An Algorithmic Perspective on Imitation Learning

——Part 3

温昭晋 2021.01.28

Behavioral Cloning

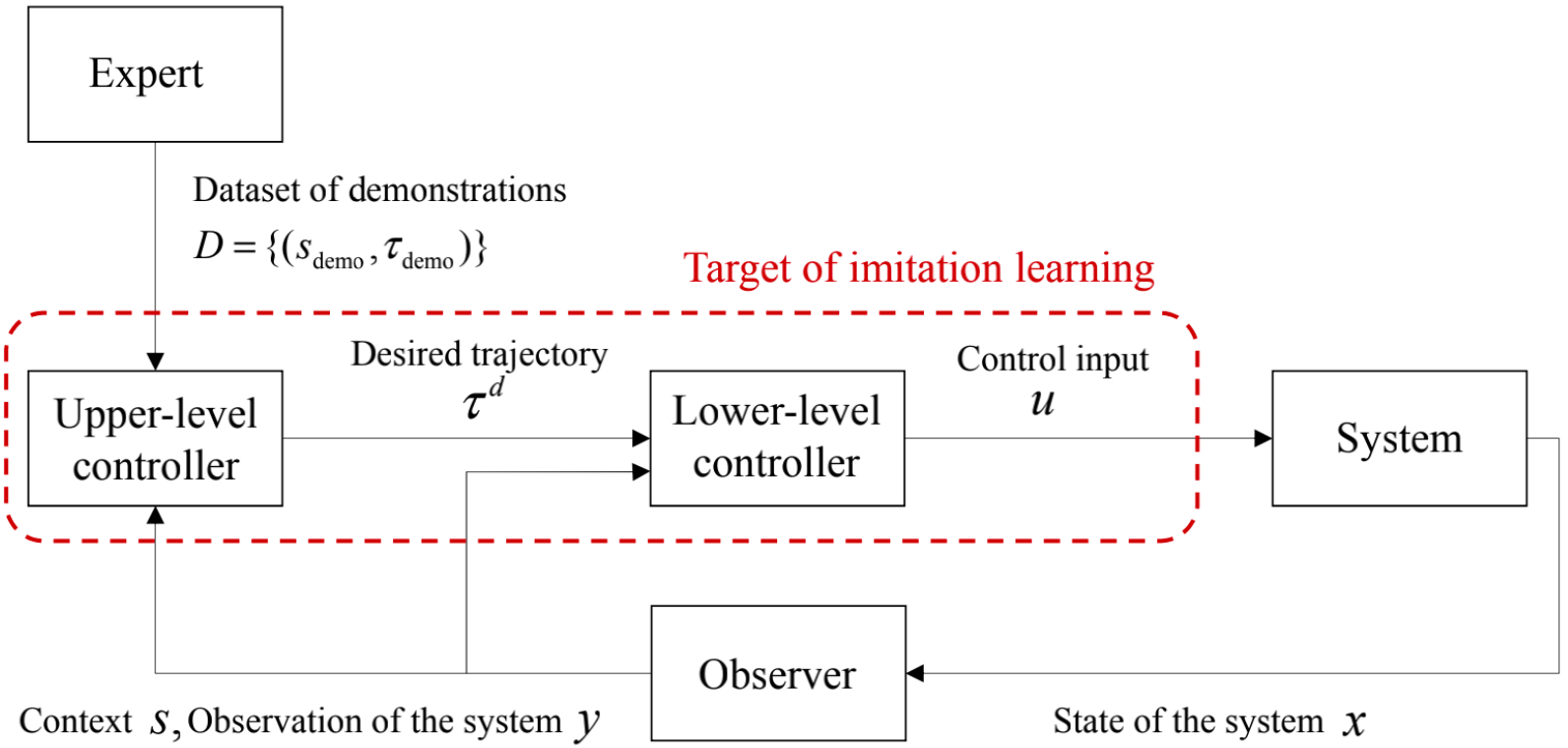
$$state/context \rightarrow trajectories/actions$$

- Model-free BC methods

- Model-based BC methods

3.1 问题陈述

机器人系统的主要目的：学习控制器



3.1 问题陈述

The aim of imitation learning:

$$\tau^d = \pi(s)$$

s *can be the initial state of the robotic manipulator* x_0 *or the state of objects relevant to a given task*

In action-state space learning:

$$u_t = \pi(x_t, s)$$

3.1 问题陈述

Learning trajectories: $\mathcal{D} = \{(\tau_i, s_i)\}_{i=1}^{N}$

Action-state space learning: $\mathcal{D} = \{(u_i, x_i)\}_{i=1}^{N}$

Algorithm 1 Abstract of behavioral cloning

Collect a set of trajectories demonstrated by the expert $\mathcal{D}$
Select a policy representation $\pi_{\boldsymbol{\theta}}$
Select an objective function $\mathcal{L}$
Optimize $\mathcal{L}$ w.r.t. the policy parameter $\boldsymbol{\theta}$ using $\mathcal{D}$
return optimized policy parameters $\boldsymbol{\theta}$

3.2 行为克隆的设计

Questions:

1.应该使用什么代理损失函数来表示演示和生成的行为的差异：

BC方法需要一个代理损失函数，该函数量化演示的行为和学习的策略之间的差异。 代理损失函数的选择对如何训练策略有很大的影响，我们需要选择合适的代理损失函数来实现高效的学习。

2.应该使用什么回归方法来表示策略：

为了获得满意的系统性能，必须选择合适的回归方法。回归模型应该具有足够的可解释性，以表示所需的行为，但要足够简单，以便对模型进行有效的训练。

3.2 行为克隆的设计

3.2.1 代理损失函数的选择

Quadratic Loss Function:

$$l_{quad}(x_1, x_2) = (x_1 - x_2)^T (x_1 - x_2)$$

在高斯分布假设下，最小化平方损失函数与期望对数似然(expected log likelihood) 的最大化密切相关

$$y = f_\theta(x) + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma)$$

$$p(y|x, \theta) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y - f_\theta(x))^2}{2\sigma}\right)$$

3.2 行为克隆的设计

$$\begin{aligned}\underset{\theta}{\operatorname{argmax}} E[logp] &= \underset{\theta}{\operatorname{argmax}} E[logexp(-\frac{(y-f_\theta(x))^2}{2\sigma})^2] \\ &= \underset{\theta}{\operatorname{argmin}} E[(y-f_\theta(x))^2] \\ &\approx \operatorname{argmin}_\theta \frac{1}{N}\sum_i (y-f_\theta(x))^2\end{aligned}$$

DMP和ProMP都是通过最小化平方损失函数学习轨迹

Weighted Quadratic Loss Function:

$$l_{wquad}(x_1, x_2, W) = (x_1 - x_2)^T W\ (x_1 - x_2)$$

3.2 行为克隆的设计

$l_1 - Loss\ Function$:

$$l_{abs}(x_1, x_2) = \sum_i |x_{1,i} - x_{2,i}|$$

Log Loss Function:

$$l_{log}(q, p) = -\sum_i q_i ln p_i$$

3.2 行为克隆的设计

Hinge Loss Function:

$$l_{hinge}(x_1, x_2) = \max(0, 1 - x_1 x_2)$$

Kullback-Leibler Divergence:

$$D_{KL}(p(x)||q(x)) = \int p(x) ln \frac{p(x)}{q(x)} dx$$

3.2 行为克隆的设计

3.2.2 回归方法的选择

Trajectory Learning	Gaussian Model	[Paraschos et al., 2013, Maeda et al., 2016]
	GMR	[Calinon and Billard, 2009, Gribovskaya et al., 2011, Khansari-Zadeh and Billard, 2014, Calinon, 2016]
	LWR	[Schaal and Atkeson, 1998, Mülling et al., 2013, Osa et al., 2017a]
	LWPR	[Vijayakumar et al., 2005]
	GPR	[Osa et al., 2017b]
Action-State Space	Look-Up Table	[Chambers and Michie, 1969]
	Linear Regression	[Widrow and Smith, 1964]
	Neural Network	[Pomerleau, 1988, LeCun et al., 2006, Stadie et al., 2017, Duan et al., 2017]
	Decision Tree	[Sammut et al., 1992]
	LWR	[Atkeson and Schaal, 1997]
	LWPR	[Vijayakumar and Schaal, 2000]

3.3基于模型和无模型的行为克隆方法

无模型BC不需要学习系统动态，不需要迭代学习，学习过程相对简单，但并不能保证在给定系统中的可行性

基于模型BC使用系统动态相关知识学习策略，即使机器欠驱动，也可达到学得接近专家策略的行为轨迹，但往往需要迭代学习，较为耗时

	Model-free	**Model-based**
Advantages	A policy can be usually learned without iterative learning.	Applicable to underactuated systems.
Disadvantages	Hard to apply to underactuated systems. Hard to predict future states.	Model learning can be very difficult. An iterative learning process is often required.

3.4动作状态空间中的无模型BC Methods

3.4.1Model-Free Behavioral Cloning as Supervised Learning

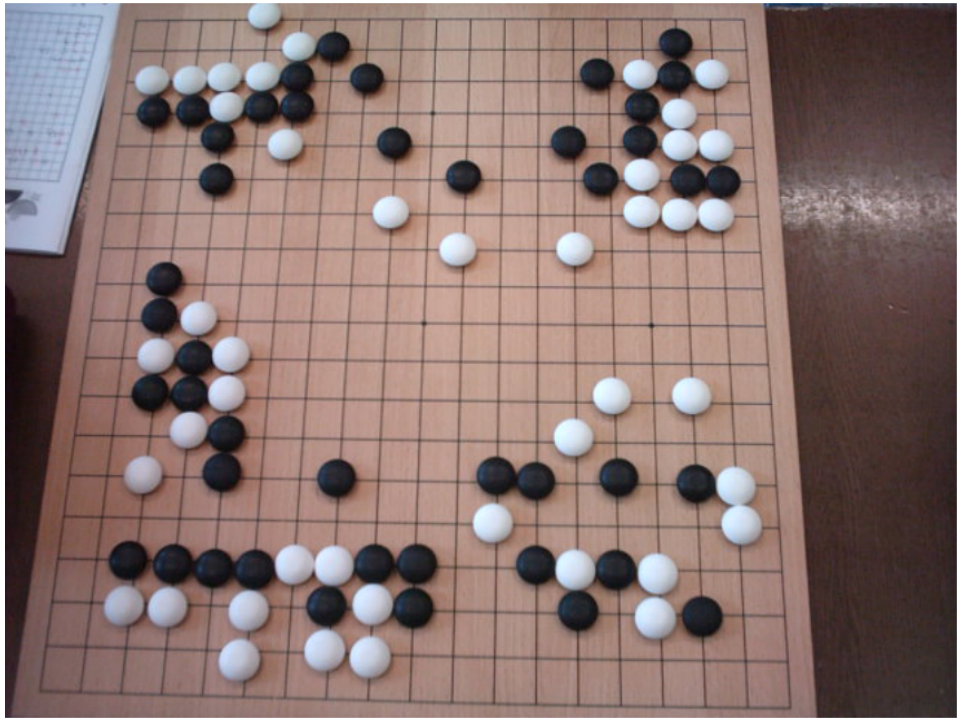
在早期对模仿学习的研究中，采用监督学习的方法在动作状态空间中进行模仿学习，例如在自动驾驶系统中，训练一个神经网络实现从图像到转向角的映射，但这种方法无法用于实践，在顺序决策中产生的极联误差会让学习者遇到未知状态，汽车会很快的驶离道路。

监督学习通常基于训练样本是独立、同分布的假设，但模仿学习的问题会违背这种假设，尤其是在学习一个需要做出顺序决策的策略时，3.4.3将介绍一种交互监督学习(supervised learning with interaction)的方法解决这个问题

3.4动作状态空间中的无模型BC Methods

3.4.2 Imitation as Supervised Learning with Neural Networks

- *Recent successes of Imitation Learning with Neural Networks*



a value network:
approximate the value function to predict the expected outcome of the game

a policy network:
output actions using a representation of the image input of the board

The value and policy networks are improved using data collected through self-play

3.4动作状态空间中的无模型BC Methods

3.4.2 Imitation as Supervised Learning with Neural Networks

- *Learning with Recurrent Neural Networks*

Dialogue act:
inform(name="red door cafe", goodformeal="breakfast", area="cathedral hill", kidsallowed="no")
Generated samples:
red door cafe is a good restaurant for breakfast in the area of cathedral hill and does not allow children .
red door cafe is a good restaurant for breakfast in the cathedral hill area and does not allow children .
red door cafe is a good restaurant for breakfast in the cathedral hill area and does not allow kids .
red door cafe is good for breakfast and is in the area of cathedral hill and does not allow children .
red door cafe does not allow kids and is in the cathedral hill area and is good for breakfast .

generate human like natural language using a special form of the long short-term memory(LSTM)

神经网络生成从人类演示中学习的自然语言。该神经网络以对话行为为条件，将生成的句子限制在特定的意义下

3.4动作状态空间中的无模型BC Methods

3.4.3 Teacher-Student Interaction during Behavioral Cloning

This method highlights a central difference between imitation learning and the traditional setting of supervised learning, where we typically assume the input distribution to be independent and identically distributed.

但是，即使在适度规模的模仿学习问题中，在所有可能情况下收集演示数据也是不可行的，因此我们必须把更正集中在最相关的场景下。

- *SEARN*
- *Confidence-Based Approach*
- *Data Aggregation Approach: DAGGER*

3.4动作状态空间中的无模型BC Methods

3.4.3 Teacher-Student Interaction during Behavioral Cloning

- *Confidence-Based Approach*

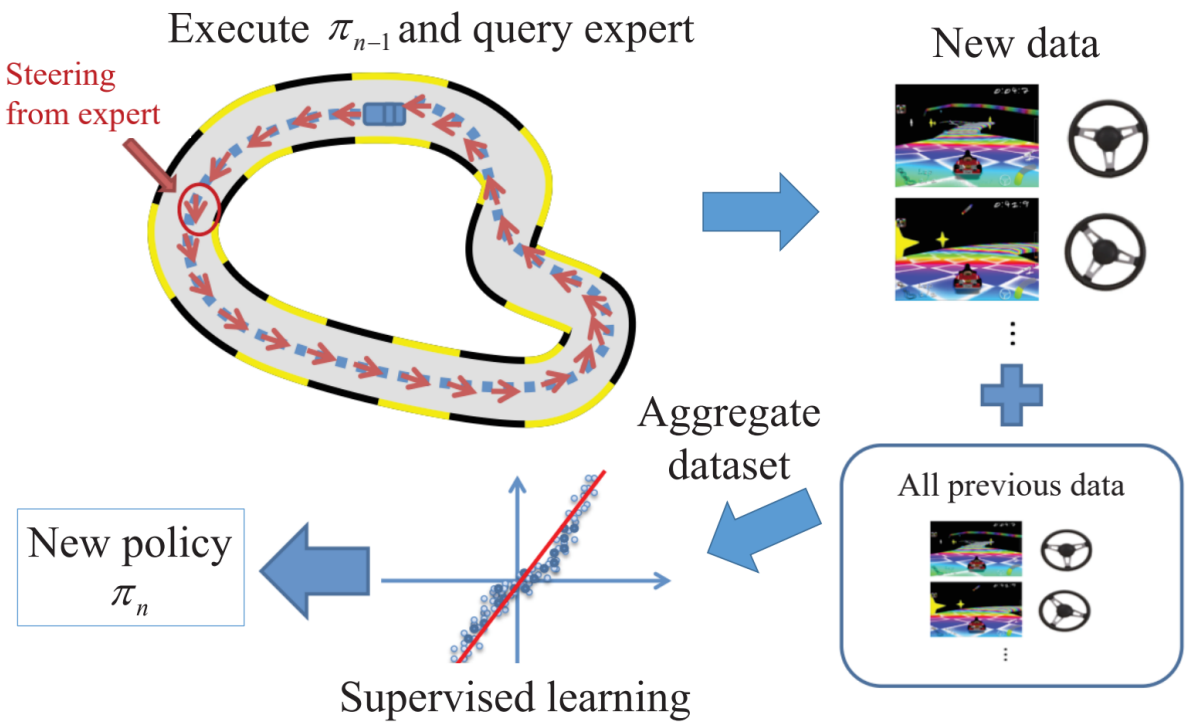
Algorithm 2 Confidence-based autonomy algorithm: confident execution and corrective demonstration [Chernova and Veloso, 2009]

Input: Demonstration of the action-state pairs $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{u}_i)\}_{i=1}^{N}$, confidence threshold c_0
Initialize the policy π
repeat
  Observe the state $\boldsymbol{x}$
  Compute the confidence $c(\boldsymbol{x})$
  Plan action $\boldsymbol{u}^{\mathrm{L}}$
  if $c(\boldsymbol{x}) < c_0$ **or** Corrective demonstration is necessary **then**
    Receive the demonstration data $\mathcal{D}_{\mathrm{new}} = \{(\boldsymbol{x}^{\mathrm{new}}, \boldsymbol{u}^{\mathrm{new}})\}$
    Update the dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_{\mathrm{new}}$
    Update the policy π^{L}
  end if
until the task learned

3.4动作状态空间中的无模型BC Methods

3.4.3 Teacher-Student Interaction during Behavioral Cloning

- *Data Aggregation Approach: DAGGER*



3.4动作状态空间中的无模型BC Methods

3.4.3 Teacher-Student Interaction during Behavioral Cloning

- *Data Aggregation Approach: DAGGER*

Algorithm 3 DAGGER [Ross et al., 2011]

Input: initial dataset of demonstrations $\mathcal{D} = \{(\boldsymbol{x}, \boldsymbol{u})\}$, $\{\beta_i\}$ such that $\frac{1}{N}\sum_{i=1}^{N}\beta_i \to 0$
Initialize: π_1^{L}
for $i = 1$ **to** N **do**
  Let $\pi_i = \beta_i \pi^{\mathrm{E}} + (1-\beta_i)\pi_i^{\mathrm{L}}$.
  Sample trajectories $\boldsymbol{\tau} = [\boldsymbol{x}_0, \boldsymbol{u}_0, ..., \boldsymbol{x}_T, \boldsymbol{u}_T]$ using π_i
  Get dataset $\mathcal{D}_i$ of visited states by π_i and actions given by expert.
  Aggregate datasets: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$
  Train the policy π_{i+1}^{L} on $\mathcal{D}$.
end for
return best π_i^{L} on validation.

3.5学习轨迹的无模型BC Methods

3.5.1 Trajectory Representation

- Keyframe/Via-Point Based Approaches
- Hidden Markov Models(HMM)
- Dynamic Movement Primitives(DMP)
- Probabilistic Movement Primitives(ProMPs)
- Stable estimator of dynamical systems(SEDS)

3.5学习轨迹的无模型BC Methods

3.5.1 Trajectory Representation

- *Hidden Markov Models(HMM)*

X : a finite set of latent state
Y : a finite set of observation labels
A=$\{a_{ij}\}$: a state transition matrix
B=$\{b_{ij}\}$: an output probability matrix
d_i : an initial distribution vector
$\lambda = (A, B)$

$$\lambda^* = \underset{\lambda}{\operatorname{argmax}}\, p(Y'|\lambda)$$

3.5学习轨迹的无模型BC Methods

3.5.1 Trajectory Representation

- *Dynamic Movement Primitives(DMP)*

ensure the smoothness and continuity of the trajectory

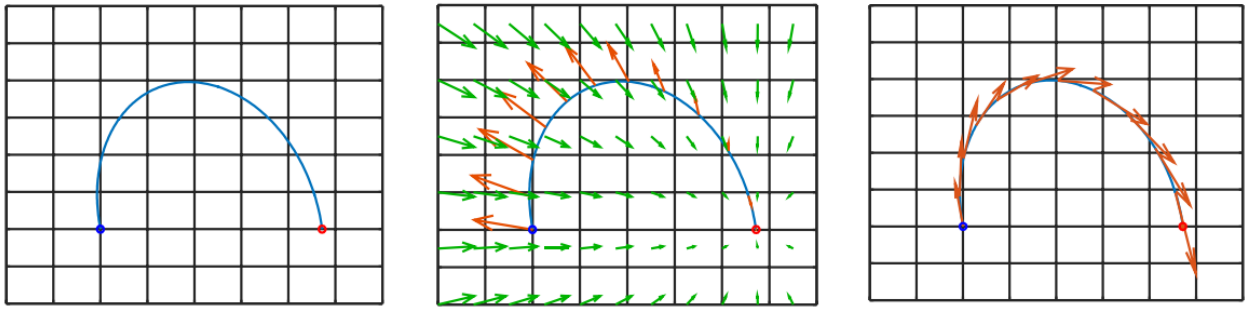


图3.4：DMP的示意图。 DMPs将演示的运动表示为非线性力项和吸引子力项的组合。 蓝点和红点分别代表开始和目标位置。 假设(A)中所示的轨迹作为演示轨迹给出。 沿轨迹的非线性力项，它依赖于运动的相位，显示为橙色矢量，在(b)绿色矢量表示吸引子力项，它是平稳的，依赖于系统的状态。 演示运动的动力学是作为(C)中所示的这些项的总和来学习的)。

3.5学习轨迹的无模型BC Methods

3.5.1 Trajectory Representation

- *Dynamic Movement Primitives(DMP)*

We describe details of DMPs in the following. In a DMP, the demonstrated motion with one degree of freedom (DoF) is modeled as a spring-damper system

$$\tau^2\ddot{x} = \alpha_x\left(\beta_x(g-x) - \tau\dot{x}\right) + f, \tag{3.22}$$

For a striking movement,

$$\tau\dot{z} = -\alpha_z z \qquad \alpha_z is\ a\ constant$$

Using a Gaussian basis function with z,

$$f(z) = (g - x_0)\sum_{i=1}^{M}\psi_i(z)\omega_i z$$

3.5学习轨迹的无模型BC Methods

3.5.1 Trajectory Representation

- *Dynamic Movement Primitives(DMP)*

$$f_{target}(t) = \tau^2 \ddot{x}^{demo}(t) - \alpha_x(\beta_x\left(g - x^{demo}(t)\right) - \tau\dot{x}^{demo}(t))$$

$$\mathcal{L}_{DMP} = \sum_{t=0}^{T}(f_{target}(t) - \xi(t)\Psi\omega)$$

3.5学习轨迹的无模型BC Methods

3.5.1 Trajectory Representation

- *Dynamic Movement Primitives(DMP)*

Algorithm 4 Learning dynamic movement primitives [Schaal et al., 2004, Ijspeert et al., 2013]

Input: demonstrated trajectory $\boldsymbol{\tau}^{\text{demo}}$, parameters $\alpha_x, \beta_x, \tau, \alpha_z, \omega_z$
Choose a system of a phase variable z, e.g., (3.23)
Choose a basis function ψ of the forcing function f
Compute the forcing function at each time step using $\boldsymbol{\tau}^{\text{demo}}$ with (3.27)
Find a weight vector $\boldsymbol{w}$ that minimize $\mathcal{L}_{\text{DMP}}$ in (3.28) using a least-square solution (3.29)

3.5学习轨迹的无模型BC Methods

3.5.1 Trajectory Representation

- *Probabilistic Movement Primitives(ProMPs)*

在t时刻，系统的状态可表示为：

$$\boldsymbol{x}(t) = \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} = \boldsymbol{\Psi}(t)^\top \boldsymbol{\omega} + \boldsymbol{\epsilon}_x, \tag{3.34}$$

where $\boldsymbol{\Psi}(t)$ is a $M \times 2$ dimensional time-dependent basis matrix defined as

$$\boldsymbol{\Psi}(t) = \begin{bmatrix} \psi_1(t) & \dot{\psi}_1(t) \\ \vdots & \vdots \\ \psi_M(t) & \dot{\psi}_M(t) \end{bmatrix}, \tag{3.35}$$

$\boldsymbol{\omega}$ is a weight vector, and $\epsilon_x \sim \mathcal{N}(0, \boldsymbol{\Sigma}_x)$ is zero-mean i.i.d. Gaussian noise. Here, the probability of observing the state $\boldsymbol{x}(t)$ is expressed as

$$p(\boldsymbol{x}(t)|\boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{x}(t)|\boldsymbol{\Psi}(t)^\top \boldsymbol{\omega}, \boldsymbol{\Sigma}_x). \tag{3.36}$$

3.5学习轨迹的无模型BC Methods

3.5.1 Trajectory Representation

- *Probabilistic Movement Primitives(ProMPs)*

$$\mathcal{L}_{\text{ProMP}} = \sum_{t=0}^{T} \left\| \boldsymbol{x}(t) - \boldsymbol{\Psi}(t)^{\top} \boldsymbol{w} \right\|^2, \tag{3.39}$$

where $\boldsymbol{x}(t) = [q(t)\ \dot{q}(t)]^{\top}$. The solution is given by a least squares solution

$$\boldsymbol{\omega}^i = \left(\boldsymbol{\Gamma}\boldsymbol{\Gamma}^{\top}\right)^{-1} \boldsymbol{\Gamma} \begin{bmatrix} q^i(0) \\ \dot{q}^i(0) \\ \vdots \\ q^i(T) \\ \dot{q}^i(T) \end{bmatrix}, \tag{3.40}$$

where the basis function matrix $\boldsymbol{\Gamma}$ is given by

$$\boldsymbol{\Gamma} = \begin{bmatrix} \psi_1(0) & \dot{\psi}_1(0) & \cdots & \psi_1(T) & \dot{\psi}_1(T) \\ \vdots & & \ddots & & \vdots \\ \psi_M(0) & \dot{\psi}_M(0) & \cdots & \psi_M(T) & \dot{\psi}_M(T) \end{bmatrix}. \tag{3.41}$$

3.5学习轨迹的无模型BC Methods

3.5.1 Trajectory Representation

- *Probabilistic Movement Primitives(ProMPs)*

Algorithm 5 Learning probabilistic movement primitives [Paraschos et al., 2013]

Input: Multiple demonstrated trajectories $\mathcal{D} = \{\boldsymbol{\tau}_i^{\text{demo}}\}_{i=1}^{N}$
Choose a basis function ψ and the number of the basis function M
Compute the basis function matrix $\boldsymbol{\Psi}(t)$
for each demonstrated trajectory **do**
  Obtain $\boldsymbol{\omega}$ by computing (3.40)
end for
Compute $p(\boldsymbol{\omega}) \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\omega}}, \boldsymbol{\Sigma}_{\boldsymbol{\omega}})$

3.5学习轨迹的无模型BC Methods

3.5.1 Trajectory Representation

Time-Invariant

- *Stable estimator of dynamical systems(SEDS)*

$$\dot{x} = f(x)$$

Let us define $\boldsymbol{x}$ as the state vector of the system. When a set of demonstrated trajectories is given, the joint distribution of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ can be estimated from the observations using a GMM. The kth component of the GMM models the distribution $p(\boldsymbol{x}, \dot{\boldsymbol{x}}|k)$ as

$$p(\boldsymbol{x}, \dot{\boldsymbol{x}}|k) \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x} \\ \dot{\boldsymbol{x}} \end{bmatrix} \middle| \middle| \begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{x}} \\ \boldsymbol{\mu}_{\dot{\boldsymbol{x}}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{x},k} & \boldsymbol{\Sigma}_{\boldsymbol{x}\dot{\boldsymbol{x}},k} \\ \boldsymbol{\Sigma}_{\dot{\boldsymbol{x}}\boldsymbol{x},k} & \boldsymbol{\Sigma}_{\dot{\boldsymbol{x}},k} \end{bmatrix} \right). \tag{3.45}$$

The estimated dynamics function $\hat{\boldsymbol{f}}$ is learned as

$$\hat{\boldsymbol{f}} = \sum_{k=1}^{K} h_k(\boldsymbol{x}) \left(\boldsymbol{\mu}_{\dot{\boldsymbol{x}}} + \boldsymbol{\Sigma}_{\dot{\boldsymbol{x}}\boldsymbol{x},k} \boldsymbol{\Sigma}_{\boldsymbol{x},k}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x},k}) \right), \tag{3.46}$$

where

$$h_k(\boldsymbol{x}) = \frac{p(k)p(\boldsymbol{x}|k)}{\sum_{i=1}^{K} p(i)p(\boldsymbol{x}|i)} = \frac{\pi_x \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{\boldsymbol{x},k}, \boldsymbol{\Sigma}_{\boldsymbol{x},k})}{\sum_{i=1}^{K} \pi_i \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{\boldsymbol{x},i}, \boldsymbol{\Sigma}_{\boldsymbol{x},i})}, \tag{3.47}$$

where π_k is the prior of the kth Gaussian component.

3.5学习轨迹的无模型BC Methods

3.5.1 Trajectory Representation

Time-Invariant

- *Stable estimator of dynamical systems(SEDS)*

The study by Khansari-Zadeh and Billard [2011] showed that the system described by (3.46) is globally asymptotically stable at the target $\boldsymbol{x}^*$ if the condition

$$\begin{cases} \boldsymbol{A}^k + (\boldsymbol{A}^k)^\top \text{is negative definite}, \\ -\boldsymbol{A}^k \boldsymbol{x}^* = \boldsymbol{\mu}_{\dot{\boldsymbol{x}},k} - \boldsymbol{A}^k \boldsymbol{\mu}_{\boldsymbol{x},k}, \end{cases} \tag{3.48}$$

is satisfied for all $k = 1, \ldots, K$ where $\boldsymbol{A}^k = \boldsymbol{\Sigma}_{\dot{\boldsymbol{x}}\boldsymbol{x},k}(\boldsymbol{\Sigma}_{\boldsymbol{x},k})^{-1}$.

3.5学习轨迹的无模型BC Methods

3.5.2 Comparison of Trajectory Representations

选择要求：

1. 选择最简洁的轨迹描述
2. 选择适合所需行为模型复杂度的表示

	Time dependence	Stabile attraction to a target position	Stochasticity of trajectories	Encoding spatial co-ordination patterns
Way points / Keyframe [Abbeel et al., 2010, Nakaoka et al., 2007]	✓	-	-	-
HMMs [Inamura et al., 2004, Takano and Nakamura, 2015]	(✓)	-	✓	✓
DMP [Schaal et al., 2004, Ijspeert et al., 2013]	✓	✓	-	-
ProMP [Paraschos et al., 2013, Maeda et al., 2016]	✓	-	✓	✓
SEDS [Khansari-Zadeh and Billard, 2011, 2014]	-	✓	-	✓

3.5学习轨迹的无模型BC Methods

3.5.3 Generalization of Demonstrated Trajectories

Method	Generalizable context	Advantages	Disadvantages
DMP [Schaal et al., 2004, Ijspeert et al., 2013]	Start and goal positions	Guarantee of stable behavior	Limited generalization capabilities
ProMP [Paraschos et al., 2013, Maeda et al., 2016]	Any subset of the observations of the system	Generalization based on stochasticity of demonstrations	No guarantee of stable behavior
SEDS [Khansari-Zadeh and Billard, 2011, 2014]	State of the system with fixed dimensionality	Generalization with guarantee of stable behavior	No time-dependence
Way points with non-rigid registration [Schulman et al., 2013]	A point cloud of the given scene	Generalization based on point clouds of a given scene	Stochasticity of demonstrations is not considered

3.5.4 Information Theoretic Understanding of Model-Free BC

minimize a sum-of-squares error function
=maximize the likelihood for the given dataset of demonstration
=minimize the KL divergence given by:

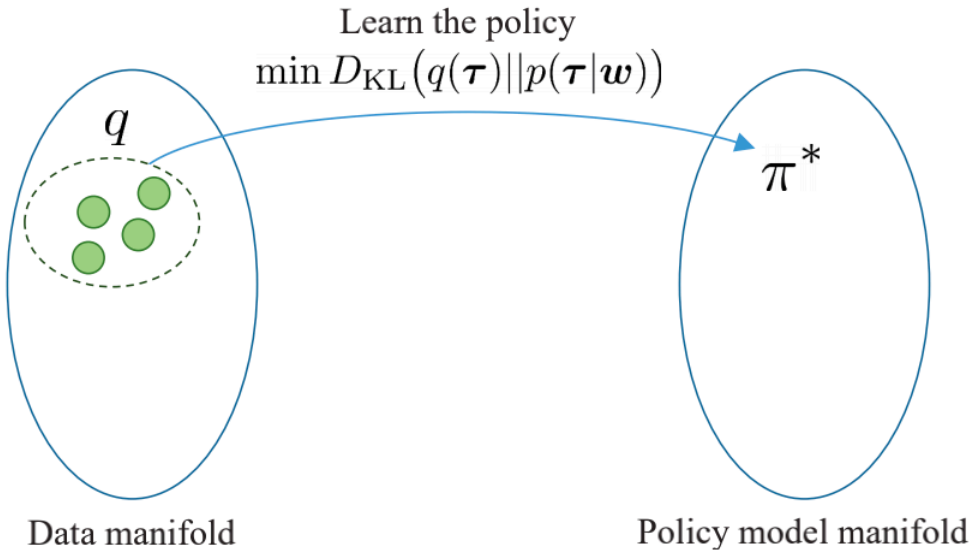
$$D_{KL}(q(\tau)||p(\tau|w)) = \int q(\tau)\, ln\frac{q(\tau)}{p(\tau|\omega)}\, d\tau$$

$$D_{KL}(q(\tau)||p(\tau|\omega)) \simeq \frac{1}{N}\sum_{i=1}^{N}(-\ln(\tau_i^{demo}|\omega) + lnq(\tau_i^{demo}))$$

3.5学习轨迹的无模型BC Methods

3.5.4 Information Theoretic Understanding of Model-Free BC

$$maximize\ the\ likelihood\ \ln p(\tau|\omega)$$



3.5学习轨迹的无模型BC Methods

- Time Alignment of Multiple Demonstrations
- Learning Coupled Movements
- Incremental Trajectory Learning

3.5学习轨迹的无模型BC Methods

3.5.8 Combing Multiple Expert Policies

Given multiple experts' policies $\{\pi_i\}_{i=1}^{M}$,

mixture of experts: $\pi(x) = \frac{\sum_{i=1}^{M} o_i \pi_i(x)}{\sum_{i=1}^{M} o_i}$

products of experts: $\pi(x) = \frac{\prod_{i=1}^{M} \pi_i(x)}{\int \prod_{i=1}^{M} \pi_i(x) dx}$

3.6 Model-Free Behavioral Cloning for Task-Level Planning

3.6.1 Segmentation and Clustering for Task-Level Planning

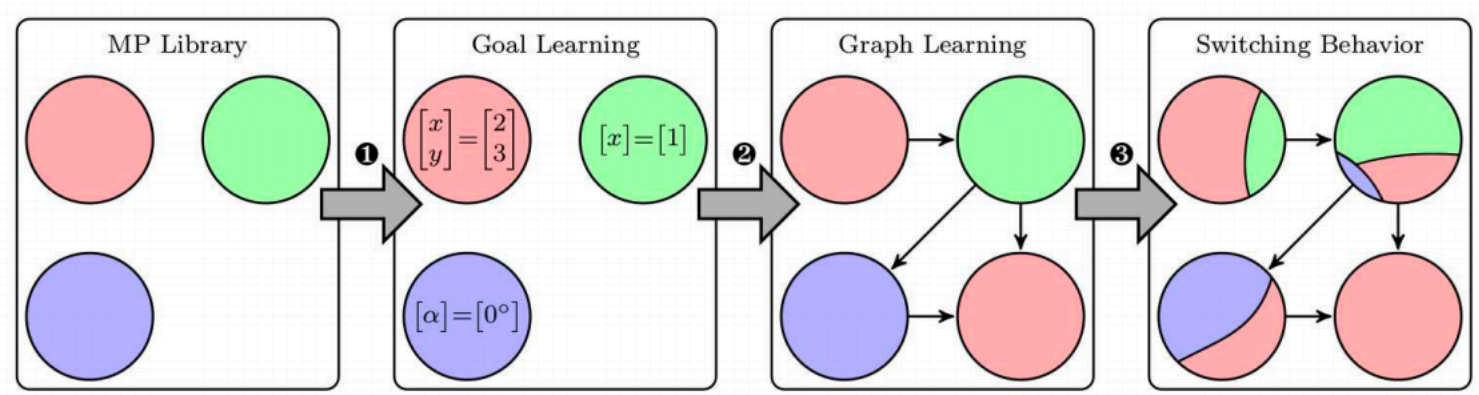
虽然无模型的轨迹学习方法通常隐含地假设每个演示的轨迹包含一个运动，但在实践中，演示的轨迹可能由不同类型的原始运动序列组成。因此，为了学习每个原始运动，有必要对演示的轨迹进行分割。 此外，在对轨迹进行分割后，为了学习多种类型的原始运动，往往需要对分割后的运动进行聚类。然而，手工分割和聚类轨迹往往是耗时的。

基于HMM的在线分割方法：通过邻近数据间距离的计算，对人类行为的数据用无监督学习方法进行分割

利用分解HMMs分割和聚类全身运动的方法：计算HMMs之间 的距离，并将分割后的观察到的运动聚类成树结构

3.6 Model-Free Behavioral Cloning for Task-Level Planning

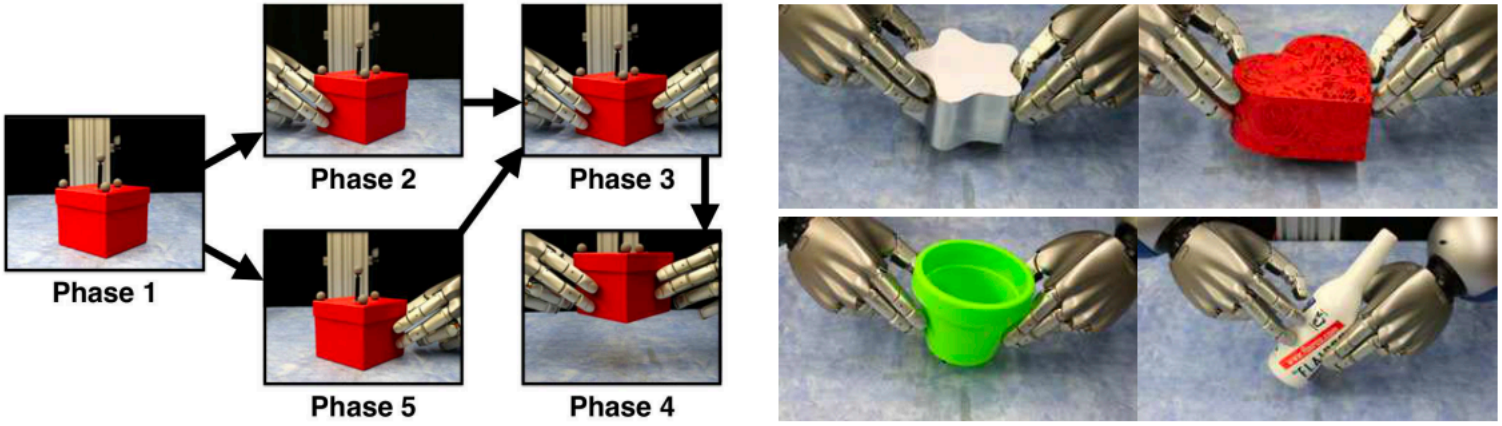
3.6.2 Learning a Sequence of Primitive Motions



从示例中学习不同movement primitives，并用支持向量机求解多分类问题建模选择下一个动作

3.6 Model-Free Behavioral Cloning for Task-Level Planning

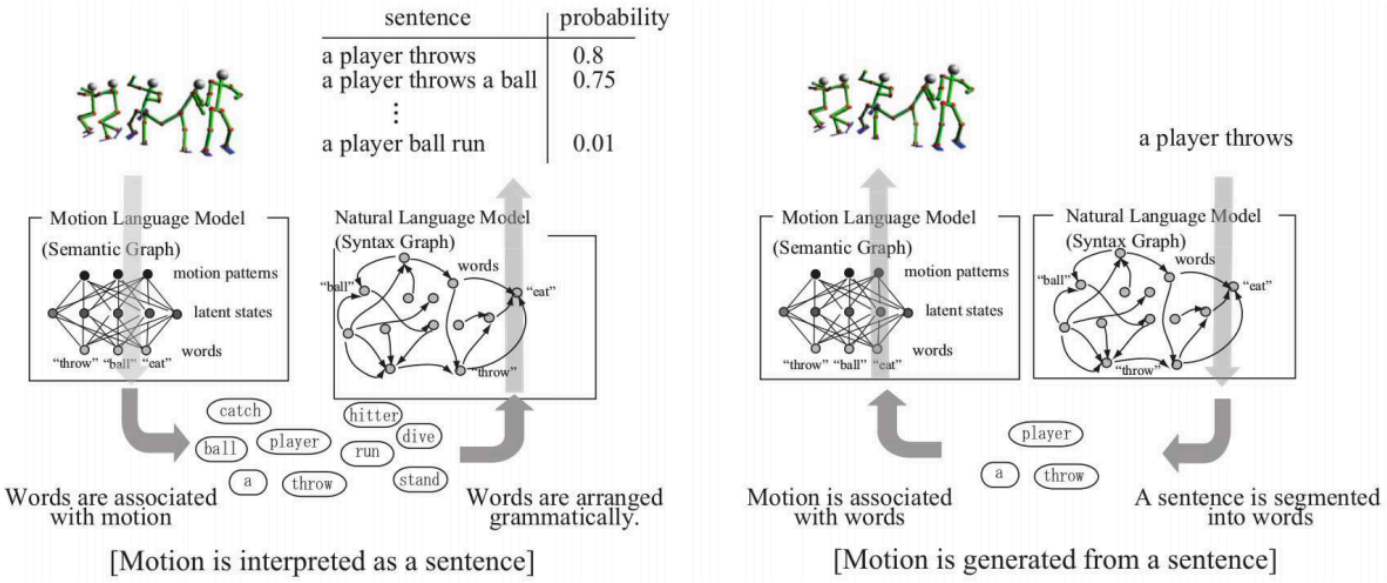
3.6.2 Learning a Sequence of Primitive Motions



使用HMM对MP之间的转换进行建模，学习技能系列；
并用学习的DMPs使其能够适应不同物体

3.6 Model-Free Behavioral Cloning for Task-Level Planning

3.6.2 Learning a Sequence of Primitive Motions



从运动中产生句子或从句子中产生运动

3.6 Model-Free Behavioral Cloning for Task-Level Planning

Algorithm 10 Motion language model [Takano and Nakamura, 2015]

Learning:
Input: demonstrated trajectories and sentences $\mathcal{D} = \{\boldsymbol{\tau}^{\text{demo}}, \boldsymbol{y}\}$
Train a set of HMMs that represent the primitive motions
Train the motion language model and the natural language model
Prediction:
Input: a motion sequence or a sentence
if the given input is a motion sequence **then**
  Recognize the motion symbol λ^{in} using HMMs
  Predict words for the given motion
    $\boldsymbol{y}^* = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\lambda^{in})$
  Arrange the order of the words using the natural language model
  return sentence
end if
if the given input is a sentence **then**
  Predict a motion symbol corresponding to the given sentence $\boldsymbol{y}^{\text{in}}$
    $\lambda^* = \arg\max_{\lambda \in \Lambda} p(\lambda|\boldsymbol{y}^{\text{in}})$
  Predict the motion sequence from the motion symbol λ^*
  return motion sequence
end if

3.7 基于模型的行为克隆方法

3.7.1 基于向前动力学模型的行为克隆方法

Correspondence problem

Learning a forward dynamics model:

$$x_{t+1} = f(x_t, u_t)$$

forward dynamics model learning can be framed as a regression problem

Regression	Employed by ...
Locally Weighted Regression	[Atkeson et al., 1997, Schneider, 1997]
Gaussian Mixture Regression	[Grimes et al., 2006b, Grimes and Rao, 2009]
Gaussian Process Regression	[Grimes et al., 2006a, Englert et al., 2013, Deisenroth et al., 2014]
Neural Networks	[Baram et al., 2017, Nair et al., 2017]

3.7 基于模型的行为克隆方法

3.7.1 基于向前动力学模型的行为克隆方法

- Imitation with a Gaussian Mixture Forward Model

Algorithm 11 Behavior acquisition via Bayesian inference and learning [Grimes and Rao, 2009]

Observe an expert's demonstrations $[\boldsymbol{o}_1, \cdots, \boldsymbol{o}_T]$
Estimate the kinematics of the expert
Initialize the forward model f
Infer bootstrap actions based on the forward model
repeat
  Execute actions
  Learn/update the GMR forward model
  Infer constrained actions
until task learned

3.7 基于模型的行为克隆方法

3.7.1 基于向前动力学模型的行为克隆方法

- Imitation with a Gaussian Process Forward Model

Algorithm 12 Probabilistic model-based imitation learning [Englert et al., 2013]

Input: n trajectories $\boldsymbol{\tau}_i$ demonstrated by the expert
Estimate the expert distribution over trajectories $q(\boldsymbol{\tau}^{\text{demo}})$
Record state-action parts of the robot through applying random control inputs
repeat $i = 1$ **to** N **do**
  Learn/update probabilistic GP forward model
  Predict the new trajectory distribution $p(\boldsymbol{\tau})$
  Learn policy $\pi^{\text{L}} = \arg\min_{\pi} D_{\text{KL}}\left(q(\boldsymbol{\tau}^{\text{demo}})||p(\boldsymbol{\tau})\right)$
  Apply π^{L} to the system and record data
until task learned

3.7.2 Imitation Learning through Iterative Learning Control

不需要forward dynamics model

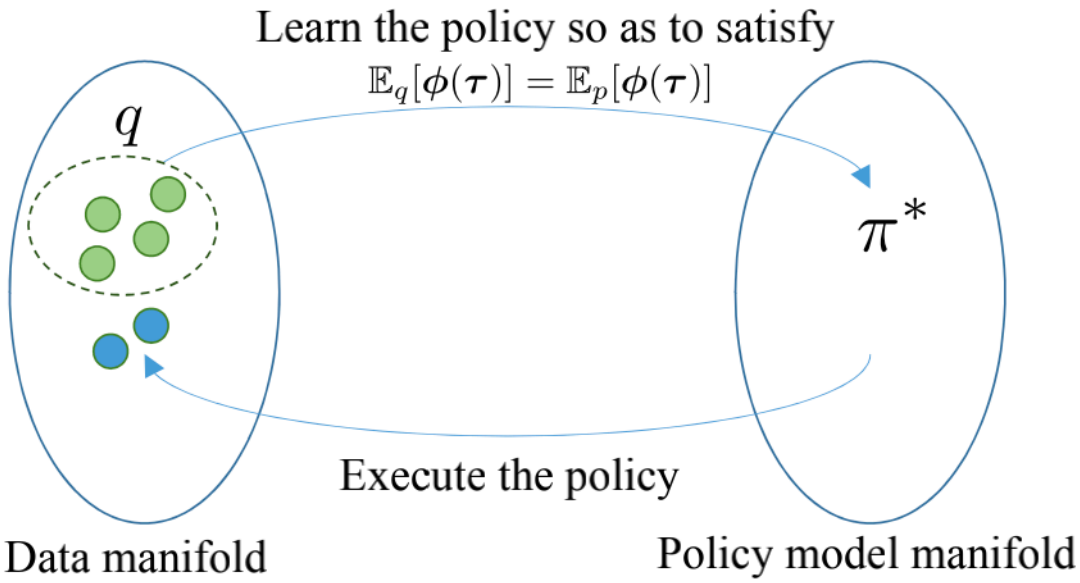
Algorithm 13 Iterative control learning [van den Berg et al., 2010]

Input: desired trajectory $\boldsymbol{\tau}^d$, learning rate α
Initialize the target trajectory as $\boldsymbol{\tau} = \boldsymbol{\tau}^d$
repeat
  Execute a controller with the target trajectory $\hat{\boldsymbol{\tau}}$
  Record the executed trajectory $\boldsymbol{\tau}$
  Update the target trajectory $\hat{\boldsymbol{\tau}} \leftarrow \hat{\boldsymbol{\tau}} - \alpha(\boldsymbol{\tau} - \boldsymbol{\tau}^d)$
until $\boldsymbol{\tau} \approx \boldsymbol{\tau}^d$

该模型无法适用于不同的期望轨迹

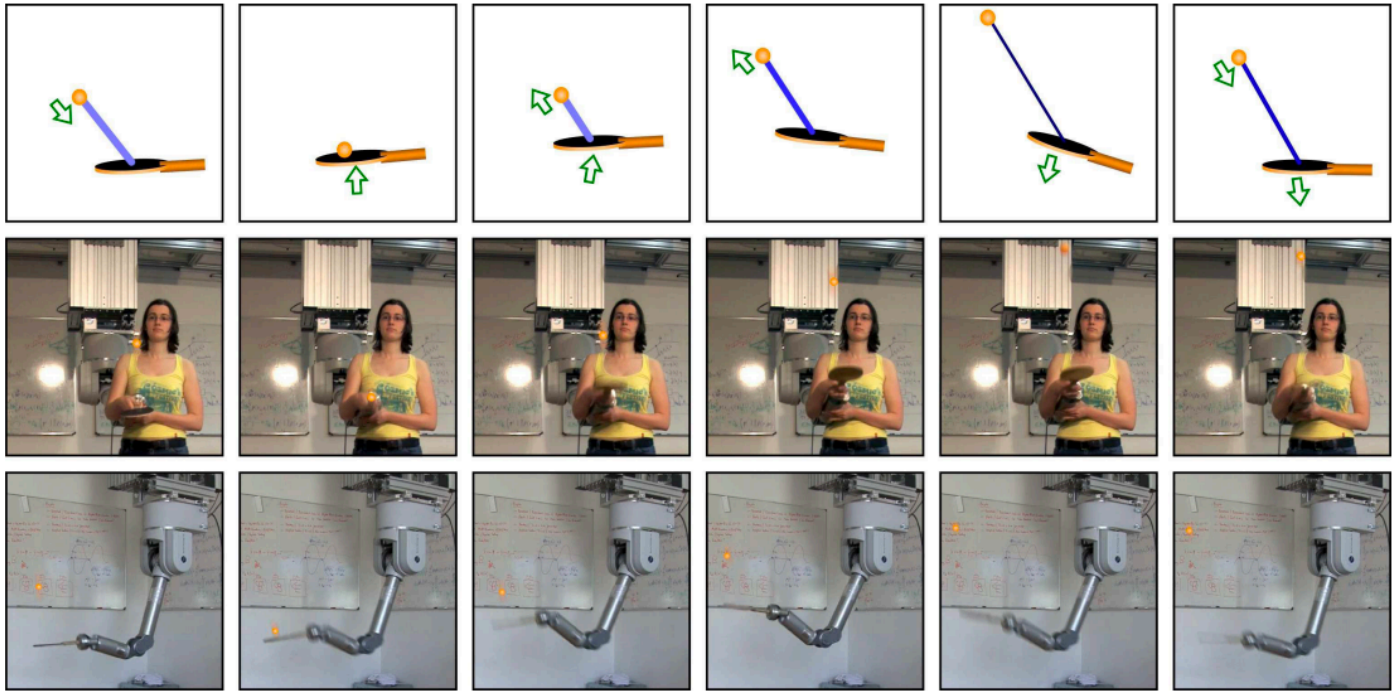
3.7 基于模型的行为克隆方法

3.7.3 基于模型BC方法的信息论理解



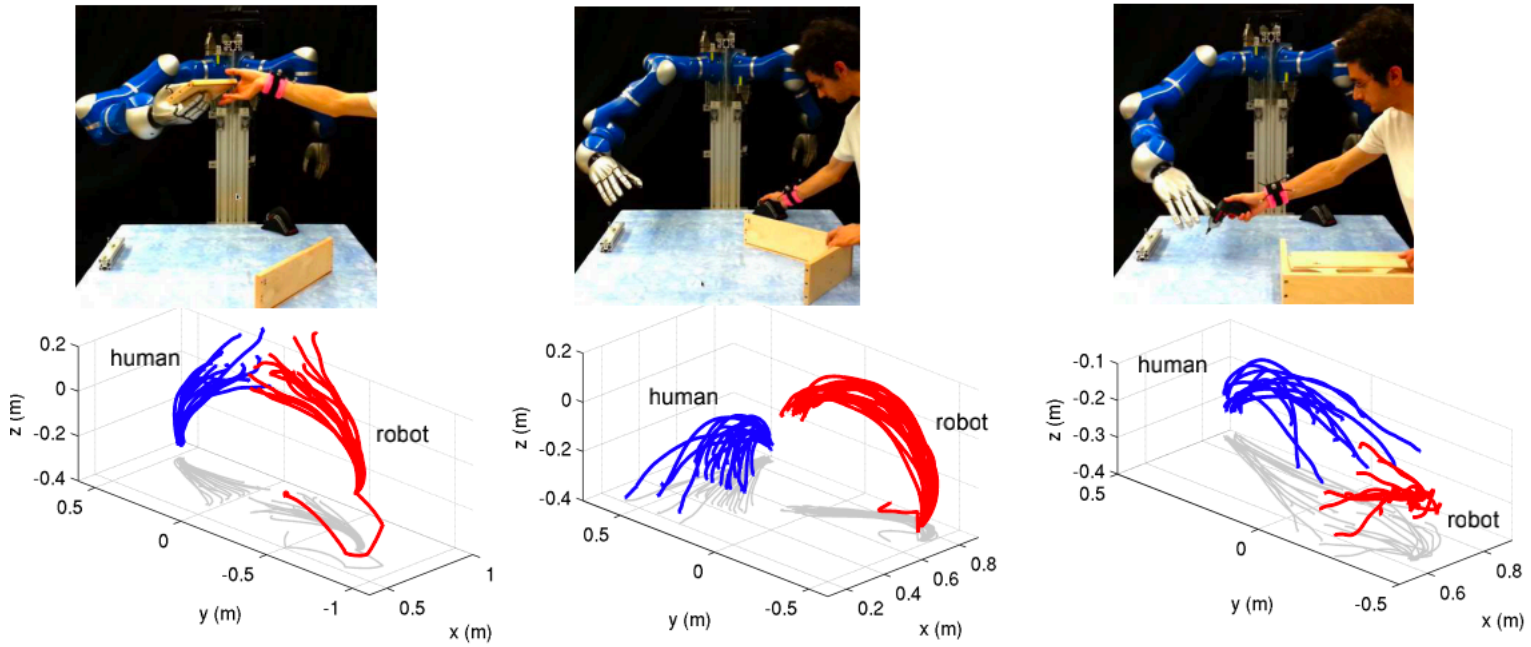
3.8 Robot Applications with Model-Free BC Methods

Learning to Hit a Ball with DMP



3.8 Robot Applications with Model-Free BC Methods

Learning Hand-Over Tasks with ProMPs



(a) Handing over a plate

(b) Handing over a screw

(c) Holding the screw driver

3.8 Robot Applications with Model-Free BC Methods

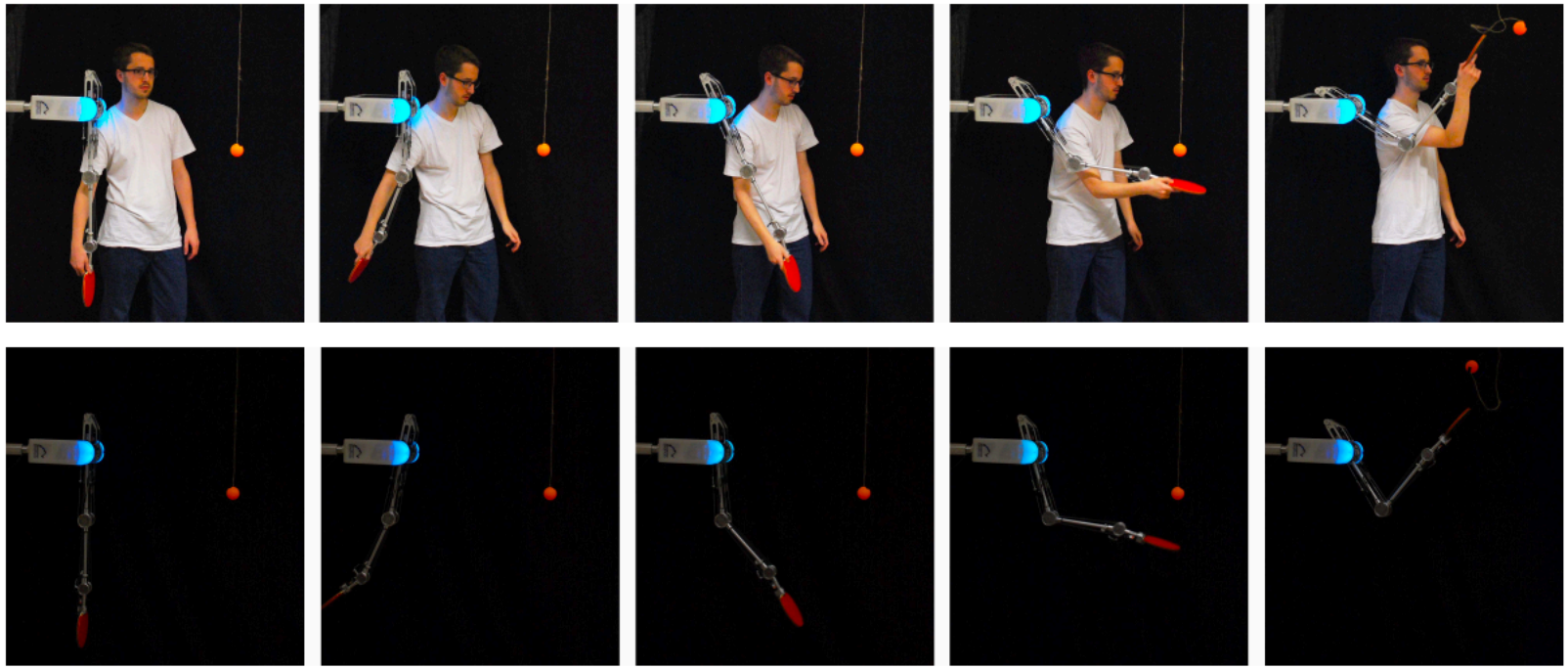
Learning To Tie a Knot Modeling the Trajectory Distribution with Gaussian Process

3.9 Robot Applications with Model-Based BC Methods

Learning Acrobatic Helicopter Flights

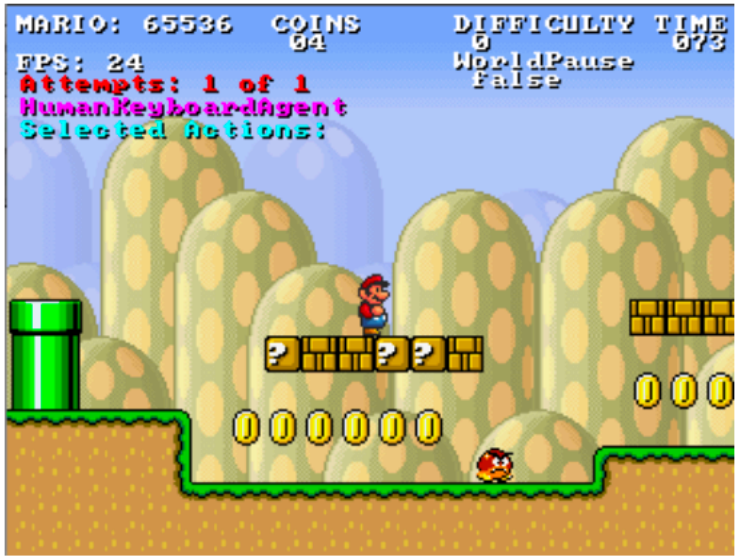
3.9 Robot Applications with Model-Based BC Methods

Learning to Hit a Ball with an Underactuated Robot



3.9 Robot Applications with Model-Based BC Methods

Learning to Control with DAGGER



Thanks!