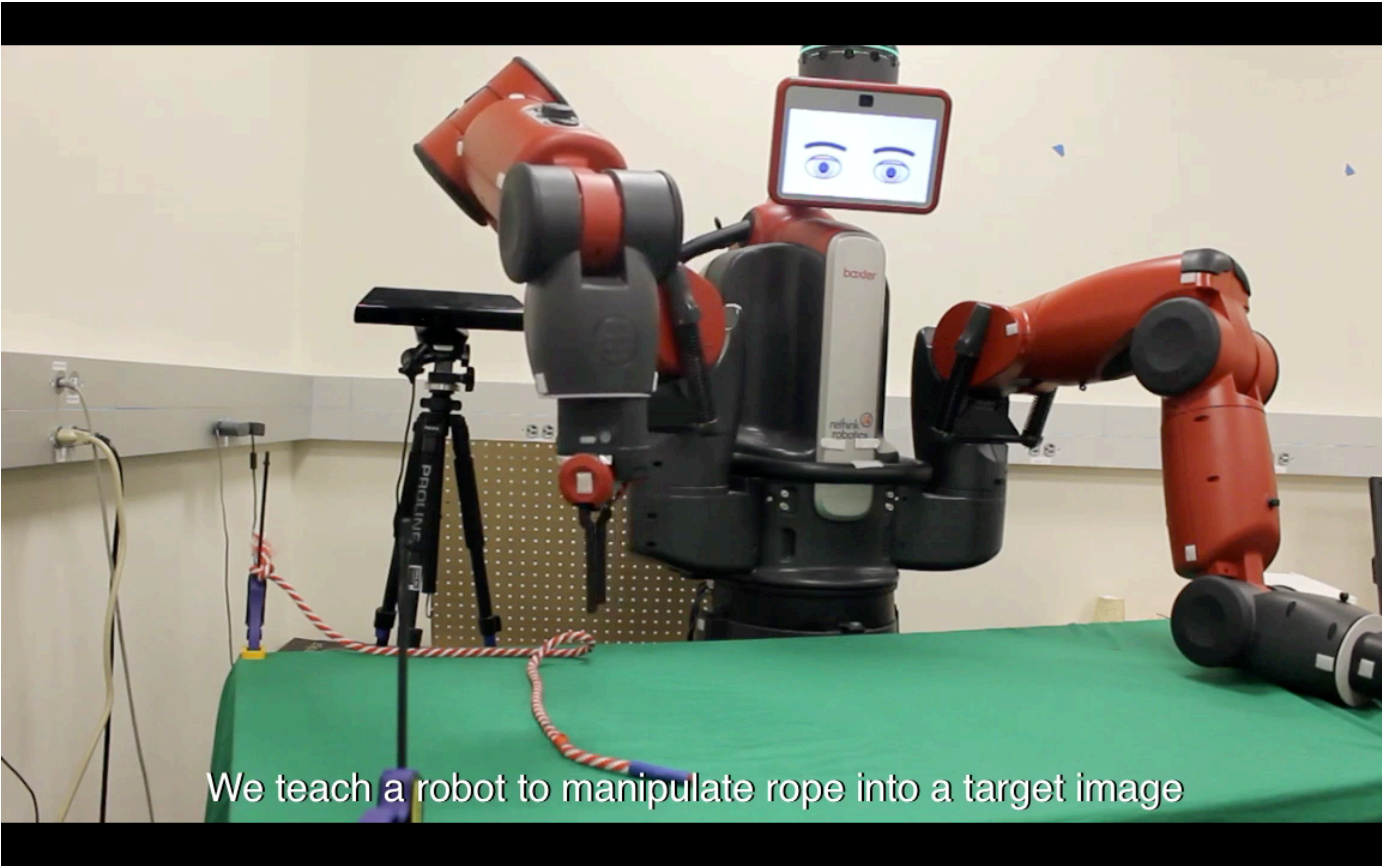
3. Drop point as displacement vector from pick point.
   Displacement Vector = $\boldsymbol{\theta} \in \mathbf{[0, 2\pi)} + \mathbf{l} \in \mathbf{[1, 15]}$
   θ and l are picked uniformly at random

Learning Through Self Exploration

4. Execute the Following Actions
   1. Grasp the rope at the pick point.
   2. Move the arm 5 cm vertically above the pick point.
   3. Move the arm to a point 5 cm above the drop point.
   4. Release the rope by opening the gripper.

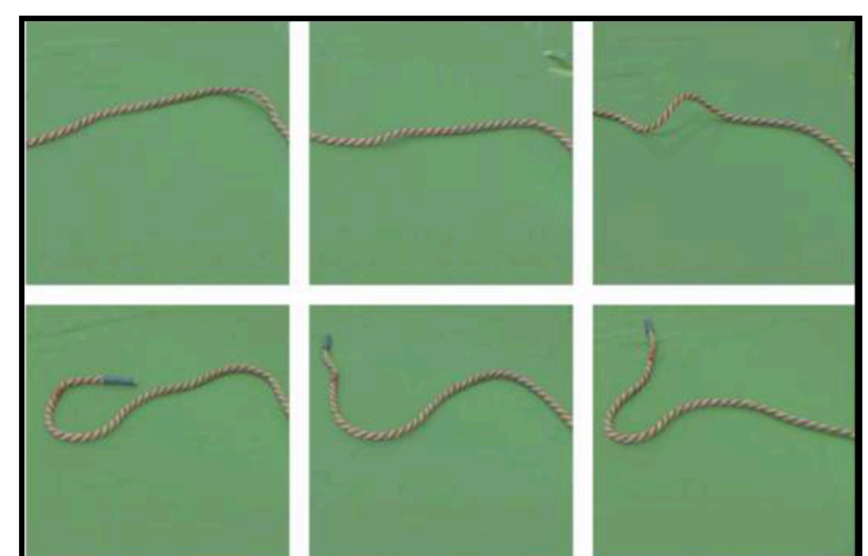
5. Resetting after 50 actions

Autonomous Data Collection to train NN



Active Data Collection

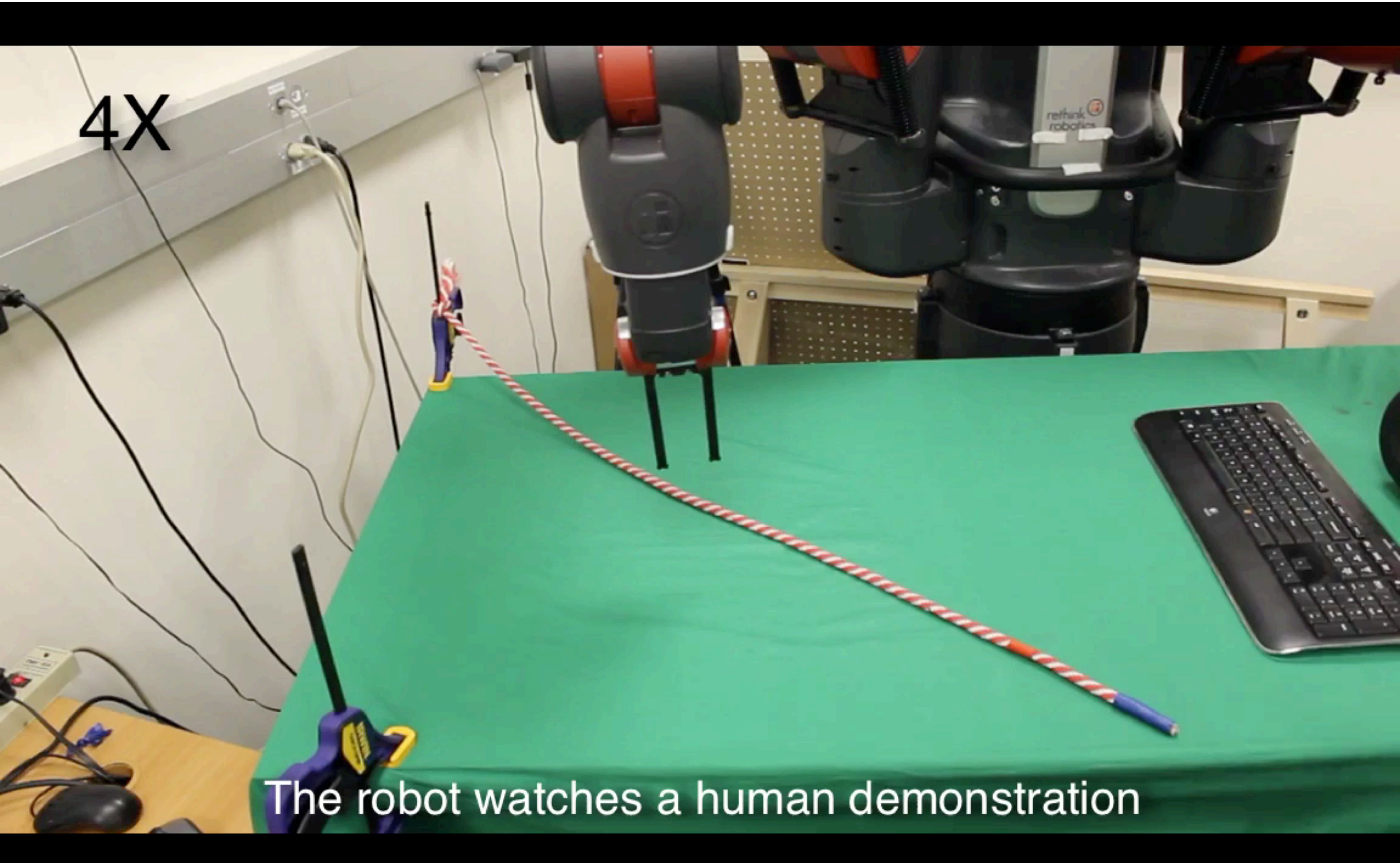
- Bias data collection towards interesting configurations.
- Manually arrange rope in Random configuration (i.e. the goal buffer).
- Instead of randomly sampling an action, randomly sample an image from goal buffer.
- Pass the current and goal image into the inverse model and use the action predicted for data collection.



Training Hyperparameters

- First five layers with pre-trained AlexNet weights.
- For first 5k iterations, Learning Rate = 0
- Rest of training, Learning Rate = 1e-4 && Adam Optimizer.
- All other layers initialised with small randomly distributed weights + Learning Rate= 1e-4.
- Validation set of 2.5 image pair for Hyper-parameter tuning.

Testing the System



Outline

- Introduction
- Robot Specification
- Neural Network Training
- **Evaluation**

Evaluation Procedure

- Task the robot to do different configurations.
- Calculate distance between human demonstrations and robot executions.
- **<u>Distance calculation</u>**:
  1. Segment the rope in each image
  2. Align points with (TPS-RPM)
  3. Calculate the mean pixel distance between the matched points.

Hand-Engineered Baseline

1. Pick the rope at the point with the largest deformation in the first image w.r.t second image.
2. Drop the rope at the corresponding point in the second image.

Nearest Neighbour Baseline

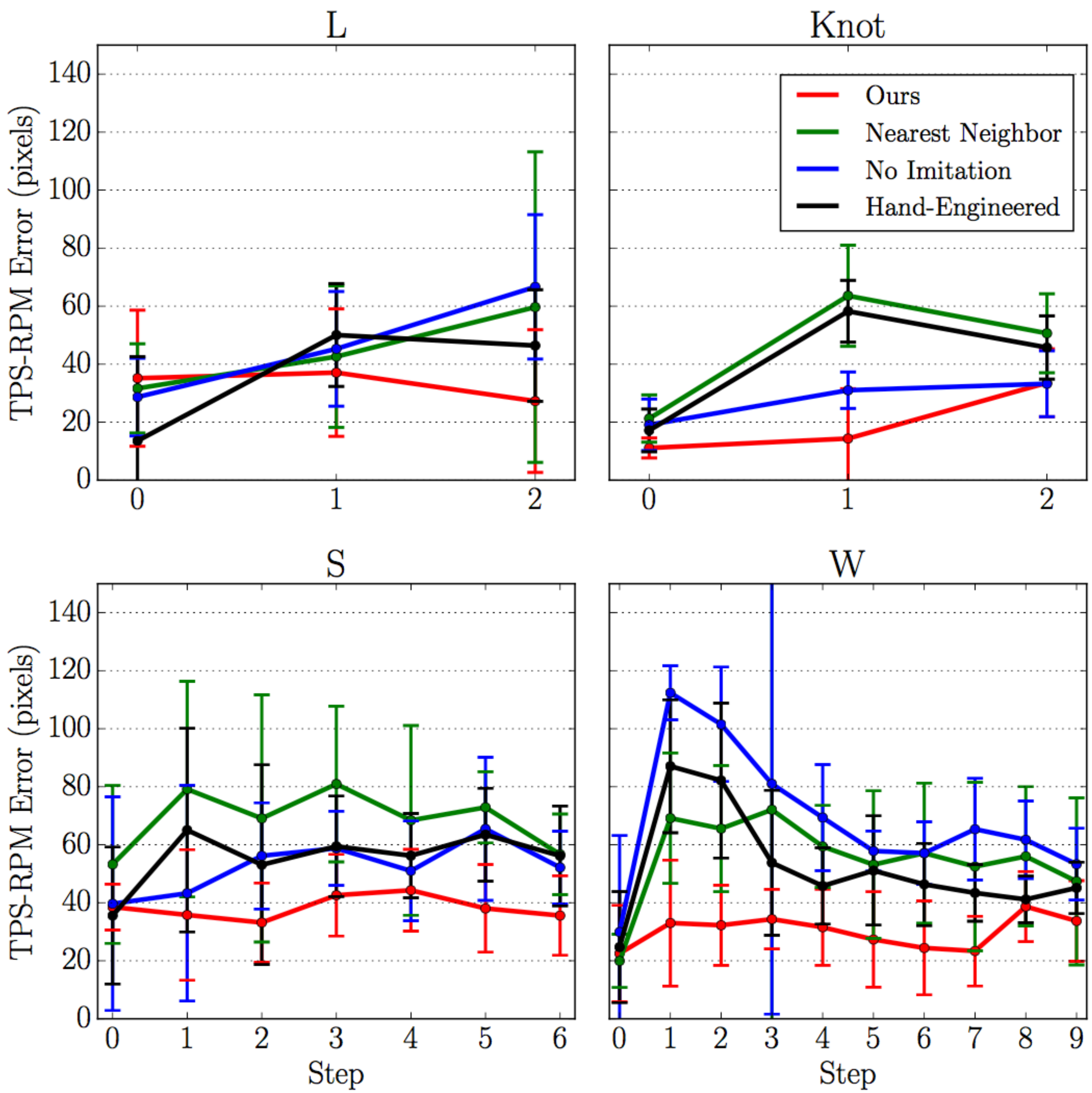
Given high level current and target Images **(I_t,$I_{t'+1}$):**

1. Pick pair of images **(I_k,I_{k+1})** in the training set that is closest to **(I_t,$I_{t'+1}$).**
2. Execute ground truth action used to collect this training sample.
3. Euclidean distance for nearest neighbour calculation.

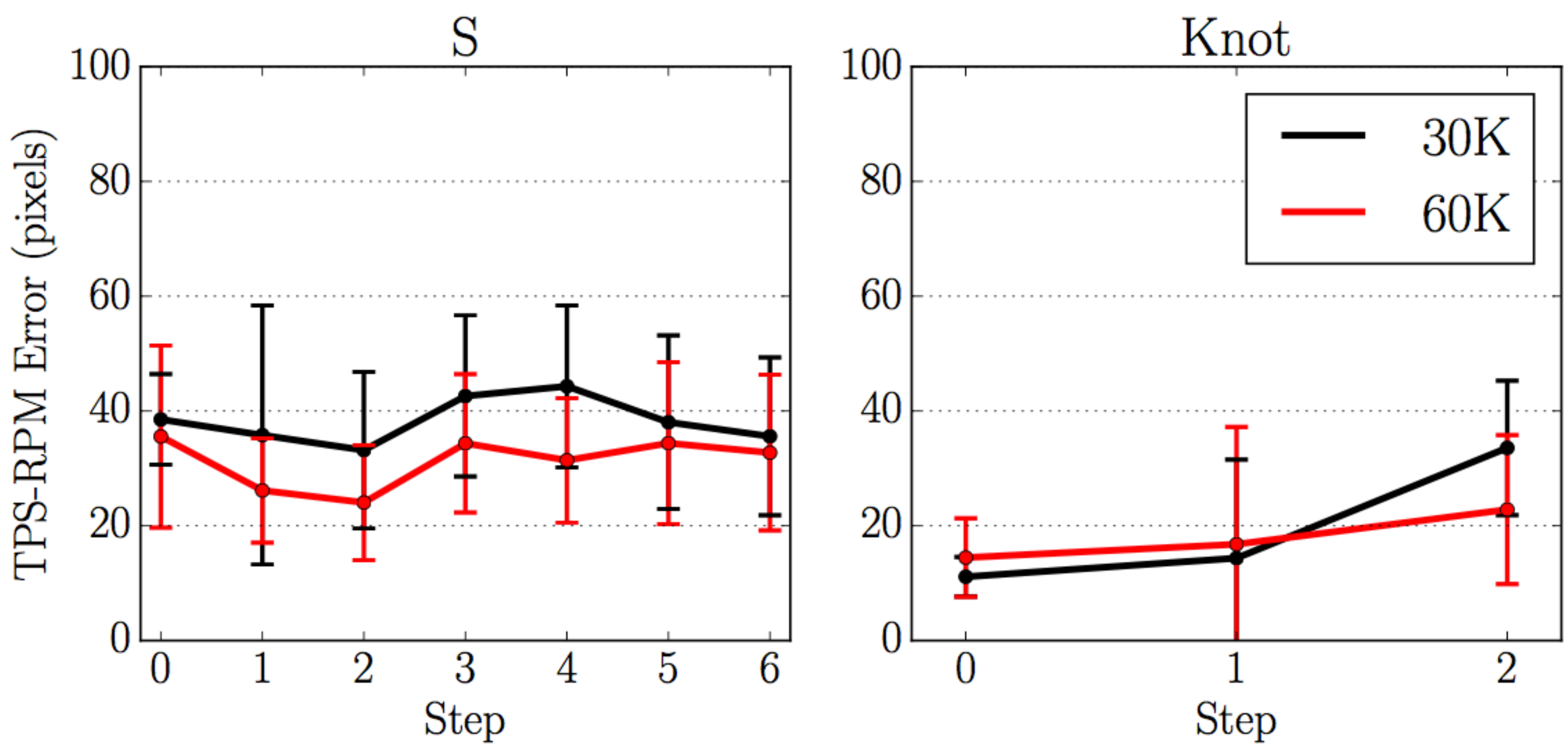
No Imitation Baseline

1. Only the initial and goal image **(I_1,I_T')** is fed to NN.
2. Predicted action is executed by robot and image **I_2** is obtained.
3. The pair **(I_2, I_T')** is fed to Inverse Model.
4. This process is repeated iteratively until we get to the final configuration.

Evaluation Results



Further Results



Knot Tying Success Rate

Imitation?	30K	60K
Yes	6/50	19/50
No	5/50	11/50

Method Generalization to other Ropes

- Successful at manipulating simpler shapes.
- Unsuccessful at manipulating complex configuration.
- Changed background also resulted in failure.

Summary

- Deformable object manipulation is a complicated task.
- Neural Nets can be used to teach robot to perform small deformations.
- Complex configurations could be learned via human imitation.
- Lot of work needs to be done in this area.

Thank You