Robot Skill Adaptation
via Soft Actor-Critic Gaussian Mixture Models

Sassan Mokhtar, Rahul Hegde, Martin Condor

Motivation

- Objective: Enabling robots to learn low-level manipulation skills in a scalable and structured manner
- Two different approaches:
  - Model-based (dynamical system): + data-efficient, + trajectory reasoning, - incompetence in handling high-dimensional data, - susceptible to noise
  - Data-driven (reinforcement learning): + high expressiveness, + robust to noise, - data hungry
- Idea: Hybrid approach: Learn a dynamical system by GMM from few demonstration, then refine the GMM by SAC

Gaussian Mixture Models

- GMM: Parametric density estimation method
- GMR: Regression approach to model probability distributions

Problem Formulation:

- Dynamical systems are aimed to represent the motion progression of robot
- Assume the robot skill model is an autonomous dynamical system

$$x' = f_\theta(x) + \epsilon$$

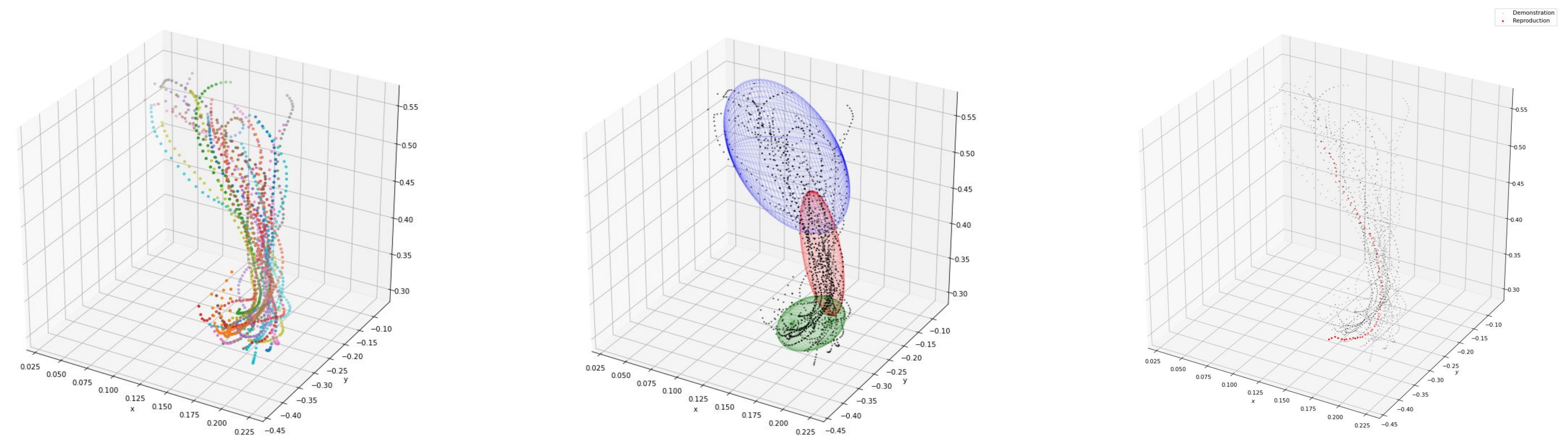
- **Training**: Optimize the GMM by EM to represent the joint distribution of the pose and velocity of the robot

$$p(x, x')$$

- **Prediction**: Generate a new robot pose by sampling the conditional distribution

$$x'_t \sim \underbrace{p(x'|x = x_t)}_{\approx f_\theta}$$

- Compute the next position: $x_{t+1} = x_t + \Delta t . x'_t$



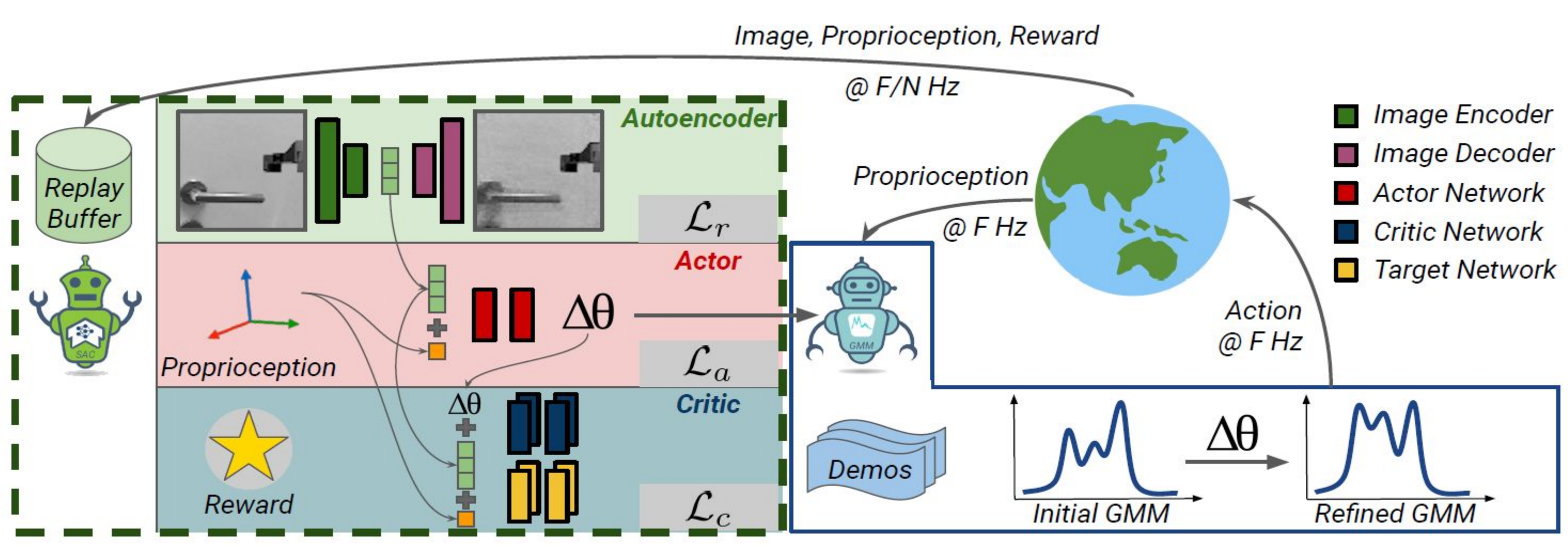
Trajectories GMM GMR

Soft Actor-Critic method

- Once GMM learned the trajectory space of the skill, the refinement can be performed by robot's interaction with the world
- The refinement of the GMM can be formulated as a reinforcement learning problem
- The SAC is used to learn the skill refinement policy
  - Action space: GMM parameters
  - State space: robot pose, proprioceptive measurements, latent representation of the sensory images

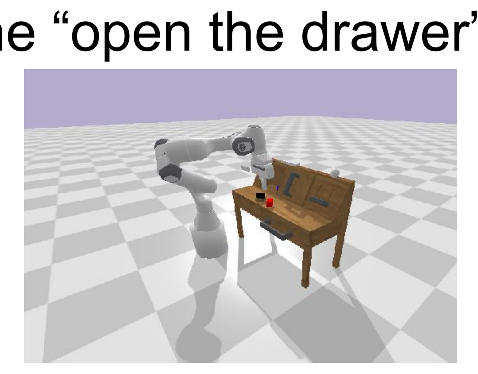
Autoencoders

- Including images in the state space lead to massive computational cost
- Idea: Use an autoencoder and append the latent representation of the images in the state space



Experimental Results

- We collect 150 different trajectories for the "open the drawer" task from the Calvin dataset
- GMM fit on 30 close trajectories and test on 90 random trajectories



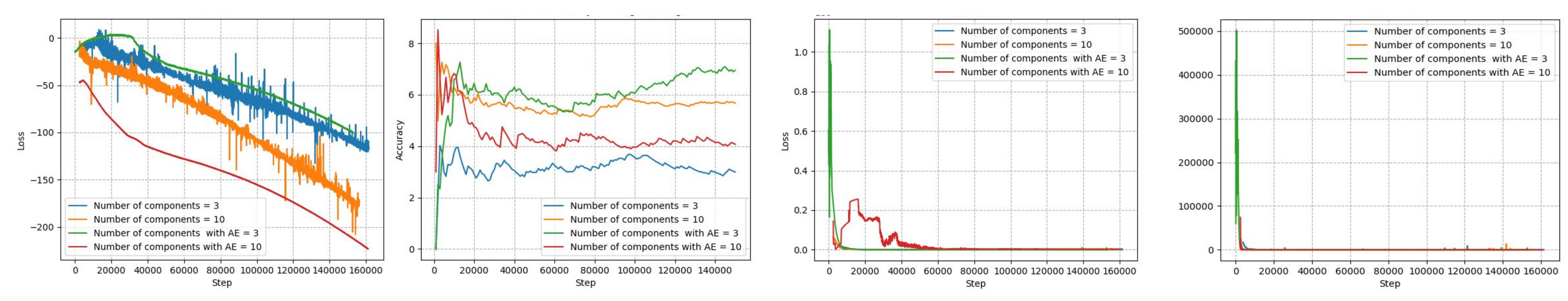
Calvin Environment

GMM:

	Training Acc	Test Acc
3 components	36.6%	**30%**
10 components	36.6%	**30%**

GMM + SAC:

After N interactions of the agent with the environment, the SAC updates the GMM model accordingly



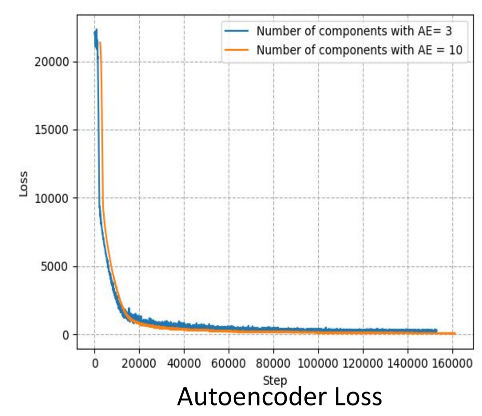
Actor Loss Validation Accuracy Value Loss Critic Loss

	Training Acc	Test Acc
3 components	53.3%	**56.6%**
10 components	96.6%	**90%**

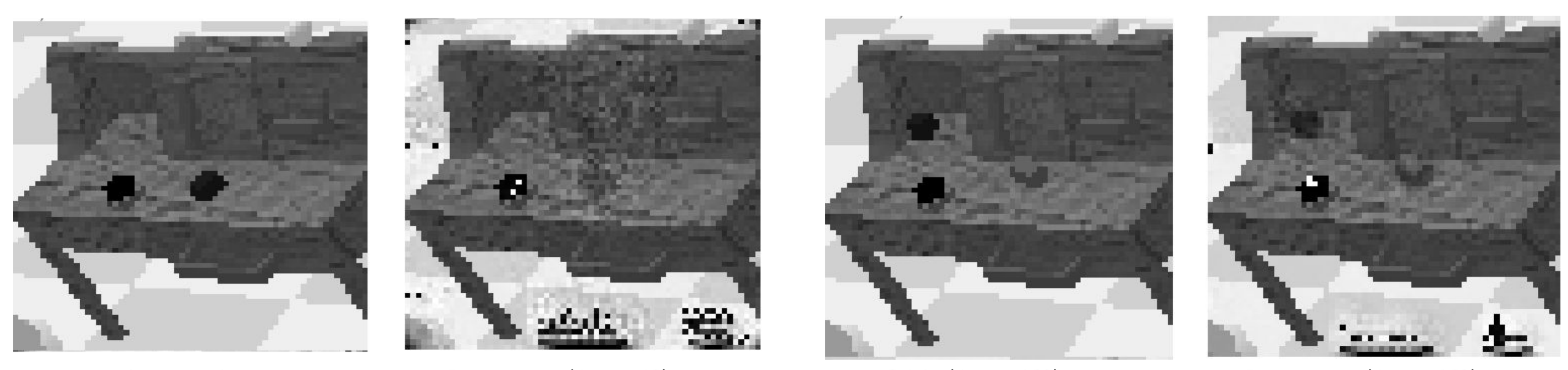
GMM + SAC + AE:

- Autoencoder trains concurrently with SAC
- Utilizing autoencoder results in more accurate and faster convergence

(34,000 steps, compare to 160,000 steps for 10 components)



Autoencoder Loss



Original Image 40k steps Reconstructed Image 40k steps Original Image 160k steps Reconstructed Image 160k steps

	Training Acc	Test Acc
3 components	76.6%	**75.5%**
10 components	96.6%	**97.7%**