Learning 2D Linear Dynamics in Image Space Using Neural Networks

Jeffrey Mahler, Michael Laskey, John Schulman, Sergey Levine, Jeff Donahue, Sachin Patil, Pieter Abbeel, and Ken Goldberg

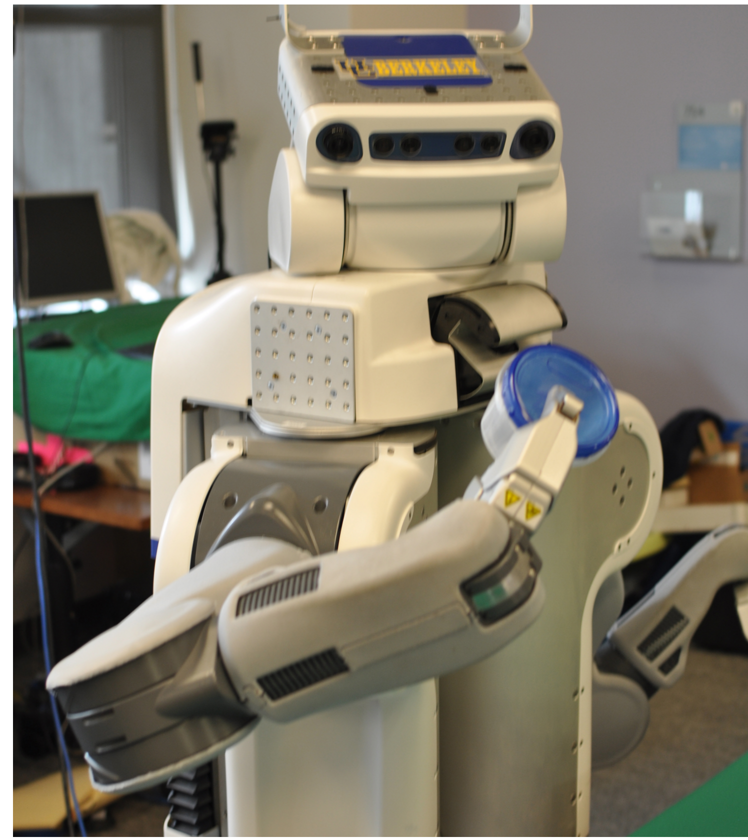
UC Berkeley EECS

RSS Workshop on Information-Based Grasp and Manipulation Planning

July 13th, 2014

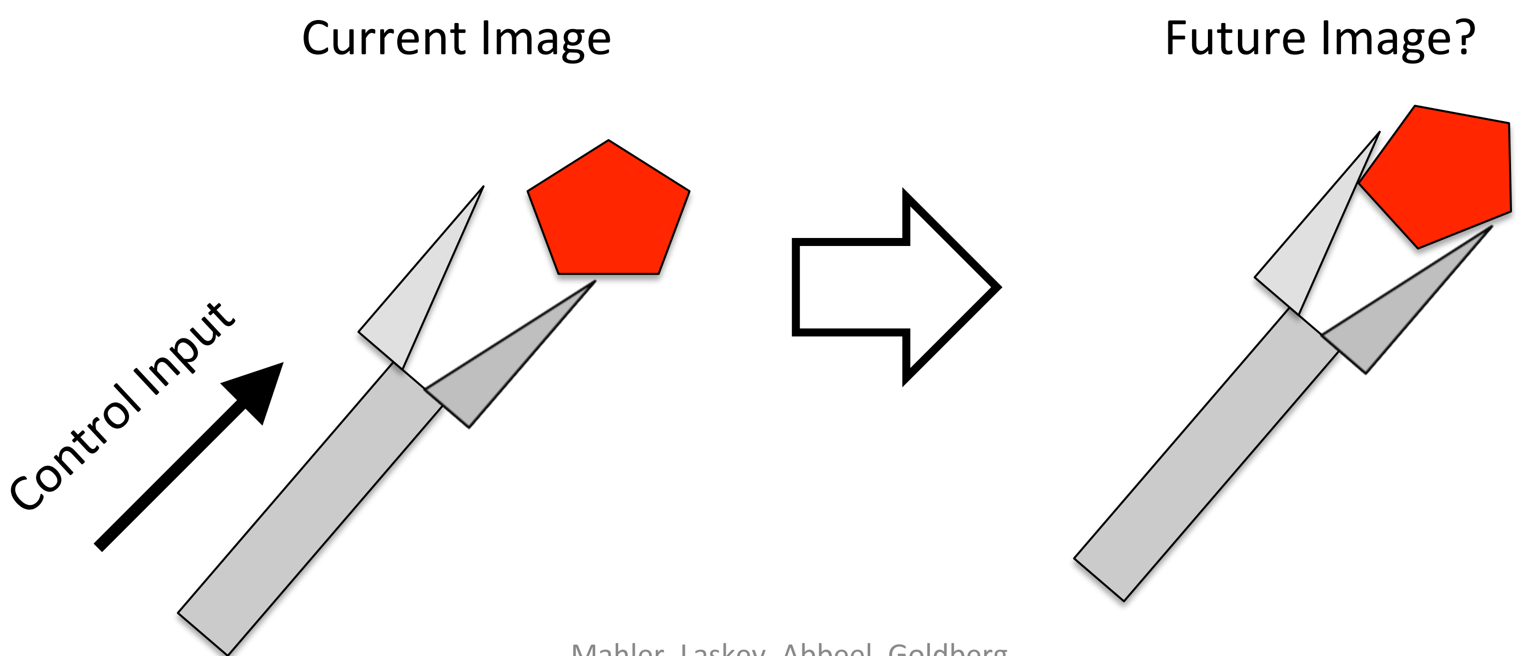
Motivation

- Planning for grasping and manipulation tasks with unknown objects
- Only sensor data and applied controls available



Objective

- Predict future observations from current observations and control inputs



Approach

- Learn the parameters of a dynamical system:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{v}_t \qquad \mathbf{v}_t \sim N(0, \sigma_1^2 I)$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{w}_t \qquad \mathbf{w}_t \sim N(0, \sigma_2^2 I)$$

  – Sensory data (e.g., images) $\mathbf{y}_t$

Approach

- Learn the parameters of a dynamical system:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{v}_t \qquad \mathbf{v}_t \sim N(0, \sigma_1^2 I)$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{w}_t \qquad \mathbf{w}_t \sim N(0, \sigma_2^2 I)$$

  - Sensory data (e.g., images) $\mathbf{y}_t$
  - Learned feature representations $\mathbf{x}_t$

Approach

- Learn the parameters of a dynamical system:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{v}_t \qquad \mathbf{v}_t \sim N(0, \sigma_1^2 I)$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{w}_t \qquad \mathbf{w}_t \sim N(0, \sigma_2^2 I)$$

  – Sensory data (e.g., images) $\mathbf{y}_t$
  – Learned feature representations $\mathbf{x}_t$
  – Control inputs (e.g., torques) $\mathbf{u}_t$

Approach

- Learn the parameters of a dynamical system:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{v}_t \qquad \mathbf{v}_t \sim N(0, \sigma_1^2 I)$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{w}_t \qquad \mathbf{w}_t \sim N(0, \sigma_2^2 I)$$

  – Sensory data (e.g., images) $\mathbf{y}_t$
  – Learned feature representations $\mathbf{x}_t$
  – Control inputs (e.g., torques) $\mathbf{u}_t$
  – Linear dynamics A, B

Approach

- Learn the parameters of a dynamical system:

$$\mathbf{x}_{t+1} = A\mathbf{x}_t + B\mathbf{u}_t + \mathbf{v}_t \qquad \mathbf{v}_t \sim N(0, \sigma_1^2 I)$$

$$\mathbf{y}_t = h(\mathbf{x}_t) + \mathbf{w}_t \qquad \mathbf{w}_t \sim N(0, \sigma_2^2 I)$$

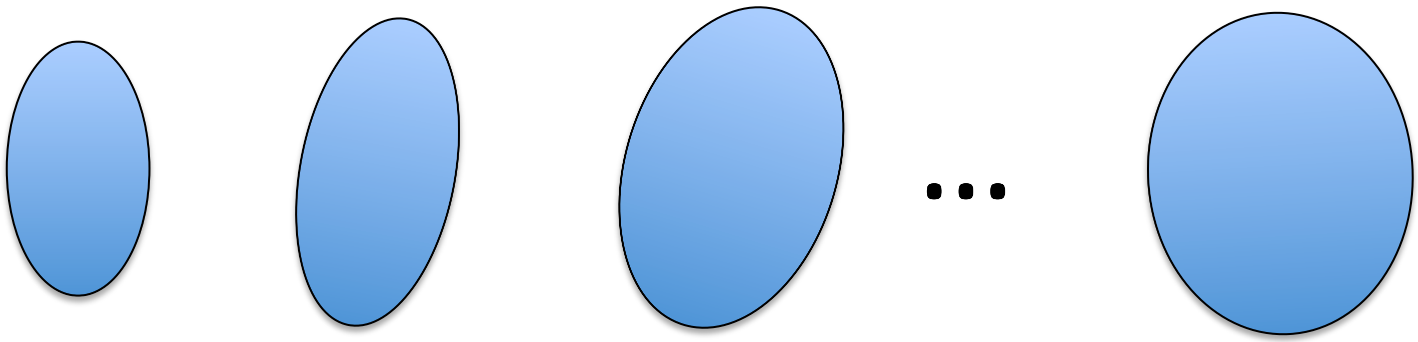
  - Sensory data (e.g., images) $\mathbf{y}_t$
  - Learned feature representations $\mathbf{x}_t$
  - Control inputs (e.g., torques) $\mathbf{u}_t$
  - Linear dynamics A, B
  - Neural network-generated image $h(\mathbf{x}_t)$

Algorithm

- Expectation maximization for (approximate) maximum-likelihood parameters

$P(\mathbf{x}_0|\mathbf{y}_{0:T})$ $P(\mathbf{x}_1|\mathbf{y}_{0:T},\mathbf{x}_0)$ $P(\mathbf{x}_2|\mathbf{y}_{0:T},\mathbf{x}_1)$ $P(\mathbf{x}_T|\mathbf{y}_{0:T},\mathbf{x}_{t-1})$

E-Step

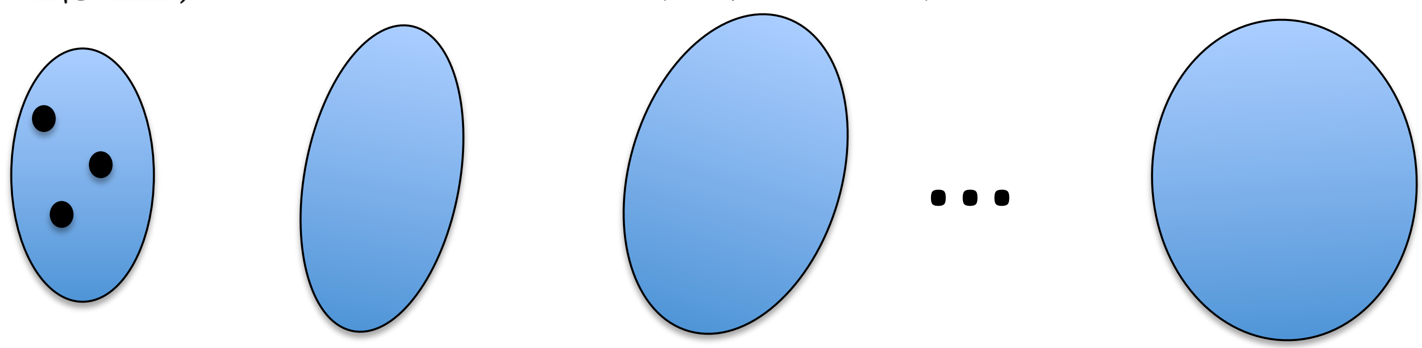


Algorithm

- Expectation maximization for (approximate) maximum-likelihood parameters

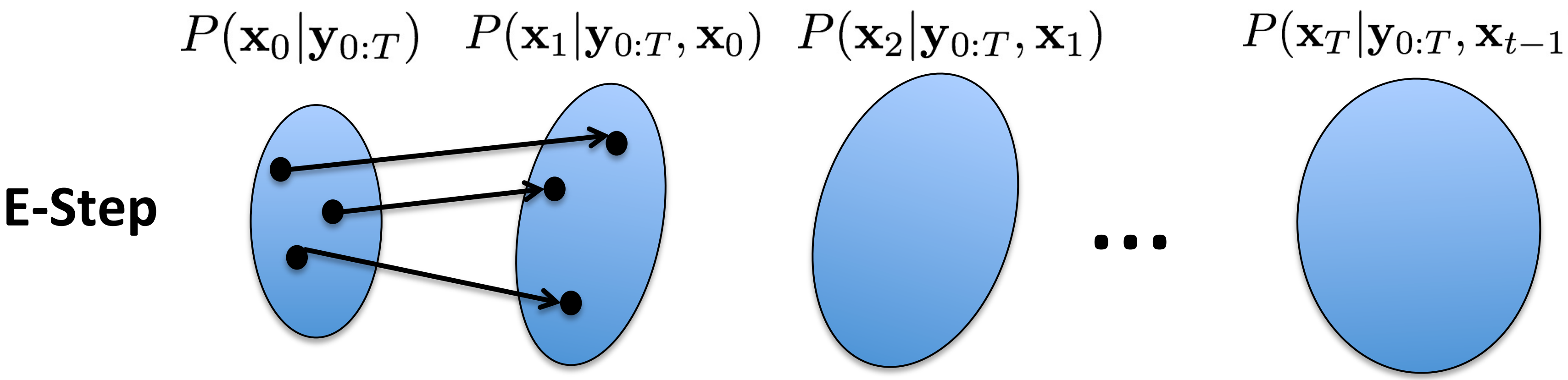
$P(\mathbf{x}_0|\mathbf{y}_{0:T})$ $P(\mathbf{x}_1|\mathbf{y}_{0:T},\mathbf{x}_0)$ $P(\mathbf{x}_2|\mathbf{y}_{0:T},\mathbf{x}_1)$ $P(\mathbf{x}_T|\mathbf{y}_{0:T},\mathbf{x}_{t-1})$

E-Step



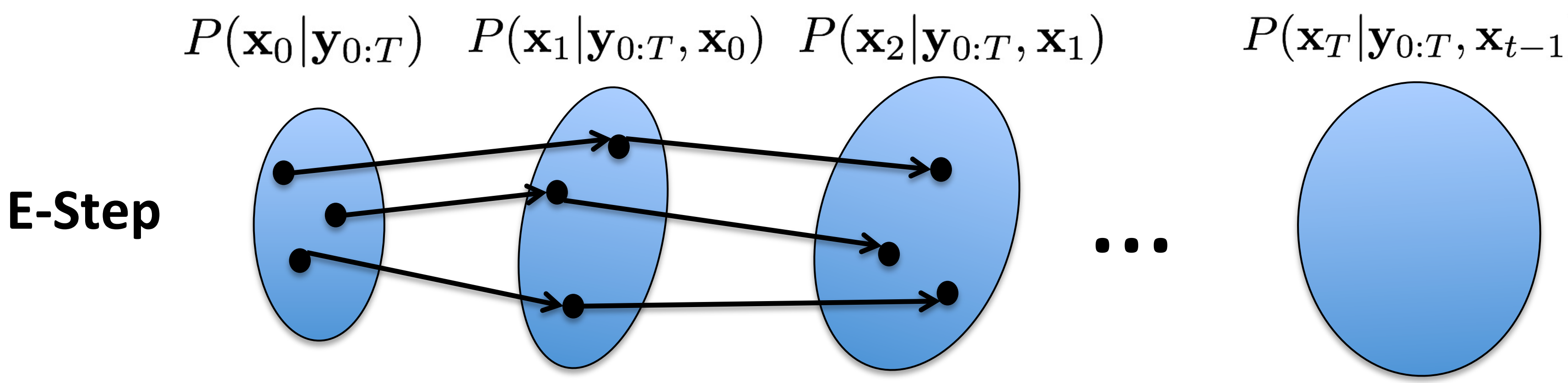
Algorithm

- Expectation maximization for (approximate) maximum-likelihood parameters



Algorithm

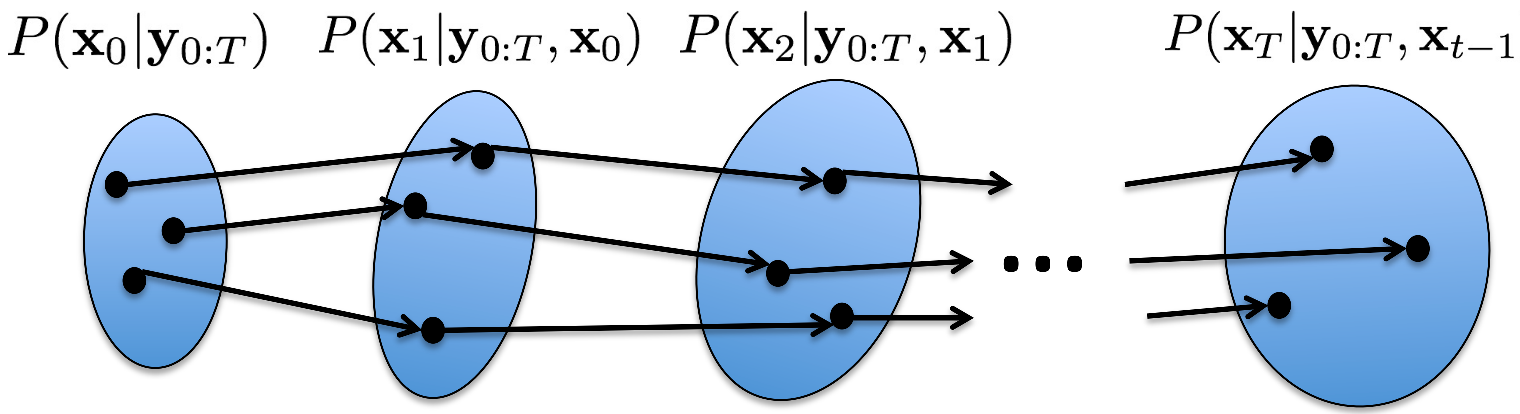
- Expectation maximization for (approximate) maximum-likelihood parameters



Algorithm

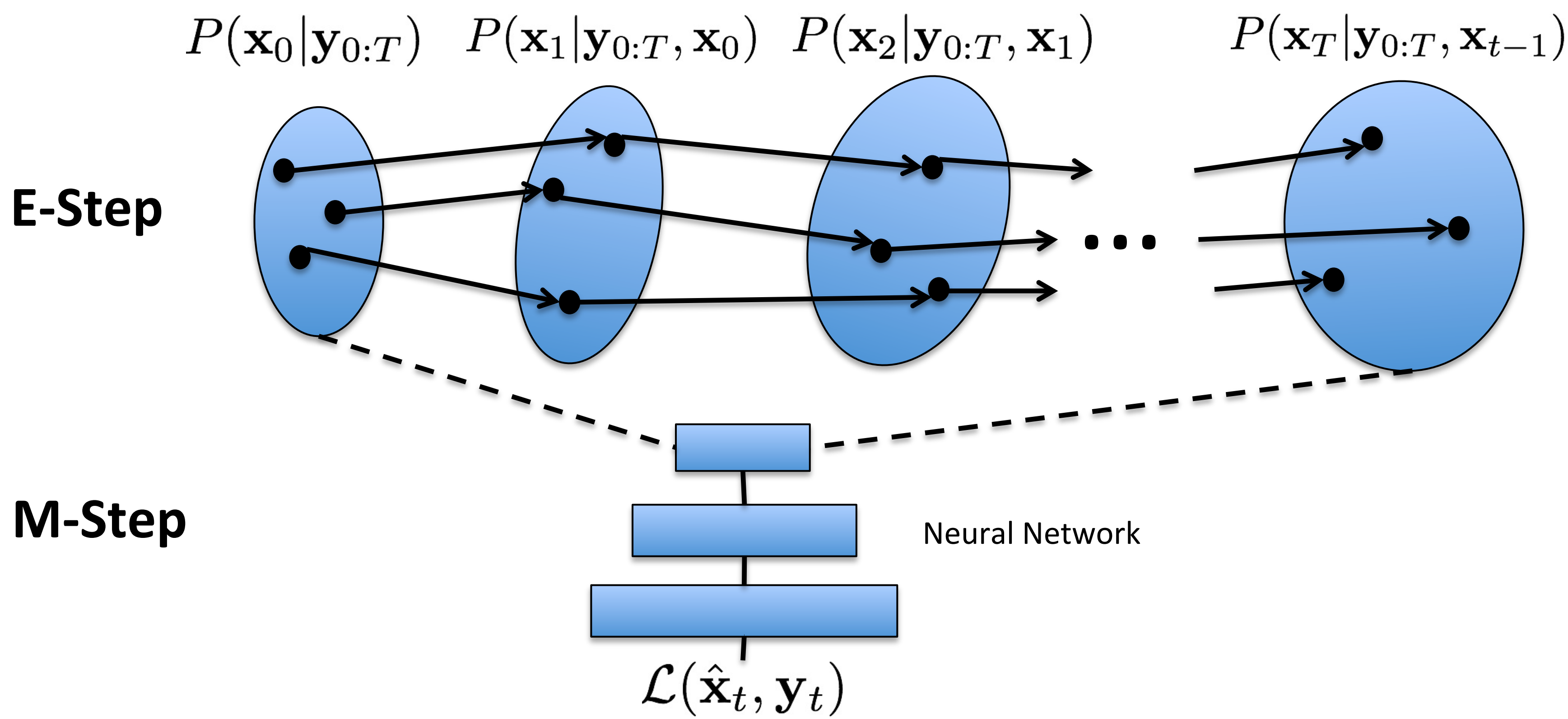
- Expectation maximization for (approximate) maximum-likelihood parameters

E-Step



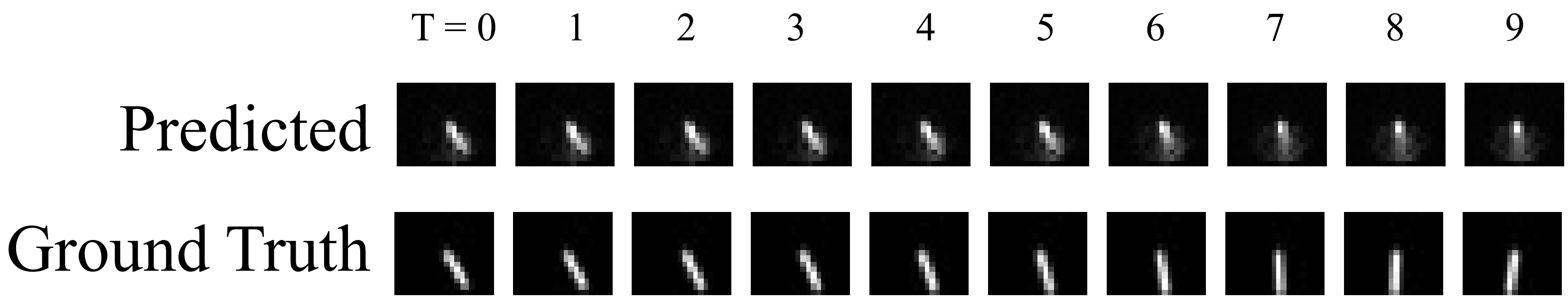
Algorithm

- Expectation maximization for (approximate) maximum-likelihood parameters



Discussion

- Initial results



- Future Work
  - Natural images and depth images
  - Planning a manipulation task
  - Alternative dynamics functions