Auto-conditioned Recurrent Mixture Density Networks for Learning Generalizable Robotic Manipulation Skills

Hejia Zhang, Eric Heiden, Stefanos Nikolaidis, Joseph J. Lim, Gaurav S. Sukhatme

***Abstract*— Personal robots assisting humans must perform complex manipulation tasks that are typically difficult to specify in traditional motion planning pipelines, where multiple objectives must be met and the high-level context be taken into consideration. In this paper, we introduce a state transition model (STM) that generates joint-space trajectories by imitating motions from expert behavior. Given a few demonstrations, we show in real robot experiments that the learned STM can quickly generalize to unseen tasks and synthesize motions having longer time horizons than the expert trajectories. Compared to conventional motion planners, our approach enables the robot to accomplish complex behaviors from high-level instructions without laborious hand-engineering of planning objectives, while being able to adapt to changing goals during the skill execution. In conjunction with a trajectory optimizer, our STM can construct a high-quality skeleton of a trajectory that can be further improved in smoothness and precision.**

I. Introduction

While numerous *learning from demonstration* (LfD) algorithms [1] have been proposed for robot manipulation skills learning, teaching robots generalizable skills is still challenging.

In this paper, we propose a learned *state transition model* (STM) that can imitate a variety of motions. We show our proposed model has the generalizability to perform tasks with unseen goals and plan tasks with longer time horizons than the demonstrated tasks.

In this work, we present 1) a training procedure and stochastic recurrent neural network architecture that can efficiently learn robot skills from demonstrations in joint position space, 2) real-robot experiments that demonstrate the generalizability of our STM, 3) trajectory optimization results attained from the STM-generated trajectory as skeleton.

We refer the reader to the full paper version of this work [2] for more details.

II. Problem Formulation

In this paper, we study the problem of learning robot skills directly from a set of expert trajectories which are represented by sequences of states.

Given n expert trajectories $\{\xi_i^*\}_{i=0}^n$, where each trajectory ξ_i^* is a state sequence $\{\mathbf{s}_{t_i}^*\}_{t_i=0}^{T_i}$ of length T_i, the problem is to estimate a model $p_\theta(\mathbf{s}_{t+1}|\ \mathbf{s}_t)$ that, when unrolled for T_i time steps from a start state $\mathbf{s}_0$, computes trajectories that resemble the expert demonstrations.

Hejia Zhang, Eric Heiden, Stefanos Nikolaidis, Joseph J. Lim, and Gaurav S. Sukhatme are with the Department of Computer Science, University of Southern California, Los Angeles, USA {hejiazha, heiden, nikolaid, limjj, gaurav}@usc.edu.

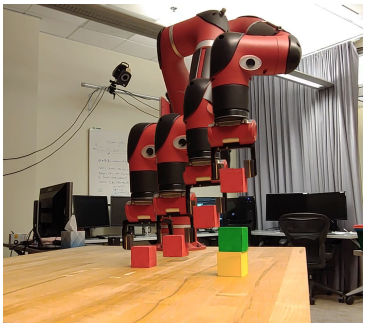

Fig. 2: Action sequence of the block-stacking task on the Sawyer robot using the proposed state transition model (STM) synthesizing trajectories in joint-position space.

Throughout this work, we define a *state* at a discrete time step t as a vector of real numbers

$$\mathbf{s}_t = (\Delta q_t^0, \Delta q_t^1, \ldots, \Delta q_t^6, \phi_t, \psi_t),$$

where $\Delta\mathbf{q} = \{\Delta q_t^j\}_{j=0}^6$ describes the changes in joint angles relative to the previous time step, ϕ_t and ψ_t are vectors that denote the *task-specific input* and the *task description*, respectively.

III. Methodology

The STM $p_\theta(\mathbf{s}_{t+1}|\ \mathbf{s}_t)$ is a machine learning model parameterized by vector θ that captures the probability distribution over state transitions between the current state $\mathbf{s}_t$ and the next state $\mathbf{s}_{t+1}$.

A. Mixture Density Network (MDN)

Capturing the stochasticity of the state transitions is an integral ingredient for the deployment of our model on a real robot as future states are uncertain and high-dimensional. To address our first requirement of representing uncertainty, we use a *mixture density network* (MDN) [3] to estimate the probability distribution of future states.

The MDN parameterizes a multivariate mixture of Gaussians by estimating the distribution over the next states as a linear combination of Gaussian kernels:

$$p(\mathbf{s}_{t+1}|\mathbf{s}_t) = \sum_{i=1}^{m} \alpha_i(\mathbf{s}_t) g_i(\mathbf{s}_{t+1}|\mathbf{s}_t),$$

where m is the number of Gaussians modelled by the MDN, α_i is the learned mixing coefficient and $g_i(\mathbf{s}_{t+1}|\mathbf{s}_t)$ is the i-th Gaussian kernel of the form

$$g(\mathbf{s}_{t+1}|\mathbf{s}_t) = \frac{1}{\sqrt{2\pi}\sigma_i(\mathbf{s}_t)} \exp\left\{-\frac{||\mathbf{s}_{t+1} - \mu_i(\mathbf{s}_t)||^2}{2\sigma_i(\mathbf{s}_t)^2}\right\}.$$

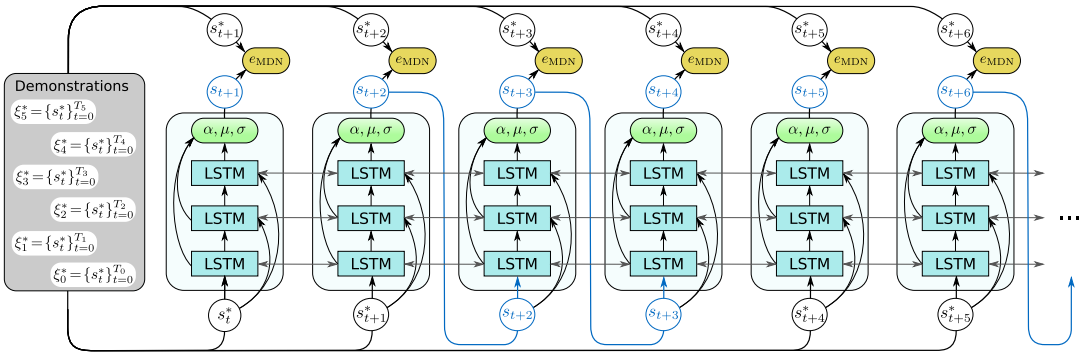

Fig. 1: Architecture of the proposed auto-conditioned recurrent mixture density network to model state transitions, unrolled over 6 time steps, with an exemplary auto-conditioning length $v = 2$ and ground truth length $u = 2$ (see Sec. III-C).

In addition to α_i, the kernel mean μ_i and standard deviation σ_i are learned by the MDN.

Given the ground-truth state pair $(\mathbf{s}_t^*, \mathbf{s}_{t+1}^*)$, we define the MDN loss as the negative log-likelihood:

$$e_{\text{MDN}} = -\ln\left\{\sum_{i=1}^{m} \alpha_i(\mathbf{s}_t^*) g_i(\mathbf{s}_{t+1}^* | \mathbf{s}_t^*)\right\}.$$

B. Long Short-Term Memory (LSTM)

To learn sequences of states, we require a model with an internal memory that allows it to remember states over long time horizons. As shown in Fig. 1, we propose to use the *long short-term memory* (LSTM) architecture that maintains a hidden state $\mathbf{h}_t$. This allows the STM to make predictions of states over long time horizons.

C. Auto-conditioning

We train the recurrent MDN with auto-conditioning [4], a learning schedule that, for every u iterations of a sequence of v time steps, feeds the LSTM's output as input into the cell computing the next state (Fig. 1). This enables the network to correct itself from states that deviate from demonstrations.

D. STM as Initial Solution for Trajectory Optimization

To combine data-driven methods with trajectory optimization methods, we sample from our model first to generate a feasible initial trajectory skeleton, $\{\tilde{\mathbf{q}}_t\}_{t=1}^T$. We want to retain the shape of the initial trajectory, while improving its smoothness by minimizing the objective similar to [5]:

$$V(\{\mathbf{q}_t\}_{t=1}^T) = \sum_{t=1}^{T-1} \|\mathbf{q}_t - \tilde{\mathbf{q}}_t\|_2^2 + \gamma \|\mathbf{q}_{t+1} - \mathbf{q}_t\|_2^2.$$

IV. Evaluating Generalizability

To evaluate the generalizability of our proposed method we investigate if our model can adapt online to changing goals. In the first experiment, as shown on the left in Fig. 3, we let the STM synthesize a trajectory that makes the gripper reach to the goal position (blue). Midway through the execution, we change the goal coordinates (red) and observe that our model is able to quickly adapt to this change, exceeding the length of all demonstration trajectories our model was trained on. In our second experiment (Fig. 3 right), Sawyer picks up the block from a preset location (drawn with black solid lines). After grasping, we change the goal location of the block (orange box) and the STM exhibit fast adaption to these new conditions. The adapted trajectory (green line) is close to the movement planned directly for the new goal location (blue line).

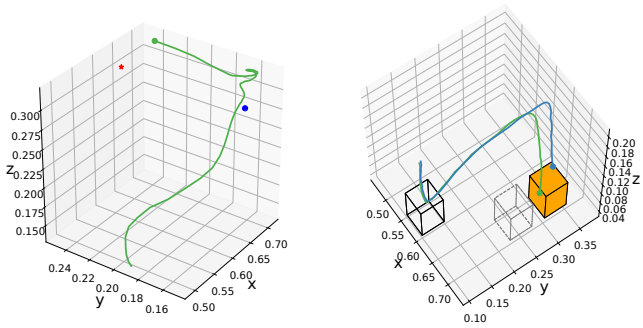

Fig. 3: Plot of the gripper position trajectory generated for the goal changing reacher (Left) and pick-and-place (Right) task.

V. Conclusion

In this work, we present a recurrent neural network architecture and training procedure that enables the efficient generation of complex joint position trajectories. Our experiments have shown that our STM can generalize to unseen tasks and plan tasks with longer time horizons than the demonstrated tasks.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] H. Zhang, E. Heiden, S. Nikolaidis, J. J. Lim, and G. S. Sukhatme, "Auto-conditioned recurrent mixture density networks for learning generalizable robot skills," *CoRR*, vol. abs/1810.00146, 2019.

[3] C. M. Bishop, "Mixture density networks," Aston University, Tech. Rep., 1994.

[4] Y. Zhou, Z. Li, S. Xiao, C. He, Z. Huang, and H. Li, "Auto-conditioned recurrent networks for extended complex human motion synthesis," in *International Conference on Learning Representations*, 2018.

[5] P. Kratzer, M. Toussaint, and J. Mainprice, "Towards combining motion optimization and data driven dynamical models for human motion prediction," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 202–208.