Autonomous Robots
https://doi.org/10.1007/s10514-018-9757-y

An ensemble inverse optimal control approach for robotic task learning and adaptation

Hang Yin[1,2] · **Francisco S. Melo**[2] · **Ana Paiva**[2] · **Aude Billard**[1]

Received: 30 December 2016 / Accepted: 18 April 2018

Abstract
This paper contributes a novel framework to efficiently learn cost-to-go function representations for robotic tasks with latent modes. The proposed approach relies on the principle behind ensemble methods, where improved performance is obtained by aggregating a group of simple models, each of which can be efficiently learnedq. The maximum-entropy approximation is adopted as an effective initialization and the quality of this surrogate is guaranteed by a theoretical bound. Our approach also provides an alternative perspective to view the popular mixture of Gaussians under the framework of inverse optimal control. We further propose to enforce a dynamics on the model ensemble, using Kalman estimation to infer and modulate model modes. This allows robots to exploit the demonstration redundancy and to adapt to human interventions, especially in tasks where sensory observations are non-Markovian. The framework is demonstrated with a synthetic inverted pendulum example and online adaptation tasks, which include robotic handwriting and mail delivery.

Keywords Learning from demonstrations · Human-robot collaboration · Ensemble methods · Inverse optimal control

1 Introduction

Enabling robots to acquire novel skills from human demonstrations enhances production automation and increases the proximity between robots and humans. Learning from demonstration (LfD, a.k.a. programming by demonstration and imitation learning) addresses the acquisition of robot skills by interpreting the observed human behavior in terms of specific task constraints, which are then used to drive the robot operation. One simple yet widely used approach amounts to reproducing the observed demonstrations—which consist, for example, of state-action pairs—using supervised learning techniques. This approach is often regarded as behavior cloning (Pomerleau 1991) because a reactive behavior is directly obtained by associating the original state and action spaces. However, without a grip on the latent objective underlying these raw representations, such "naive" methods sometimes face difficulties in generalization. On the one hand, the effect of an erroneously predicted action from an unseen state may be accumulated, leading to behaviors that are completely different from the demonstrations (*error cascading* Bagnell 2015). On the other hand, in tasks involving both humans and robots, it might be impossible to straightforwardly map human skills to the robot control interface, due to the significantly different embodiments of humans and robots (*correspondence problem* Nehaniv and Dautenhahn 2002).

One way to alleviate these challenges is to transfer task skills via a representation, e.g. in a shared task state space. Inverse optimal control (IOC) learns a scalar function which assigns low cost or high rewarding values to observed states and uses the function to steer the agent behavior in task

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10514-018-9757-y) contains supplementary material, which is available to authorized users.

✉ Hang Yin
hang.yin@tecnico.ulisboa.pt

Francisco S. Melo
fmelo@inesc-id.pt

Ana Paiva
ana.paiva@inesc-id.pt

Aude Billard
aude.billard@epfl.ch

[1] Learning Algorithms and Systems Laboratory, École Polytechnique Fédérale de Lausanne, CH 1015 Lausanne, Switzerland

[2] Intelligent Agents and Synthetic Characters Group, INESC-ID and IST, University of Lisbon, 1649-004 Lisbon, Portugal

Published online: 30 April 2018

reproduction. The scalar function can be a cost defined over instantaneous task states or a cost-to-go, which depends on the current state and the future ones by following a specific policy. The former one is regarded as a more succinct and dynamics-independent representation. However, learning a cost function requires disentangling the dynamical dependency and usually yields a harder problem. Learning a cost-to-go function is more tractable and straightforward to obtain a controller. It can also be used as an intermediate step to recover cost by constructing regression signals (Dvijotham and Todorov 2010). On the downside, a cost-to-go representation is more restrictive for task transfer due to its dependency on the underlying dynamics. To derive control in novel task configurations, one might need to build cost-to-go functions over a spectrum of dynamics and resort to adaptive estimation.

Learning either a cost or cost-to-go function, IOC-based frameworks work as unsupervised learning to reveal data patterns, thus are generally more computationally expensive than fitting a deterministic policy or dynamics. Demonstration instances are almost never identical due to "unconscious" motor noise. More importantly, "conscious" task preferences can lead to a systematic variation in performing the given task. Such a type of demonstration diversity results in data with multiple modes and may lead to sub-optimality during learning and execution. To this end, encapsulating different viable modes is important for task reproduction: it allows for robust and adaptive task execution in contexts where the optimal behavior might be infeasible. For this reason, and in order to accommodate for diverse demonstrations, it is necessary that the function space is rich enough to fully express rich, multi-mode demonstrations. Such complexity often leads to expensive iterations.

Existing IOC approaches often assume the task state is complete and fully observable. This poses challenges when the state is only partially observable and the latent information is necessary. Let us consider, for example, a handwriting task. In this task, monitoring the final path of the pen provides only a partial view of the dynamics of handwriting and its control. The control of handwriting requires careful balancing of forces along fingers as well as precise guidance of the end-point to generate legible letters. Moreover, there are other factors that are only partially observable but play an important role in the controller and its final output, such as the writer's style. Another example, which motivates the latent state from the perspective of industrial applications, is shown in Fig. 1. The robot assists in a collaborative task in which the human might have different preferred assembly steps. The human preference is hence a salient latent state for the robot to infer and adapt its behavior to supply desired pieces.

In this paper, we present an ensemble inverse optimal control framework that efficiently learns from human demonstrations that exhibit implicit "preferences" or "styles". The paper is by no means aiming to fully address general imitation learning challenges, e.g., the correspondence problem, but attempts to explore IOC to facilitate robot adaptation applications which entail efficient learning and inference. The idea is to leverage ensemble methods to learn simple cost-to-go representations and then aggregate them to yield a powerful model. The validity of the simple models is ensured by learning over a subset of consistent data. The high similarity of the data in such subset means that it can be labeled as a latent state that can be cast as the mode of the demonstrations. Such latent state is then used together with the learned ensemble model for online estimation and adaptation (Fig. 2). The work builds on our previous work Yin et al. (2016), including a substantial extension in terms of both theoretical and empirical contributions. Summarizing, the main contributions are:

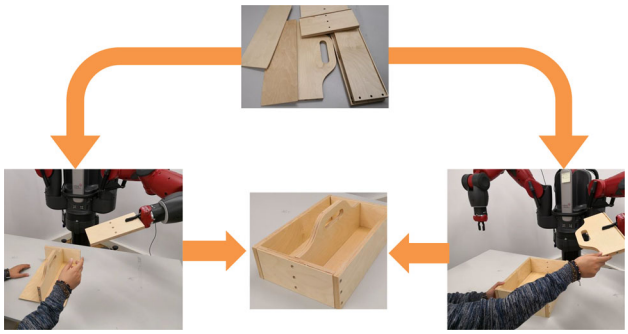


Fig. 1 A motivating example of tasks involving multiple execution modes or preferences: in a robot collaboration task, the box can be built by following different sequences, e.g., first to assemble the handle (left) or the frame (right). The robot needs to identify the preferred way in which collaborators do this task and adapt the actions accordingly

- An extension of Yin et al. (2016) from a maximum entropy (MaxEnt) assumption to the general linear-solvable system. Such extension makes the aforementioned work a special case of the framework proposed here.
- A new perspective on GMMs (Gaussian mixture models) in the context of IOC. Our results shed light on what GMMs actually learn (local MaxEnt models) and how can they be used as a guaranteed approximation.
- Integration of the task dynamics with the latent state to handle the challenge from incomplete state observation, for which a direct multi-mode policy encoding fails. The augmented dynamics provide a strategy to accommodate the disturbances or human intervention on-the-fly, by exploiting the task redundancy.

2 Related work

Our discussion of related work focuses on the task demonstration, inverse optimal control and learning human-robot collaboration.

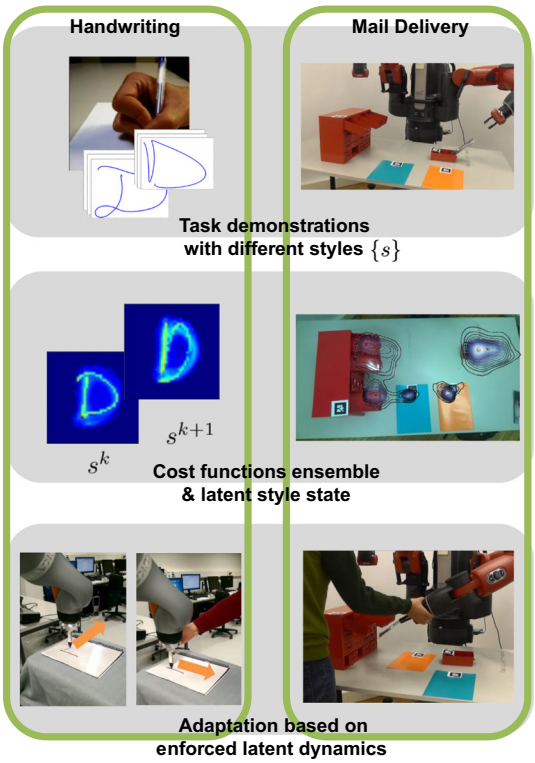


Fig. 2 Schematic illustration of learning cost-to-go ensemble representations for demonstrations with multiple modes. The extracted modes are cast as latent style states s in the enforced robot dynamics, which allows for task adaptations (moving direction in handwriting and reaching target in mail delivery) according to online interventions

Demonstrating task skills to robots: A typical way of demonstrating the desired task skill is kinesthetic teaching (Akgun et al. 2012; Khansari et al. 2014). From the correspondence perspective, in these works, the demonstration data and robot control share an identical space, hence a straightforward mapping is feasible. Another solution was adopted in the work Kukliski et al. (2014), where people used a data glove to remotely demonstrate the task via teleoperation. This is similar to the kinesthetic teaching in terms of the adopted control space.

Besides a direct mapping, other approaches choose to register the imitating trajectories into the robot control space, allowing for more natural demonstrations. For example, skill imitation was achieved by matching the demonstrated task trajectories and the ones derived from the robot control space (Englert et al. 2013). The matching was realized by minimizing the divergence between the trajectory distributions through reinforcement learning. Different from the above work (Englert et al. 2013), we use inverse optimal control (IOC) to cope with absence of control equivalence.

Inverse optimal control: The earliest IOC for the basic linear-quadratic system dates back to the pioneering work of Kalman Kalman (1964). In the last decade, nonlinear cost functions with a linear parameterization were learned with maximum margin planning (Ratliff et al. 2006), maximum entropy probabilistic models (Ziebart et al. 2008) and linearly-solvable systems (Dvijotham and Todorov 2010). These frameworks solve a convex optimization problem with a discrete-state description, which leads to a tractable evaluation of dual constraints or partition functions. For continuous states, the intractability of the sub-forward problems was tackled through discretization (Dvijotham and Todorov 2010), Laplacian approximation (Levine and Koltun 2012) or local sampling (Kalakrishnan et al. 2013). Also, learning non-parametric (Levine et al. 2011) and deep featured cost functions (Wulfmeier et al. 2015; Finn et al. 2016) were explored. Instead of attacking the complicated learning in one batch, we choose a divide-and-conquer strategy, namely decomposing and solving a bag of naive models which, when aggregated, leads to superior performance. The models are naive in that the assumed cost-to-go functions are of a simple quadratic form. The similar form is also adopted in Levine and Koltun (2012) and Monfort et al. (2015) for the trajectory optimization to approximate the partition function evaluation. Unlike Monfort et al. (2015), here the dynamics is assumed to be control-affine, which is more general and realistic than the maximum entropy assumption in aforementioned work. Levine and Koltun (2012) also formalizes the problem under a maximum entropy assumption. It exploits local trajectory optimization under a general deterministic dynamics to approximate the partition function evaluation. Our paper departs from a stochastic linearly-solvable system, which is comparatively restrictive but features a useful structure for efficient optimal control.

Learning for human-robot collaboration tasks: Applying imitation learning to a collaborative transportation task, Rozo et al. (2015) used the task parameterized GMM to encode and derive motion compliance that ensures safe interaction between robots and humans. The state reference was developed through model regression without explicitly considering the preferences of the collaborators. Another work Ewerton et al. (2015) proposed to use a robot to hand over different tools in a collaborative assembly task. Again GMM was used to model the cooperative behaviors in the primitive parameter space. The preference was assumed to be determined by the static waiting pose of the human hand. Our work differs from this by only assuming partial observability on the human intention. Here the intention is also assumed to be subject to a latent dynamics. In the work of Nikolaidis et al. (2015), the robot learned to cooperate a painting task by holding and adjusting the pose of a cube according to the preferred sequence of human collaborators. Similar to our work, it employed inverse reinforcement learning over classified demonstrations with different styles. Our approach is distinct by using the learned ensemble as the mode observational model, which was user-specific in Nikolaidis et al. (2015). Moreover, our system directly learns with continuous

demonstration data, while Nikolaidis et al. (2015) resorted to state discretization and the adopted mixed-observation Markov Decision Process is limited to low dimensional tasks.

Though not directly motivated in the topic of robot collaboration, a recent work (Abdolmaleki et al. 2016) proposes to contextualize policy search, which shares some technical similarity in that our work also determines behavior based on some context variable (task mode). Differently, our work is situated in the domain of imitation learning rather than policy search. Concretely, the task mode is implicit and dynamically inferred based on the demonstrated behaviors, while Abdolmaleki et al. (2016) assumes an explicit label of the context variable is available.

3 Implicit imitation learning of robotic tasks

3.1 Preliminaries

Given a robotic task to learn from humans, we assume expert demonstrations as a dataset $\mathcal{D} = \{\varsigma^i\}$ with i as the data index. Each trajectory ς^i corresponds to a sequence $\{\boldsymbol{x}^i_{0:T}, \boldsymbol{u}^i_{0:T-1}\}$, with the feature $\boldsymbol{x}^i_t$ denoting the task-relevant sensory information and $\boldsymbol{u}^i_t$ representing the recorded control input. For the example of a handwriting task, the demonstrated data could be a set of trajectories that form different styles of written letters in the Cartesian space, with the planar position coordinate and velocity as the features $\boldsymbol{x}^i_t$ and $\boldsymbol{u}^i_t$. We note that the subscript t refers to the phase index which, does not necessarily correspond to a "time" index. The demonstrated trajectories can be aligned by scaling the horizon to the same phase interval, e.g., from 0.0 to 1.0.

We do not assume that the demonstrated trajectories are necessarily similar, and admit they can exhibit different modes. These are indexed by variables $s^i \in \mathbf{N}$, but are not explicitly observed. Each mode indicates a particular way of executing the target motion, e.g., writing a letter with a specific style. We refer to such variables s^i interchangeably as "styles" or "modes" throughout the paper.

The human and robotic agents are constrained by their corresponding dynamical models, which can be different as long as the task-relevant perceptions are equivalent. Each of the models can be formulated as a Gaussian stochastic process with a nonlinear dynamical system plus an additive noise $\mathcal{N}(0, \boldsymbol{\Sigma}_0)$. Moreover, the deterministic part is assumed as a control-affine system with a constant input gain:

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t) + \boldsymbol{B}\boldsymbol{u}_t \tag{1}$$

where $\boldsymbol{B}$ represents the gain matrix of the control input. The term $f(\boldsymbol{x}_t)$ corresponds to the nonlinear dynamics which govern the state in the absence of the active control. These task-irrelevant dynamics are constructed with agent dependent $f(\cdot)$, $\boldsymbol{B}$ and $\boldsymbol{\Sigma}_0$, which are known or empirically determined.

Besides the agent dynamics, the states are also steered by task constraints, which are real-valued cost functions that are implicit in imitation learning. Specifically, when the state transition is Markovian, the observed demonstrations are assumed as (local) optima w.r.t. an accumulated cost-to-go function:

$$\mathcal{J}_\varsigma(\boldsymbol{x}_0, \boldsymbol{\theta}) = \sum_{t=t_0}^{T} C(\boldsymbol{x}_t, t, \boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{u}_t^T \boldsymbol{R}\boldsymbol{u}_t \tag{2}$$

where the cost is parameterized by an unknown parameter $\boldsymbol{\theta}$. Also, we further assume that (i) the control penalty weight $\boldsymbol{R}$ is known; and (ii) the covariance of the Gaussian noise is also known and inversely proportional to $\boldsymbol{R}$. Therefore, only the state dependent term must be determined for task learning. This formulation results in a special *linear-solvable system*, whose duality between optimal control and estimation can be exploited in general nonlinear learning and control problems (Kappen et al. 2012; Dvijotham and Todorov 2010). Concretely, according to Dvijotham and Todorov (2010), the probability of a controlled state or a finite-horizon trajectory could be parameterized by the cost-to-go function

$$P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \boldsymbol{\theta}) = \frac{P_0(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)e^{-\mathcal{J}_\varsigma(\boldsymbol{x}_{t+1}, \boldsymbol{\theta})}}{\int P_0(\boldsymbol{x}'_{t+1}|\boldsymbol{x}_t)e^{-\mathcal{J}_\varsigma(\boldsymbol{x}'_{t+1}, \boldsymbol{\theta})}d\boldsymbol{x}'_{t+1}} \tag{3}$$

where P_0 is the Gaussian denoting the agent passive dynamics without any active control, i.e., $\mathcal{N}(f(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_0)$. The demonstrated behavior can be learned by maximizing the data likelihood. By learning the cost-to-go function, the task can be represented and reproduced on an agent with a same or slightly perturbed dynamics.

For the forward control synthesis, under such a system setting and the learned cost-to-go function, the corresponding optimal controller could be derived by following Bellman optimality:

$$\boldsymbol{u}^*_t = -\boldsymbol{R}^{-1}\boldsymbol{B}\frac{\partial \mathcal{J}_{\varsigma^*}(\boldsymbol{x}_{t+1})}{\partial \boldsymbol{x}_{t+1}} \tag{4}$$

with $\boldsymbol{u}^*_t$ denoting the optimal control input. Hence the cost-to-go can reveal a applicable controller when the control is expected to be exercised on a homogeneous agent.

3.2 Challenges

The challenges are twofold. In terms of cost function learning, maximizing the demonstration likelihood (3) requires iteratively performing the partition function evaluation, which effectively solves a forward optimal control problem. On the other hand, for the empirical robot control, (4) is

still not nearly as tractable as a closed-form, even though the Bellman equation in the finite time horizon case is linear.

Fortunately, both problems can be efficiently solved when considering the linear-quadratic-regulator (LQR), namely:

$$\begin{aligned} f(\boldsymbol{x}_t, t) &= \boldsymbol{A}_t \boldsymbol{x}_t \\ C(\boldsymbol{x}_t, t) &= \frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{r}_t)^T \boldsymbol{Q}_t (\boldsymbol{x}_t - \boldsymbol{r}_t) \\ \mathcal{J}_\varsigma(\boldsymbol{x}_t) &= \frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Lambda}_t (\boldsymbol{x}_t - \boldsymbol{\mu}_t) \end{aligned} \tag{5}$$

where $\boldsymbol{A}_t$ denotes a linear state transformation. $C(\cdot)$ takes a quadratic form with $\boldsymbol{r}_t$ as the reference state and $\boldsymbol{Q}_t$ is a positive-definite (PD) weight matrix. Thus the sum-up of these instantaneous costs will yield another quadratic cost-to-go $\mathcal{J}_\varsigma$, with the remaining constant term ignored. $\boldsymbol{\Lambda}_{t+1}$ is the corresponding PD matrix which can be computed from the Riccati equation.[1] $\boldsymbol{\mu}_t$ denotes the reference state. Note it is a parameter different from $\boldsymbol{r}_t$ because it merges a feedforward term which appears in the LQR tracking problem. The optimal controller is given by:

$$\boldsymbol{u}_t^* = -(\boldsymbol{R} + \boldsymbol{B}^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{B})^{-1} \boldsymbol{B}^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{A}_t (\boldsymbol{x}_t - \boldsymbol{\mu}_t) \tag{6}$$

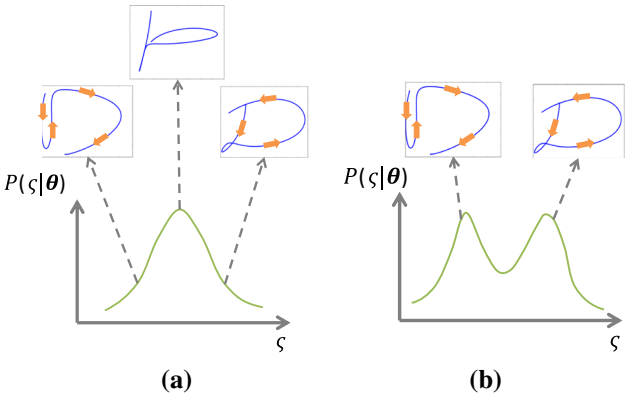
Such a system, however, is limited in that: (1) a quadratic cost is spatially local so it has to be estimated by locally consistent data; (2) temporally, the system assumes a unique trajectory mode whilst it is possible in practice to observe multiple modes or preferences in executing one specific task. As exemplified in Fig. 3a, b, there is a latent variable governing the task mode, which can not be fully decided by the current state and cost parameters.

To deal with above challenges, our approach is aimed at: (i) leveraging the efficiency of quadratic form to learn a global cost-to-go function as the task description; (ii) exploiting the cost-to-go function to infer the latent task mode for the reproduction and adaptation in control synthesis.

4 Ensemble inverse optimal control

This section develops an ensemble approach to address the two goals identified above. The motivation is that learning quadratic cost-to-go functions is arguably efficient when the data trajectories are locally consistent and of similar styles. The idea is, then, to obtain subsets of similar demonstrations to learn local quadratic cost-to-go functions. We expect that the construction to incur an acceptable extra computational load, if the subsets used can be efficiently constructed. We then build an ensemble that performs well by aggregating a set of such "weak" models.

[1] Namely, by recursively evaluating $\boldsymbol{\Lambda}_t = \boldsymbol{Q}_t + \boldsymbol{A}_t^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{A}_t - \boldsymbol{A}_t^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{B}_t (\boldsymbol{B}_t^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{B}_t + \boldsymbol{R}_t)^{-1} \boldsymbol{B}_t^T \boldsymbol{\Lambda}_{t+1} \boldsymbol{A}_t$ with $\boldsymbol{\Lambda}_T = \boldsymbol{Q}_T$.



(a) (b)

Fig. 3 Unique and multiple modes of demonstration trajectories to execute a task, with handwriting motion as an example. **a** Poor model to encapsulate the diversity and redundancy of styles in forming the letter "*D*". Actually, the unique mode, which approximately represents the mean trajectory, is not legible, and should be assigned with low probability (high cost value) instead. Also, the state itself (the point coordinate on the arc) is not sufficient to determine the next desired position. **a** Unimodal. **b** Multi-modal

4.1 Local IOC on subsets of similar style demonstrations

A collection of demonstrated trajectories is of similar style if they are closely distributed, quantitatively resulting in a low entropy probabilistic model in (3). From a control perspective, for both cost-to-go and cost with a quadratic form, the demonstrations are regarded as optimal w.r.t. a local linear-quadratic (LQ) system as (5). Such a system is always possible, as one can linearize and expand the nonlinear dynamics and the original cost with the given trajectory as the nominal reference. By exploiting this fact, a set of similar demonstrations, when factored as state pairs, can be modeled by setting $\mathcal{J}_\varsigma(\boldsymbol{x}_t, \boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Lambda}_t (\boldsymbol{x}_t - \boldsymbol{\mu}_t)$ in (3), yielding

$$\begin{aligned} P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) &= \frac{1}{\sqrt{|2\pi \boldsymbol{Z}|}} e^{-\frac{1}{2}(\boldsymbol{x}_{t+1}-\boldsymbol{y})^T \boldsymbol{Z}^{-1}(\boldsymbol{x}_{t+1}-\boldsymbol{y})}, \\ \boldsymbol{y} &= \boldsymbol{Z}[\boldsymbol{\Sigma}_0^{-1} f(\boldsymbol{x}_t) + \boldsymbol{\Lambda}_{t+1}\boldsymbol{\mu}_{t+1}], \\ \boldsymbol{Z} &= (\boldsymbol{\Sigma}_0^{-1} + \boldsymbol{\Lambda}_{t+1})^{-1}, \end{aligned} \tag{7}$$

where $\boldsymbol{\Sigma}_0$ is covariance of the Gaussian noise of the passive dynamics. $\boldsymbol{Z}$ is the covariance matrix, which depends on $\boldsymbol{\Lambda}_{t+1}$. Therefore, the likelihood in (3) can be written in an explicit way, thanks to the closed-form evaluation of the integral of the product of two Gaussian functions. Moreover, a maximum-entropy (MaxEnt) formulation ($\|\boldsymbol{\Sigma}_0\| \to \infty$) leads to a standard Gaussian distribution whose maximum likelihood estimation is trivial, given sufficient demonstra-

tion data. Given that $\boldsymbol{y}$ is dependent on $\boldsymbol{Z}$ in Eq. (7), an iterative optimization can be used. The MaxEnt result thus appears as a good starting point for the estimation of $\{\boldsymbol{\mu}_{t+1}, \boldsymbol{\Lambda}_{t+1}\}$. In fact, we have the following guarantee, regarding this estimation surrogate:

Proposition 1 *The optimal estimation of* $\{\boldsymbol{\mu}_t, \boldsymbol{\Lambda}_t\}$ *for a MaxEnt formalization ensures a lower bound of the original likelihood* (7) *and the gap depends on* $\boldsymbol{\Sigma}_0$. *In particular, the gap decreases as* $\|\boldsymbol{\Sigma}_0\| \to \infty$.

See "Appendix" for the proof. The above conclusion means we can estimate the cost-to-go efficiently for such a local Gaussian-like model through a MaxEnt approximation.

We take the estimation of cost-to-go functions as the local IOC problem because it is more efficient than learning a cost function (Dvijotham and Todorov 2010). Also, a local controller can be immediately derived from a cost-to-go function, as is shown by Eq. (4). We will demonstrate learning both time-independent and time-dependent cost-to-go functions for modeling time-invariant task (inverted pendulum example in Sect. 6.1) and finite-horizon trajectories (robot tasks in Sects. 6.2.1 and 6.2.2). It is known that, for first-exit problems, the cost-to-go function corresponds to the cost function in the Bellman equation:

$$C(\boldsymbol{x}_t) = \mathcal{J}(\boldsymbol{x}_t) + \log \int P_0(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) e^{-\mathcal{J}(\boldsymbol{x}_{t+1})} d\boldsymbol{x}_{t+1} \tag{8}$$

In Dvijotham and Todorov (2010), the relation is suggested to be used for the inference of the cost function. Here, however, this is not exploited and we focus on the development of learning cost-to-go functions.

4.2 Random subspace partitioning for similar demonstrations

The above derivation implies that learning a quadratic cost function is cheap when the demonstrations are locally consistent so that the LQ assumption may apply. This entails obtaining groups of similar trajectories. Furthermore, for certain tasks, the local segmentation and state variable themselves may be inadequate to describe the status of task execution. Taking Fig. 3b as an example, the stroke direction of forming the circle in writing the two types of "D" depends on the global trajectory profile instead of the local geometry. We tackle this problem by introducing a latent variable, which complements the observable variables but is not explicitly specified in the demonstrations. This variable can be understood as the global "style" of the demonstration, or the mode of the task dynamics. Therefore, we propose to partition the demonstrations to obtain the data for a local IOC problem, by considering both global and local similarities. To this end, we need to efficiently group demonstration trajectories and the corresponding state, before applying (7).

There are numerous applicable clustering techniques for this preprocessing. A simple and rapid method such as K-means is a possible option. However, as is demonstrated in Sect. 6, it does not work well in the cases where the similarity metric is nontrivial.

Here we adopt an equally simple yet effective approach to partition the original dataset. It works in an iterative manner by recursively dividing the dataset. Let us take the trajectory grouping as the example. Each iteration of the algorithm seeks to maximize the information gain from introducing a partition on the current dataset:

$$\Delta H(\mathcal{D}, \phi(\cdot)) = H(\mathcal{D}) - [H(\mathcal{D}^{\phi(\varsigma)\geq 0}) + H(\mathcal{D}^{\phi(\varsigma)<0})], \tag{9}$$

where H denotes the entropy of the data trajectories under a probabilistic model. $\mathcal{D}^{\phi(\varsigma)\geq 0}$ and $\mathcal{D}^{\phi(\varsigma)<0}$ are the partitioned subset based the criterion $\phi(\varsigma) = 0$. We choose the MaxEnt model with the quadratic parameterization as the probabilistic model to evaluate the entropy, by exploiting the simplicity of Gaussian entropy. ϕ defines the function to decide the membership of each demonstration. This function is often constrained with a simple form, allowing the decision boundary to be efficiently searched. Existing research (Criminisi et al. 2012) provides popular options to obtain decision boundaries with different levels of complexity. The optimization in searching ϕ can be further relaxed by randomly selecting the effective features and the candidate solutions, as is suggested in Geurts et al. (2006). In the present paper, we employ a naive option, having $\phi(\varsigma) = \varsigma_{t,l} - \eta$ where $\varsigma_{t,l}$ denotes the lth dimension of the tth state $\boldsymbol{x}_t$ in trajectory ς. η is the intercept to be decided together with t and l through the random search. This in fact explores in a family of axis-aligned decision boundaries in the temporal and spatial space of the trajectories.

The above process can be performed recursively to obtain K subsets, as is demonstrated in Fig. 4. The recursive process can be terminated when the dividing violates the constraints of the minimum number of demonstrations N_D^{min} in the subsets. The establishment of partitions is efficient and effective in grouping demonstrations with a similar style (low entropy distribution). Local cost-to-go function models can be learned based on the each subset of the demonstrations by following (7). The pseudocode for this recursive partitioning subroutine is given as Algorithm 1. The algorithm returns K subsets $\mathcal{D}_{k=1:K}$ taking as input the complete demonstration set $\mathcal{D}$. Further explanation about the other parameters will be given later.

Algorithm 1 RandomSubSpace - Partitioning dataset through feature bagging

Require: $\mathcal{D}$, N_x, N_D^{min}
Ensure: $\mathcal{D}_{k=1:K}$
$\mathcal{D}_{k=1:K} \leftarrow$ SPLIT($\mathcal{D}$, N_x, N_D^{min})
function SPLIT($\mathcal{D}_{in}$, N_x, N_D^{min})
$\{\varsigma_{t,l}^i\}_{i=1:N_x} \leftarrow$ RandomSelect(ς)
$j, \eta_j^* \leftarrow \underset{i,\eta_i}{\text{argmax}}\, \Delta H(\mathcal{D}_{in}, \{\varsigma_{t,l}^i\}, \eta_i)$
$\mathcal{D}_{in}^1, \mathcal{D}_{in}^2 \geqslant \mathcal{D}_{in}$ ▷ Split the dataset according to j and η_j^*
if $|\mathcal{D}_{in}^1| \geqslant N_D^{min}$ and $|\mathcal{D}_{in}^2| \geqslant N_D^{min}$ **then return** Concatenate(SPLIT($\mathcal{D}_{in}^1$), SPLIT($\mathcal{D}_{in}^2$))
else return $\mathcal{D}_{in}$ ▷ Discard this split
end if
end function

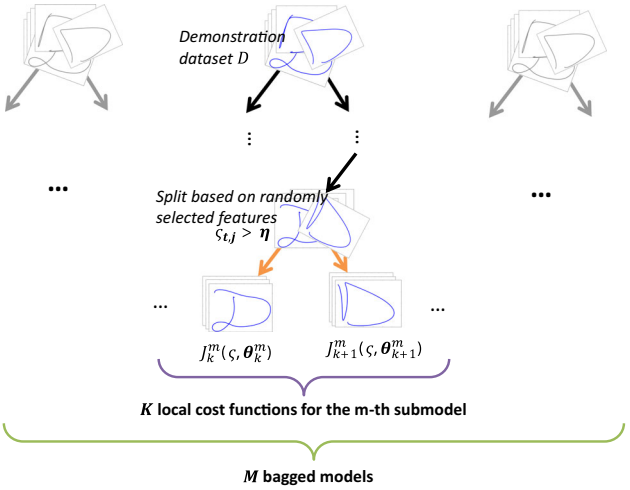


Fig. 4 An ensemble of cost-to-go functions over partitioned datasets through random feature bagging. The demonstrations are grouped according to suboptimal yet efficient decisions, resulting in trajectories with consistent styles so that a simple IOC model is plausible

4.3 Learning cost ensemble: analysis and algorithms

The resulting joint cost-to-go model can be the weighted aggregation of the local models estimated above. However, the estimation is unstable as the local learning depends on the results of data partitioning, which only considers the data correlation in a suboptimal way. The idea to mitigate such effects is to replicate the learning procedures for multiple times to build a model ensemble from a group of M models. Such bagging strategy is widely accepted and applied as a scheme to reduce model variance (Breiman 1996). There are multiple ways of integrating the model ensemble to estimate the unknown cost-to-go function. One viable option is to take a weighted log-sum over local models with a similar form as Todorov (2009):

$$\mathcal{J}^*(\boldsymbol{x}) \approx -\log \sum_{m=1}^{M} \sum_{k=1}^{K_m} w_k^m e^{-\mathcal{J}_k^m(\boldsymbol{x})} \tag{10}$$

$\mathcal{J}^*$ is the target cost-to-go that is approximated by the ensemble of quadratic $\{\mathcal{J}_k^m\}$, where the state trajectory ς was omitted and m indexes the instance of random partitions of demonstrations. $\{w_k^m\}$ denotes the weight of each *local model* (7). The weights can be defined as $\{w_k^m\} = \left\{\frac{card(\mathcal{D}_k^m)}{card(\mathcal{D})M}\right\}$, where $card(\cdot)$ denotes the cardinality of dataset.

The above ensemble strategy resembles a mixture of multiple simple probabilistic IOC models. The indices of $\{m, k\}$ can be understood as discrete latent variables, which loosely corresponds to trajectory styles s. It can be seen that the number of subsets is a partially controlled result from the random partitioning. Note that identical demonstration groups and thus local models might be obtained from different random partitions. This implies that a specific mode might be referred by multiple indices of $\{m, k\}$. The value prediction will not be impacted because the duplicated terms are weighted and merged in Eq. (10). Nevertheless, sometimes it might be more convenient to understand the latent modes as a fixed number of distinct clusters. The result of random partitioning is flexible to support this by enforcing this model prior. In fact, if the memberships of all subsets are jointly considered as a one-hot encoding of the data, the random partitioning embeds the original data into a manifold, yielding a high dimensional but sparse representation. Thus, the result of random subspace can also be used as random trees embedding (Geurts et al. 2006), which hashes the input features and constructs a non-Euclidean affinity matrix. Applying the affinity to standard techniques like K-means or spectral learning, the trajectories can be assigned into a given number of clusters with a nonlinear feature embedding.

With the cost-to-go functions learned, the control synthesis can be realized through standard backward passing or solving an invariant point problem. For instance, under a finite horizon LQR condition, Eq. (6) allows us to efficiently obtain the optimal control together with the local feedback gain. We give some additional remarks to discuss relations to other models:

– One way to explain the cost evaluation (10) is to see it as a *soft version of pointwise minimum* of a collection of cost-to-go functions. If such an evaluation is adopted, (4) yields:

$$\begin{aligned} \boldsymbol{u}_t^* &= -\boldsymbol{R}^{-1}\boldsymbol{B}\frac{\partial \mathcal{J}_{\varsigma^*}(\boldsymbol{x}_{t+1})}{\partial \boldsymbol{x}_{t+1}} \\ &= -\sum_{m,k}\left[\frac{w_k^m e^{-\mathcal{J}_k^m(\boldsymbol{x}_{t+1})}}{\sum_{m',k'} w_{k'}^{m'} e^{-\mathcal{J}_{k'}^{m'}(\boldsymbol{x}_{t+1})}}\boldsymbol{R}^{-1}\boldsymbol{B}\boldsymbol{A}_k^m(\boldsymbol{x}_{t+1}-\boldsymbol{\mu}_k^m)\right] \end{aligned} \tag{11}$$

Thus the control can be explained as a combination of state dependent local impedance controllers, which are analogous to the ones proposed in Khansari et al. (2014). Note that, the robot experiments in this paper will adopt

another type of control based on the most probable cost-to-go model.
- As another way, the local cost-to-go models can also be considered as the encoding of different potential action modes that are applicable to the task. If the model weights $\{w_k^m\}$ can be adaptively estimated, the most plausible mode can be inferred with certain decision-making mechanisms. This observation offers the possibility of trajectory adaptation in face of unmodeled disturbances.
- GMM can be cast as a special case of the ensemble with a MaxEnt assumption (7). Hence our framework answers the question of how GMM can be interpreted from a perspective of inverse optimal control. Actually, the framework extends the standard GMM by enforcing the passive dynamics, which is arguably important for physical plausibility (Dvijotham and Todorov 2010). Conversely, the connection to GMM implies a possible model parameter refinement through the expectation-maximization iteration while it is not formally explored in this paper.

The complete learning algorithm is presented as Algorithm 2 and an implementation is publicly accessible (https://github.com/epfl-lasa/ensemble_ioc). The algorithm receives demonstrations and parameters for both global trajectory clustering and local state partitioning. The partitions are used to obtain an approximated MaxEnt estimation of parameters $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Lambda}}$, as well as the partition weights w_k^m. The parameterized model Eq. (7) can then be used to evaluate the data membership to each local model:

$$\mathbb{I}_k^m(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t) = \frac{w_k^m P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \hat{\boldsymbol{\mu}}_k^m, \hat{\boldsymbol{\Lambda}}_k^m)}{\sum_{k'=1}^{K_m} w_{k'}^m P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \hat{\boldsymbol{\mu}}_{k'}^m, \hat{\boldsymbol{\Lambda}}_{k'}^m)} \tag{12}$$

The new parameters for each local model can then be obtained by solving the MaxEnt relaxation of Eq. 7, with $\mathbb{I}(\cdot)$ as the data weight.

A few arguments are open to trade-off the modeling power and the computational overhead. M_ς and M_x denote the number of aggregated models in the ensemble. Like other randomized methods, the performance of model ensemble improves monotonically as they grow (Breiman 1996). N_ς and N_x define the number of features that are involved to decide a split (see Algorithm 1). N_D^{min} specifies the minimum size of a set for the next split. These arguments can be adjusted to control the model complexity. A practical way of choosing N_ς or N_x is to take the square of the feature dimension (Geurts et al. 2006). Smaller N_D^{min} leads to finer partitioning, which implies reduced bias but increased variance and computational cost.

Algorithm 2 Learning - Learning cost-to-go ensembles from demonstrations

Require: $\mathcal{D} = \{\varsigma^i\}$, M_ς, M_x, N_ς, N_x, N_D^{min}, M(optional)
Ensure: $\mathcal{D}_{m=1:M}$, $\boldsymbol{\theta}_k^m$, $k = 1, ..., K_m$, $m = 1, ..., M$
$\mathcal{D}_{m=1:M} \leftarrow$ RandomSubSpace($\mathcal{D}, N_\varsigma, N_D^{min}$) with M_ς model ensemble
for all m in 1:M **do**
  $\mathcal{D}^x \leftarrow$ StatePairs($\mathcal{D}_m$)
  $\mathcal{D}^x_{k=1:K_m} \leftarrow$ RandomSubSpace($\mathcal{D}^x, N_x, N_D^{min}$) with M_x model ensemble
  for all k in 1:K_m **do**
    $\hat{\mu}_k^m, \hat{\Lambda}_k^m \leftarrow \underset{\boldsymbol{\theta}}{\text{argmax}} \sum_{i=1}^{|\mathcal{D}_k^x|} \log P_{MaxEnt}(\boldsymbol{x}^i|\boldsymbol{\theta})$
    $w_k^m \leftarrow \frac{|\mathcal{D}_k^m|}{|\mathcal{D}|}$
  end for
  for all $\{\boldsymbol{x}_{t+1}, \boldsymbol{x}_t\}$ in $\mathcal{D}^x$ **do**
    $\hat{\mathbb{I}}_k^m(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t) \leftarrow w_k^m P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, \mu_k^m, \Lambda_k^m)$ ▷ Membership of data to each partition under the MaxEnt approximation
  end for
  $\mathbb{I}_k^m(\boldsymbol{x}_{t+1}, \boldsymbol{x}_t) \leftarrow$ Normalize($\hat{\mathbb{I}}_k^m(\boldsymbol{x}_{t+1}), \boldsymbol{x}_t$)
  for all k in 1:K_m **do**
    $\mu_k^m, \Lambda_k^m \leftarrow \underset{\mu_k^m, \Lambda_k^m}{\text{argmax}} \sum_{i=1}^{|\mathcal{D}^x|} \mathbb{I}_k^m(\boldsymbol{x}_{t+1}^i, \boldsymbol{x}_t^i) \log P_{MaxEnt}(\boldsymbol{x}^i|\boldsymbol{\theta})$ ▷ Approximately solving (7) with the data weight $\mathbb{I}(\cdot)$
  end for
end for
$\boldsymbol{\theta}_k^m \leftarrow \{\frac{w_k^m}{M}, \mu_k^m, \Lambda_k^m\}$

5 Learning-based mode inference and adaptation

Equations (6) and (11) define deterministic controllers. Concretely, Eq. (6) assumes a fixed and known task mode s while Eq. (11) integrates out the mixture of s, assuming the control can be fully determined from the instantaneous state $\boldsymbol{x}$. However, these are not necessarily sufficient for all kinds of tasks. If we expect a potential change of the mode/latent state during execution, e.g., when the human operator changes his/her intention based on the exercised trajectory snippet, this variable should be dynamically inferred and conditioned to decide the control and adaptation.

Let us consider a toy task, where the robot end-effector is perturbed when writing a letter with a certain mode. The benefit of online mode adaptation is exemplified in Fig. 5. We note that a spring-like local feedback control, which always rejects the perturbations, would undermine the legibility of the letter. We argue that if the deviation can instead be considered as an alteration of task mode, which takes the motion history into account, the perturbation can be exploited to write the letter with another plausible style.

To achieve such adaptation mechanism, it is necessary for the robot to track the likelihood of each feasible demonstration mode during execution. The learned cost-to-go functions associated with each mode serve as natural measures. We

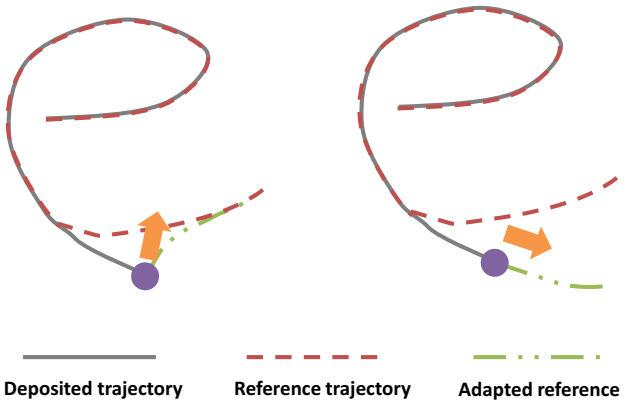


Fig. 5 Accommodating perturbation through trajectory tracking or adjusting the reference to another mode. Local feedback control (left) is inadequate while adapting the reference (right) to a redundant style is desired to retain the letter legibility

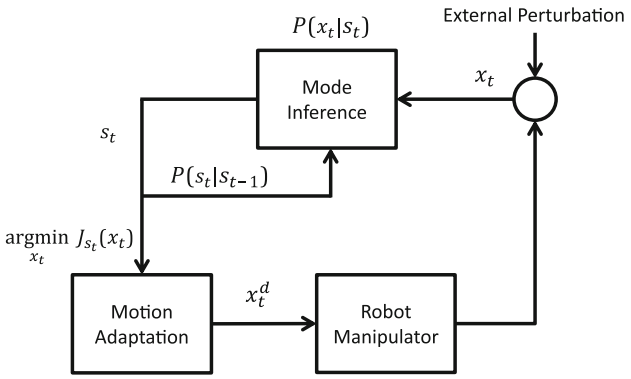


Fig. 6 Pipelines of mode estimation and control synthesis based on learned cost-to-go functions ensemble

also introduce a prior to the estimation: the latent task mode passively evolves as a Markovian process.

The goal of the prior is twofold. On the one hand, it biases the estimation process to ensure a more robust inference, because in practice the state measurement inevitably suffers from sensory noises. On the other hand, the temporally propagated prior provides a compact way to accommodate global trajectory information, which is necessary if the mode is not fully determined by instantaneous state measurements.

The pipelines of mode estimation and control synthesis are schematically depicted in Fig. 6. We denote the (unknown) state as $\boldsymbol{s} = [s^1, s^2, \ldots, s^M]$. $\boldsymbol{s}$ is an M-dimensional vector representing the belief over all possible modes and the ith entry is the likelihood of mode i. The evolution of the belief is modeled with a transition matrix $\boldsymbol{T}$, whose entry $\boldsymbol{T}_{ij}$ characterizes a prior possibility of switching from mode i to mode j. The learned cost-to-go functions provide evidence, evaluating the expected cost of all possible modes at the current state. Concretely, after observing $\boldsymbol{x}_{t+1}$, the mode belief $\boldsymbol{s}_{t+1}$, can be recursively inferred as:

$$\boldsymbol{s}_{t+1}(\boldsymbol{s}_t, \boldsymbol{x}_{t+1}) \propto (\boldsymbol{T}\boldsymbol{s}_t) \odot \begin{bmatrix} e^{-\mathcal{J}_1(\boldsymbol{x}_{t+1})} \\ \cdots \\ e^{-\mathcal{J}_i(\boldsymbol{x}_{t+1})} \\ \cdots \\ e^{-\mathcal{J}_M(\boldsymbol{x}_{t+1})} \end{bmatrix} \tag{13}$$

where $\odot$ denotes an element-wise product.

It is easy to find that such a recursive inference works as Kalman filtering. From this perspective, the motion mode or style is the latent state to be identified and the cost-to-go functions can be viewed as observational models. Also, the latent dynamics $\boldsymbol{T}$ can be estimated by counting the occurrences of mode transition given the observation model and data. This appears similar to learning an HMM-like model, though here the emission probability is separately learned and the distribution is nontrivial comparing with a categorical or a Gaussian one in HMM. In this work, we choose to obtain $\boldsymbol{T}$ in an ad-hoc manner. The reason is that here the latent state is understood as the trajectory mode, which is ideally invariant throughout each expert demonstration. This is conceptually different from most HMMs, whose latent state behaves like the label of a trajectory section. More importantly, $\boldsymbol{T}$ offers an intuitive way for users to shape the expected behavior, which requires a trade-off between the robustness against disturbances and responsiveness of mode adaptation.

Specifically, when $\boldsymbol{T}$ is given as a uniform transition, the responsiveness to mode adaptation is maximized, while robustness may be compromised. The reason is that the system will immediately adopt the new mode as long as its current state appears to be more likely w.r.t. the corresponding cost-to-go function. Moreover, this special case is similar to following a multi-mode policy, which adapts by only considering the immediate state. On the other hand, a diagonally dominant $\boldsymbol{T}$ tends to assume an invariant the mode, unless the cost-to-go functions provide strong evidence that another mode is more plausible. In the extreme case where the diagonal entries are Dirac functions, the system will reject any attempt of eliciting an adaptation to other modes, resulting in a maximized robustness.

6 Implementation and evaluation

The presented framework is implemented and evaluated in both simulated experiments and robotic applications. We begin the evaluation with an illustrative example, where we estimate the cost-to-go function for an inverted pendulum system. This low-dimensional time-invariant case offers an intuitive way to visualize the results and showcase characteristics of the algorithm. We also discuss the computational efficiency of our approach when compared to other IOC

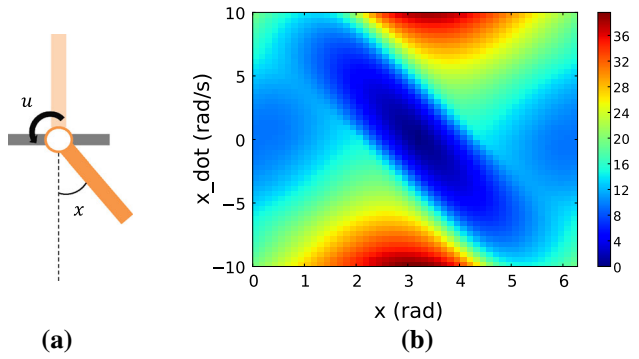


Fig. 7 An illustrative example: inverted pendulum regulation and the optimal cost-to-go function. **a** Inverted Pendulum. **b** Target cost-to-go

approaches in this task. As second example scenario, we illustrate the application of our approach in teaching a robot to produce human-like handwriting. In this example, we emphasize the ability of the system to extract different styles of handwriting in addition to learning the cost-to-go task representation. Finally, in a third example we illustrate the proposed cost-function-based adaptation scheme to generate adaptive robotic motion in a mail delivery task under a human collaborative intervention. The validations on real robots are illustrated in the supplemental video.

6.1 Inverted pendulum: an illustrative example

We consider the task of controlling an inverted pendulum. The goal is to select the input torque u so as to let the pendulum stay upright (Fig. 7a). The system has typical second-order dynamics, with one degree-of-freedom (DOF) and nonlinear passive dynamics. Thus the cost-to-go function is of a nontrivial form while simple enough for visualization.

The system parameters for the test are: pendulum mass $m = 1.0$ kg; length $l = 0.5$ m; joint damping $b = 0.1$ N m/(rad/s); gravity coefficient $g = 9.81$ kg m/s^2. The state comprises the angular position x and its derivative $\dot{x}$. A quadratic instantaneous cost function encoding the goal of control could be

$$C_{pend}(x) = \frac{1}{2}(x - \pi)^2 \tag{14}$$

where π denotes the target angular position in radians, indicating the upright configuration here. The optimal cost-to-go function can be derived through system discretization and standard value iteration. We set a constraint that saturates the control input with $u \in [-5.0, 5.0]$. The heat map of the underlying optimal cost-to-go is shown as Fig. 7b.

A total of 200 motion trajectories of 100 steps each, steered by the optimal cost-to-go function, are generated as demonstrations. Of these, 150 are used for sampling state-control pairs. The training dataset is corrupted by an additive noise

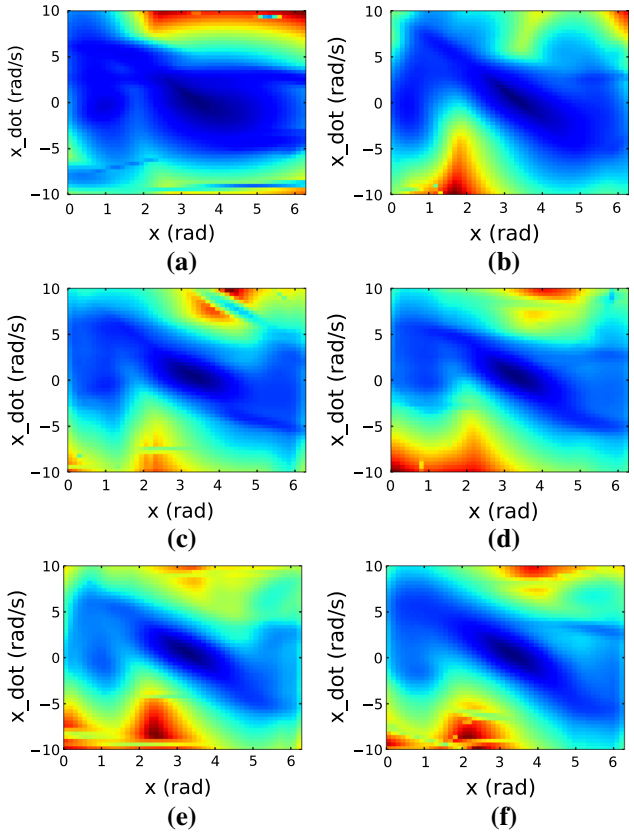


Fig. 8 Cost map of inverted pendulum states with the number of ensemble models $M = \{5, 10, 20, 30, 50, 75\}$. The legend of heatmap can be found in Fig. 7b. **a** $M = 5$. **b** $M = 10$. **c** $M = 20$. **d** $M = 30$. **e** $M = 50$. **f** $M = 75$

with a standard deviation of 0.02 to simulate the sensory noise. The task for the proposed ensemble method is to determine the time invariant cost-to-go function from the demonstrations, assuming the passive dynamics $p_0(x'|x)$ are known. Also, the angular position is truncated to $[0, 2\pi]$ to ensure the Euclidean distance is properly defined, though such approximation does bias the outcome due to the bound effect. It is worth noting that the inverse problem is addressed in continuous state and control space without discretization, though the data is generated from the standard value iteration of the discretized system.

We first examine the effects of the number of aggregated models. The learning results are depicted throughout Fig. 8a, f. Comparing with the target (Fig. 7b), it can be observed that as more models are incorporated, the learning performance improves in terms of visual consistency. The observation demonstrates the anticipated advantage of model ensemble: each of the sub-models is limited due to its high sensitivity and dependence on the data partitioning (Fig. 8a, b), while a prediction from the aggregated models leads to a better estimation than any individual model, with the overall variance significantly reduced.

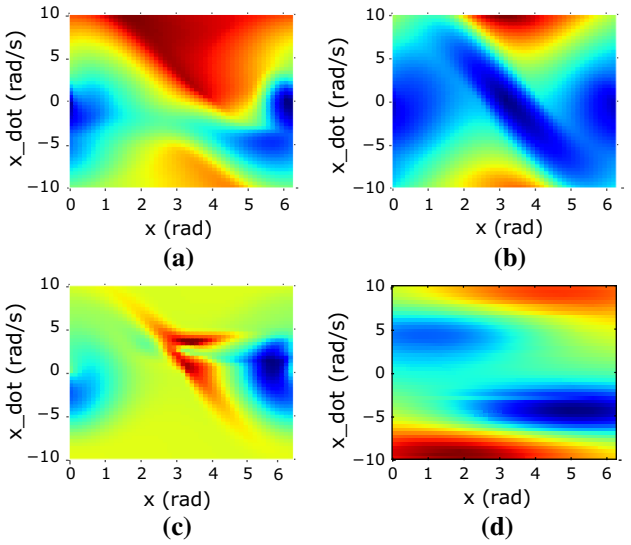


Fig. 9 Estimated cost-to-go functions from the MaxEnt (linear combination of RBF or quadratic functions), GPIRL and OptV results. An additional value iteration is performed for MaxEnt and GPIRL to visualize the cost-to-go function over the state space. OptV uses RBFs for the cost-to-go function approximation. 25 basis functions are used for all of the RBF-based approaches. The legend can be found in Fig. 7b. **a** MaxEnt+RBF. **b** MaxEnt+QF. **c** GPIRL. **d** OptV

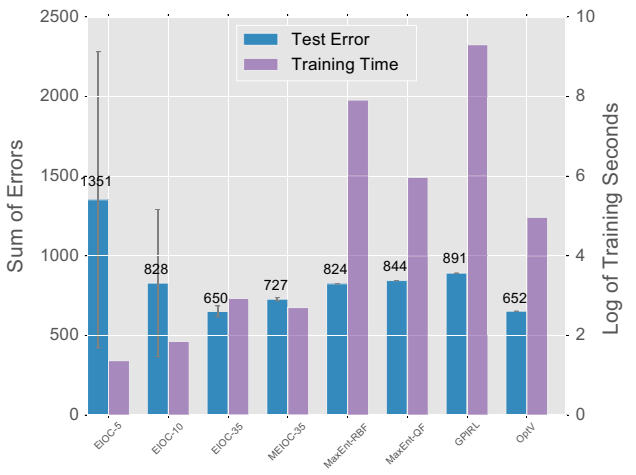


Fig. 10 Cost-to-go function errors and training time of different approaches for the inverted pendulum problem. The proposed approach is tested by integrating different number of models in the ensemble. The MEIOC indicates the application of the approach without considering the passive dynamics (MaxEnt formulation). Note the training time is transformed to its logarithm for the visualization

For a comparison, we also applied other approaches (MaxEnt+Laplacian Levine and Koltun 2012, GPIRL Levine et al. 2011 and OptV Dvijotham and Todorov 2010) to this inverted pendulum task. We are interested in comparing the performance of these approaches on this benchmark problem, both in terms of reconstructed cost-to-go function as well as the training efficiency. All approaches use 64 demonstration trajectories and retrieve the estimated state value of 2600 test state samples. The reconstruction error is obtained as the sum of errors between the estimated value and the ground truth, both of which deduct lowest values to assure comparable cost evaluations. For algorithms that estimate a cost function (MaxEnt+Laplacian and GPIRL), we compute cost-to-go functions based on the inferred cost function. The computation time for this additional step is not included for a fair comparison of the efficiency of original learning algorithms.

The estimated cost-to-go functions from these approaches are depicted in Fig. 9a–d. Apparently, one of the MaxEnt setting (Fig. 9b) shows the best qualitative results. This is expected because it learns a quadratic cost function which is consistent to the real goal. For more general cost parameterizations, such as RBFs (Fig. 9a, d) and Gaussian process (9c), the recovered cost-to-go functions show some similar local geometry in certain regions but fail to capture the overall landscape comparing with Figs. 9b and 8f. Quantitatively, in Fig. 10, we see a trend similar to what observed in terms of the cost-to-go function estimation: that reconstruction error of the ensemble method steadily decreases as more models are included. Regarding the training time, it is notable that the ensemble method is superior in terms of training speed thanks to the efficiency of learning naive local models. For the sake of comparison, we also present the result of a MaxEnt version of our method, which effectively works as a GMM over the demonstration state. It is not surprising to find a slight decrease in performance (in terms of sum-of-errors) since the MEIOC is ignorant about the real passive dynamics model. The results for other algorithms are mixed because the visually best result (Fig. 9b) does not lead to a smallest prediction error of the cost-to-go function values. This implies that the learning performance cannot be fully described by one metric and other dimensions need to be examined.

To have a more thorough conclusion, we also take the policy perspective, seeing whether the learned cost-to-go function indeed leads to behaviors that match the demonstrations. Two experiments are included with the first one focusing on the difference between the derived and demonstration trajectories, and the second one evaluating the trajectory performance under the real task cost function. Predicting the next state under the optimal policy requires a maximum posterior estimation in (3). This boils down to a nonlinear optimization, for which we use the MaxEnt mean estimation as the prior guess to ensure the optimization performance and efficiency. The initial states of 10 test trajectories are exposed to the algorithms, seeding a recursive prediction of states or a trajectory optimization for the same number of steps to compare against the ground truth.

For the first experiment, the derived trajectories are visualized in Fig. 11, where the stars denote the terminal states. It is clear that the predicted trajectories generally follow the demonstrated behavior. A quantitative result is given in

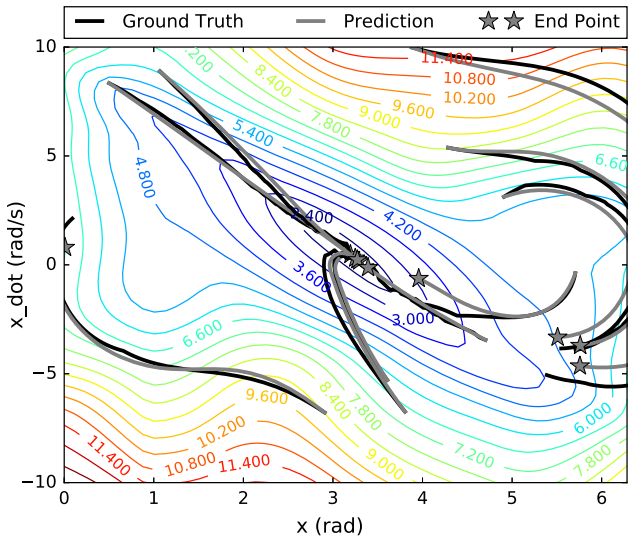


Fig. 11 Prediction of the trajectory under the learned cost-to-go ensemble and the ground-truth: the predicted trajectory is derived given the test initial state. The learned cost-to-go, which encodes the desirability of future state, is illustrated as the contour lines

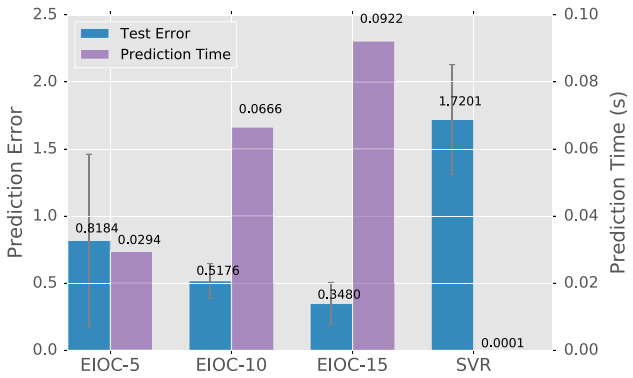


Fig. 12 Comparing different settings with a SVR-based prediction. The regression of behavior cloning is fast for each iteration of the prediction but suffers from error cascading along the trajectory horizon

Fig. 12, where a support vector regressor (SVR) is trained as a baseline. The SVR-based prediction works as behavior cloning by predicting the next state given the current one so it is very efficient for the synthesis. Unfortunately, the accuracy of overall trajectory prediction is poor, due to the error cascading effect. The IOC-based prediction is more reliable, thanks to the bias about the future from the extracted cost-to-go. Again, the model aggregation improves the performance, while in exchange, it takes longer time to conduct the optimization when more models are integrated.

In addition to a reproduction starting from test states, the developed control is also applied to task settings with different joint damping parameters. The desired state $\boldsymbol{x}_{t+1}$ is predicted from a bag of dynamics and expected to alleviate the trajectory shift, yielding a control with certain robustness to the discrepancy of dynamics. Table 1 reports the difference

Table 1 Trajectory error of applying developed control to dynamics with perturbed joint damping values

Damping b	EIOC-15	SVR
0.01	0.6866 ± 0.2556	1.8162 ± 0.5114
0.02	0.6644 ± 0.2388	1.8161 ± 0.5113
0.05	0.6112 ± 0.2011	1.8160 ± 0.5111
0.2	0.5703 ± 0.1308	1.8156 ± 0.5095
0.5	0.8829 ± 0.2481	1.8156 ± 0.5067
1.0	1.4477 ± 0.5833	1.8179 ± 0.5001

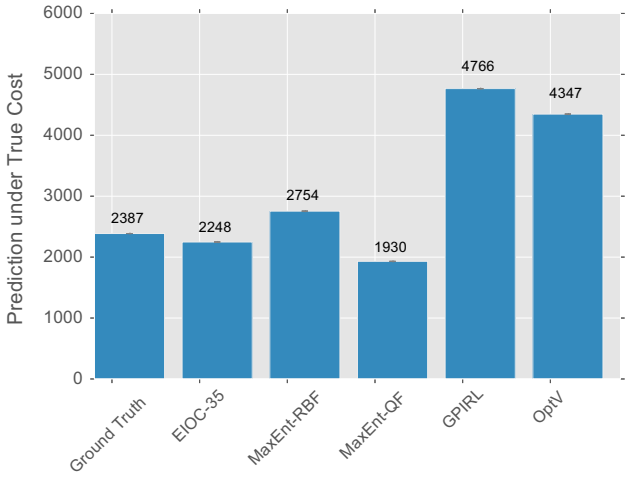


Fig. 13 The performance of the predicted trajectories under the true cost function: comparing test trajectories and the results obtained from ensemble method, MaxEnt, GPIRL and OptV

between the optimal trajectory and the realized one under the perturbed dynamics. The performance deteriorates as the damping is moving away from the nominal value ($b = 0.1$). It works well for most conditions and consistently outperforms the baseline control.

The result of the second numerical experiment is shown as Fig. 13. Specifically, the accumulated trajectory costs are evaluated under the true cost function. The proposed ensemble approach outperforms all the other algorithms on this metric, except the MaxEnt approach with the true quadratic feature. Note that both of these two approaches achieve better performance comparing with the test trajectories themselves. This is because the test trajectories are obtained from a more limited action set due to the discretization, while the IOC algorithms use continuous optimization to derive trajectories under the learned cost or cost-to-go functions.

6.2 Cost-function-based robotic motion adaptation

This section presents the application of the developed framework to online robotic motion adaptation. In a handwriting example, we show how the motion mode evolves and adapts

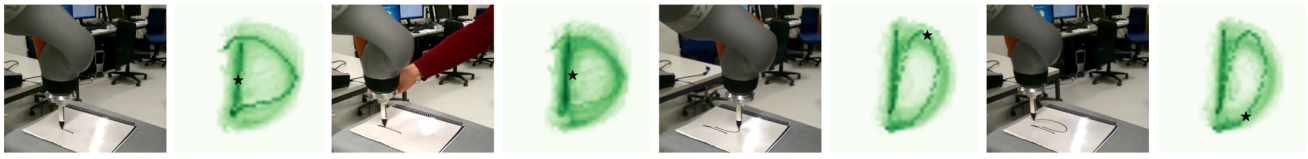


Fig. 14 Adapting the motion of writing a "D" on a KUKA IIWA 7-DOFs manipulator. The lightness of the reference trajectories indicates the associated mode weights and the star marks the current regulating point. Under the human intervention, the task mode shifts to the alternative modes that are plausible w.r.t. the deposited trajectory and future cost. The online adapted writing motion yields a different letter profile comparing with the original intention

by considering the external perturbation as well as the executed trajectory. The behavior is further validated in a more general mail delivery task, where the observable state is more complex and augments the feature descriptions concerning different reference frames.

6.2.1 Handwriting adaptation

The goal of this task is to encode multiple writing styles with the ensemble model, and to use the resulting model to adapt the letter style as a consequence to a human's intervention. In order to realize this, the robot needs to acquire redundant ways of writing the target letter and exploit this knowledge to assess and modulate the task execution.

We applied the framework to the demonstrations of a set of 120 planar trajectories forming a letter "D", naturally written by 60 different people. The ensemble parameters were set to allow a maximum of 240 local models as we are not certain about how many styles are there in the demonstrations. Learning from such a diversified dataset demonstrates the framework can successfully capture the inherent writing styles from motion variabilities. This is shown in our conference paper Yin et al. (2016). Here we show an implementation on a real robot that utilizes the extracted styles for adaptation, addressing the motivating example in Sect. 5. The robot, a 7-DOFs KUKA IIWA manipulator, is used to follow the commanded trajectory, which is initially sampled from the learned model ensemble.

Figure 14 showcases the expected behavior. Specifically, the robot follows the initial mode that deposits a downward stroke at first, and plans to finish writing on the top of the canvas (Fig. 14b). Then a human subject intervenes, making the compliant robot motion yield to moving upwards instead of following the planned direction. As a result, the perturbation elicits the need of an alternative task execution modes, as seen in the mixture of letter profiles in Fig. 14d. These modes are regarded as more probable ones according to the task costs, which jointly consider the history (the downward stroke) and the probable future motion styles. The mode estimation proceeds with the shifted mode reinforced and finally resembles an adapted written letter, which retains the legibility under the perturbation (Fig. 14e, g).

We emphasize the fact that the evolution of mode estimation serves as a compact dynamical encoding of the latent letter style, which may change subject to the human intervention. This is necessary as the position state itself is not sufficient to determine the motion, because the velocities might be conflicting at a same position for different writing styles. Here, the instantaneous position helps to decide which trajectory mode will cost less if we depart from the current state. Therefore, the learned cost-to-go representation enables the robot to evaluate, comply and, as such, *exploit* a perturbation when there exist potential modes that turn out to be suitable with the future steps taken into account.

6.2.2 Mail delivery task

This section presents the application of the proposed framework to a mail delivery task, where a robot assists in picking, transporting and delivering mail to different target mailboxes (Fig. 15). In this task, the mail messages are supposed to go via specific locations in the workspace (marked by col-

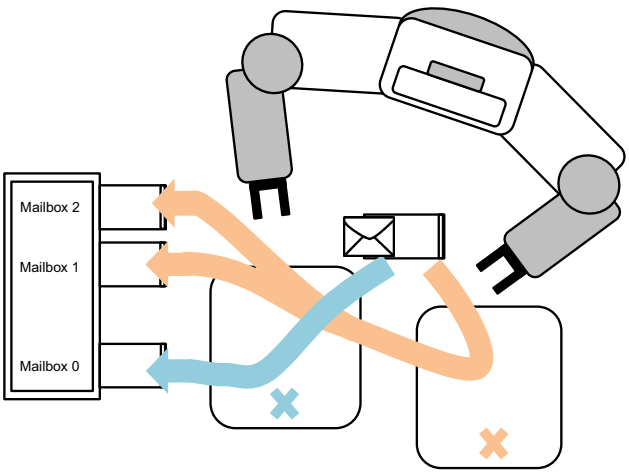


Fig. 15 Assisting in a mail delivery task. The robot needs to learn multi-mode behavior that manipulates the mail to different target boxes. The validity of the targets depends on which path was taken in the intermediate step

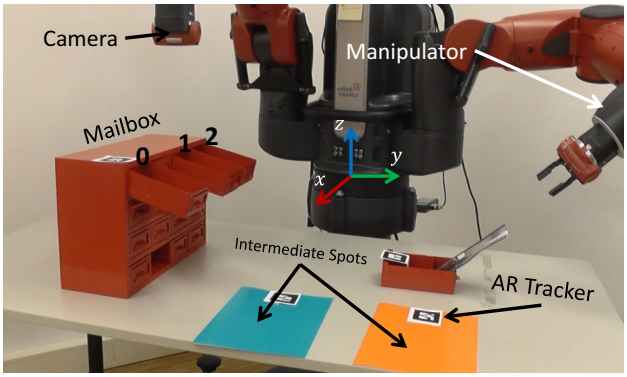


Fig. 16 Setup for the mail delivery task: the candidate objects/frames (mailbox, cyan/orange regions and mail location) are labeled by AR trackers, which can in turn be detected by a mono-camera at the right wrist of Baxter. The left arm is used for manipulation

ored crosses in the figure), for a hypothetical intermediate processing—such as stamping or labeling mails with different priorities. The delivery target depends on the spots by which the mail has passed. Moreover, during the execution, humans may intervene through a physical interaction. The robot, on the other hand, should decide if it will adapt its motion to collaborate the human intervention, or insist its current motion plan.

6.2.2.1 Experimental setup

The task is carried out on a Baxter robot platform, with the setup illustrated in Fig. 16. The AR trackers are used to label the reference frames that might be relevant to specific task modes. The poses of these frames are estimated through a camera to retrieve the current task configuration. 12 demonstrations are recorded through kinesthetic teaching, with four replications for each mode. Three task modes correspond to motion trajectories via different landmarks:

- {mail location, cyan area, mailbox-0};
- {mail location, orange area, mailbox-1};
- {mail location, orange area, mailbox-2}.

Note that the constraints of the sequence modes, e.g., which area should pass and then which mailbox to deliver to, are unknown to the robot. Humans can only convey them through demonstrations. For each demonstration, the location of the scene objects are rearranged, but the aforementioned sequences are always followed. The recorded states have a dimensionality of 18, with the position in each reference frame and the time index included. The trajectories are clustered with a random embedding from 1, 000 ensemble trees. For each trajectory mode, an ensemble of 10 models under a finite horizon formulation are trained, and the resulting models are used to infer the task mode and derive the command for the next step. We use

$$\boldsymbol{T} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 0.8 & 0.2 \\ 0.0 & 0.2 & 0.8 \end{bmatrix}$$

as the latent transition dynamics through all the experiments except those involving baseline methods. Such latent dynamics represent the prior knowledge that the motion mode tends to keep constant, although there is a moderate possibility to switch between mode 1 and mode 2.

6.2.2.2 Task goal and task-parameterized feature

The learning goal of this task is to encode constraints regarding both the static environment configuration and the process dynamics. On one hand, the robot needs to extract important task-relevant landmarks in order to adapt its movement to cope with a general environment configuration (e.g., unseen mailbox position and intermediate via-points). On the other hand, constraints about the task dynamics also need to be conveyed in the form of cost-to-go function learning. It is critical for the robot to exploit this knowledge to evaluate and react to the deviations, which can source from the motor noise or human intervention. The robot should resist the deviation when it is due to the motion noise or a human intervention that violates the task constraints, while adapt to human intended motion when it is compatible to the task constraints. Notably, here the constraints stem from the trajectory history—namely, which via point has been passed through. This implies that the adaptation cannot be exercised based on static or time invariant observations.

In order to generalize to different static configurations, we incorporate a feature representation similar to the task-parameterized models (TPGMM) (Calinon et al. 2014). Specifically, a task parametrization augments the interested state with representations in different reference frames of the task scenario. For instance, as illustrated in Fig. 17, the interested robot end-effector pose could be represented in different reference frames, such as A, B and C in the scene. The final state is the augmentation of these local descriptions thus is of a higher dimension than the original pose. A task-parameterized feature encapsulates the information relative to landmarks that are potentially important to the task execution, as such supports the generalization under an unseen arrangement of the landmark configuration. Specifically, we learn a varying quadratic cost-to-go function over this representation:

$$J(\boldsymbol{x}_t, \boldsymbol{\theta}_t) = \frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_t)^T \boldsymbol{\Lambda}_t (\boldsymbol{x}_t - \boldsymbol{\mu}_t) \tag{15}$$

with $\boldsymbol{x}_t$ denoting the concatenate state similar to Fig. 17. Note here $\boldsymbol{\Lambda}_t$ is block diagonal to factorize the cost with respect

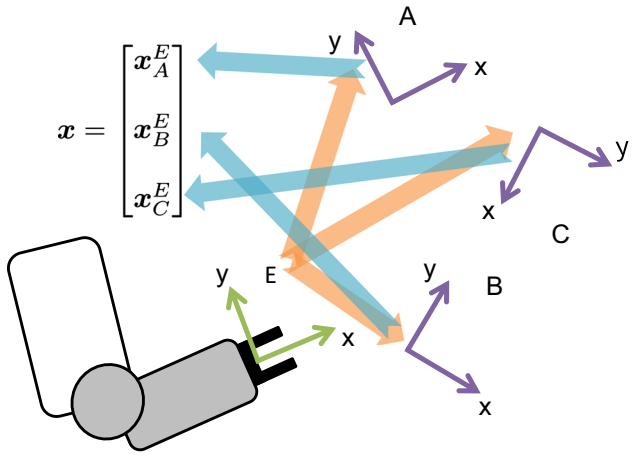


Fig. 17 An illustration of the task-parameterized representation: the interested state, e.g., the pose of the robot end-effector E, is projected into different reference frames in the scene. The resulting state is an augmentation of all relative representations, yielding a high-dimensional state variable

to landmark reference frames and impose a model sparsity to fit finite demonstrations.

The parameters vary because the importance of the via-points and destinations is not static. The inference of model parameters is compatible because the local models are also Gaussians. For the detailed Gaussian inference with a task-parameterized model, we refer to Calinon (2015).

6.2.2.3 An illustration of challenges

One might imagine that the task can be simply addressed by first grouping the trajectories with a simple clustering, e.g., K-means, and then following the closest reference trajectory given the current state. To illustrate the challenges involved in this scenario, we show this is not applicable in terms of both learning and exercising the task constraints.

First, for each demonstration sample, the locations of the starting point and the via-points are different. The invariant constraint of reaching correct via-point and destination is implicit and cannot be trivially revealed from an isotropic distance. Figure 18 shows that the K-means result is poor for assigning demonstrations to the correct behavior mode. The proposed approach is doing a better job because it assesses the similarity with an aggregated nonlinear metric. Here the insight is that the importance of the state dimensions is non-uniform and implicitly correlated to the critical reference frames which depends on the task mode. The proposed approach identifies discriminating feature dimensions through a consideration over a group of naive selections, and as a result, a nontrivial metric emerges and captures the implicit static task constraints.

Secondly, even though a perfect demonstration clustering is given, it is insufficient to decide the mode straightforwardly based on the current state. To see this, we train task-parameterized GMMs (Calinon 2015) over the perfectly clustered data. Then we start a reproduction instance by starting to follow mode 1: {mail location, orange area, mailbox-1} and the reproduction is adapted according to the likelihood of the observed state w.r.t. each mode.

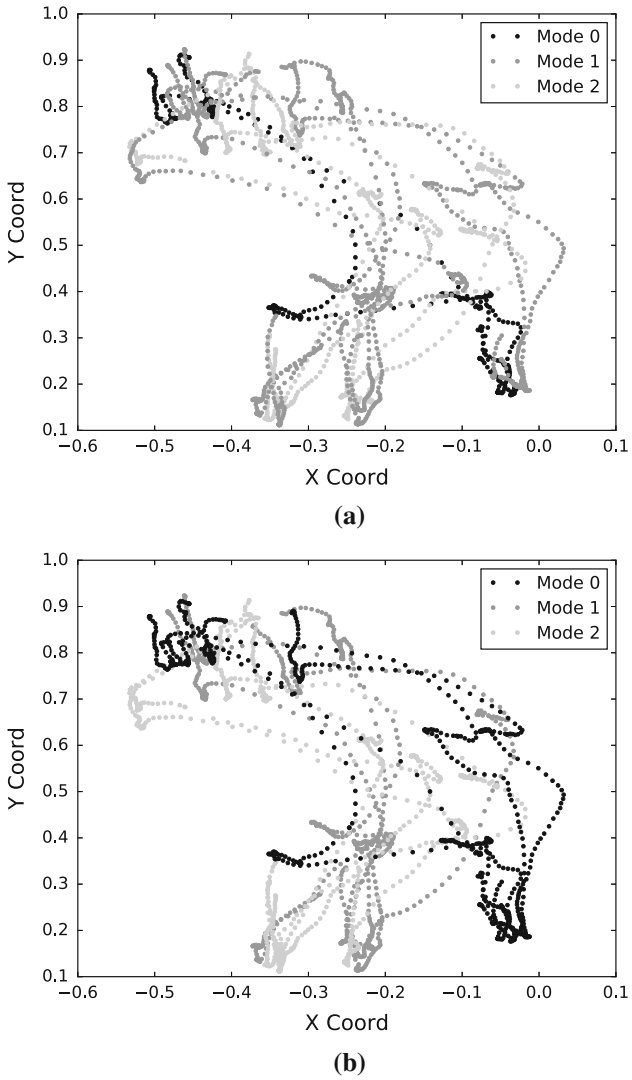


Fig. 18 Clustering demonstration trajectories (dot lines) into three modes: the trajectories are transformed to the mailbox reference frame and projected into the XY surface for the clarity of comparison. The KMeans method takes the best result from 500 random initializations of the cluster centroids. An ideal clustering is supposed to group the demonstrations with a similar behavior mode: trajectories of a same color should reach a same destination. **a** KMeans. **b** Random Embeddings

Figure 19 illustrates a typical reproduction instance. Ideally, the execution should follow the initial mode in the absence of any perturbations. However, the robot actually deviates from the intended intermediate target by heading to the cyan area. This is due to the intrinsic motor noise and the mode ambiguity. Concretely, the robot motor noise will

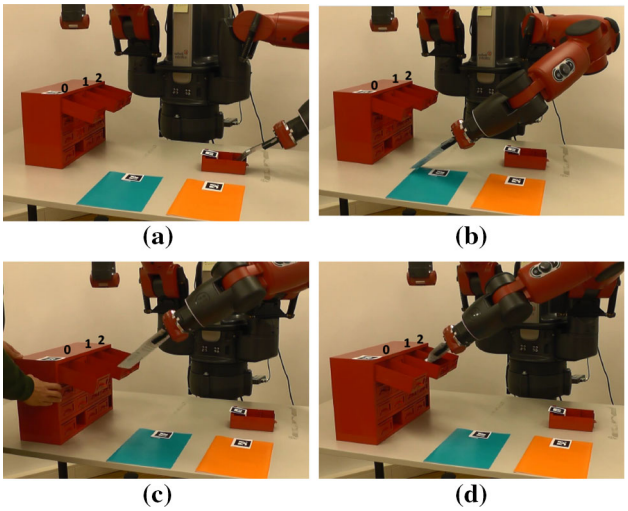


Fig. 19 Task reproduction with the baseline multi-mode behavior cloner. **a**, **b** The robot starts with the intention to follow mode 1 (mail location-orange area-mailbox-1) but heads to the wrong intermediate area under its own motor noise. **c**, **d** The location of mailbox is perturbed hence the mailbox-1 is again the most probable target given the current motion status. The robot delivers the mail to the mailbox-1 even the mail passed the cyan area (Color figure online)

occasionally result in an end-effector position that is more close to one other mode than the current one. Even worse, this effect is aggravated in earlier stages of the execution, in which all modes are follow similar trajectories to reach and collect the mail. Due to this ambiguity, the likelihood of all three modes is close and a change of mode will be triggered even under a small perturbation.

The figures illustrate yet another type of failure, which results from extrinsic disturbances. The robot, having passed via the cyan area, is moving towards mailbox-0. While it is approaching, the mailbox location is perturbed. Therefore, the motion trajectory is leading to mailbox-1, which makes mode 1 more likely, given the likelihood of the current state, and again triggering an erroneous mode shift. The evolution of the mode belief is depicted in Fig. 20. In brief, due to lack of robustness against both intrinsic and extrinsic disturbances, the baseline adaptation cannot reliably reproduce the intended behavior and conform to the demonstration constraints.

6.2.2.4 Results

Figure 21 illustrates successful reproductions, with the proposed latent dynamics enforced. In the first case (the snapshots in the upper row), the robot successfully follows the task mode 1. In second case (snapshots in the middle row), the robot correctly passes the cyan region and reaches the mailbox-0, even if the mailbox is moved on-the-fly. The difference from the baseline adaptation mechanism (Fig. 20) is

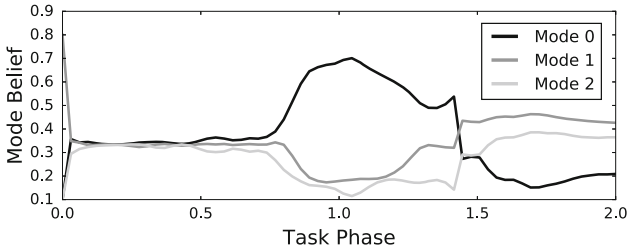


Fig. 20 History of mode activation for a multi-mode behavior cloner: the robot agent always follows the most likely mode given its observation at each time step. This will result in undesired adaptations in certain cases

evidenced from the belief estimation (bottom row of Fig. 21). Although the belief about the initial mode still decreases, because of the ambiguity in the early parts of the trajectory, the prior bias towards the current mode persists. As a result, the task reproduction is robust to the uncertainty about the robot intrinsic dynamics or a step disturbance such as pulling the mailbox away.

We further compare the baseline behavior with our approach by setting different configurations of the via-points. Here the metric is the success rate of the multi-mode controllers for delivering mails to the correct targets under randomly arranged task configurations. The results are given in Table 2. The baseline multi-mode adaptation seldom succeeds. Especially when the intended targets are mailbox-1 or mailbox-2, the robot tends to lose the target while collecting the mail, as already exemplified in Fig. 20. On the contrary, the proposed method performs consistently better, reliably generalizing and executing the motion under various task configurations.

The robustness to external disturbance can also be seen from the point of view of collaboration, where the robot chooses to dominate the execution and reject the human guidance. This is shown in Fig. 22. In this situation, the human intervenes with an impulsive correction, aiming to redirect the delivery to mailbox-0. In light of the intervention, the "human preferred mode" is temporarily more likely w.r.t. the cost values of the current state, as seen in Fig. 22d. However, since the robot has passed the orange intermediate area, a strong prior (that mode 0 is very unlikely) has been established. Thus the robot chooses to ignore the guidance and not violate the constraint imposed by the already executed trajectory.

On the other hand, the robot may also adapt and yield to the human intervention, when such intervention is in accordance with the learned constraints. Figure 23 demonstrates a similar execution but where the human intervention pushes the delivery towards mailbox-2. This example is different from one previously discussed, since the orange via-point is admissible for both modes. Therefore, there is a moderate possibility of switching modes and it does not require much

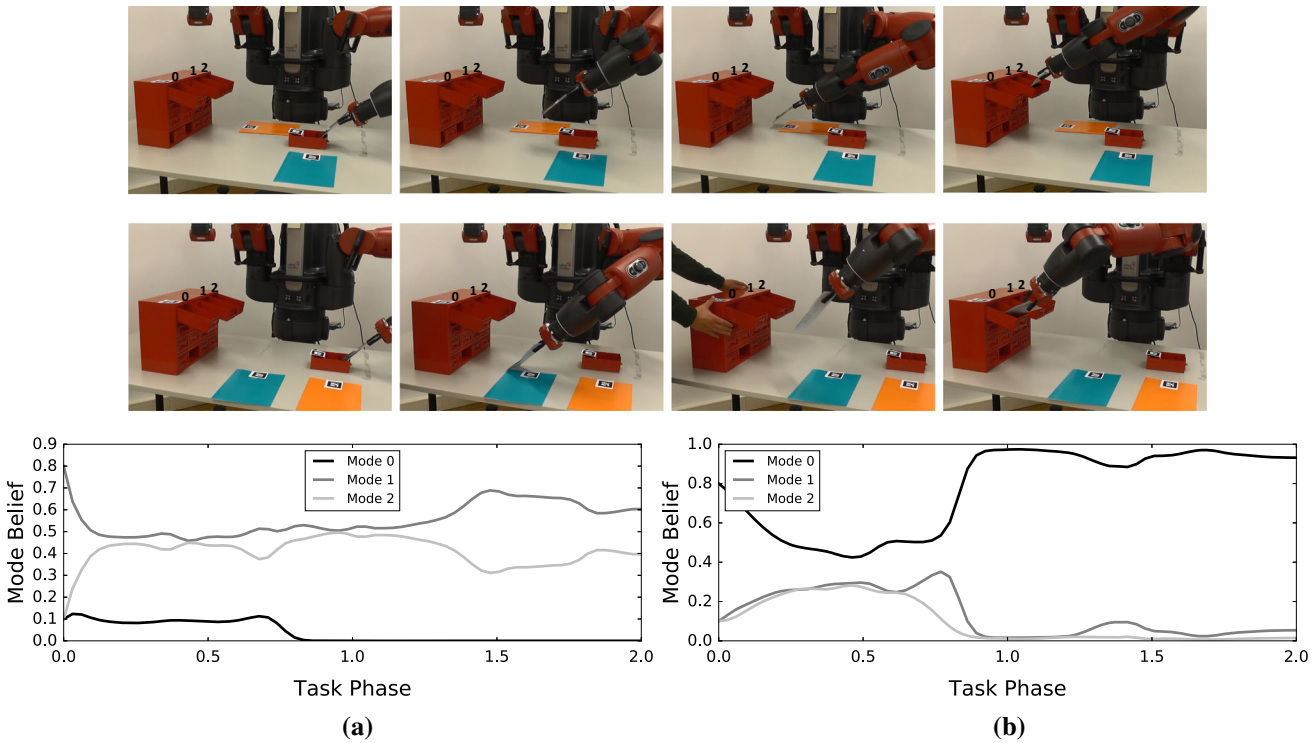


Fig. 21 Task reproduction with the proposed framework under a novel task configuration. The robot adapts the intended motion (mail location-cyan area-mailbox-0) against the external perturbation of moving the mailbox away. **a** Belief evolution under new configuration (upper reproduction). **b** Belief evolution under perturbation on-the-fly (lower reproduction) (Color figure online)

Table 2 Results of task reproduction under different targets and configurations: a reproduction is marked as a success if the robot follows the intended task mode and deliver the mail to the correct target

	Mode 0	Mode 1	Mode 2
Baseline	1/5	0/5	0/5
Proposed approach	5/5	5/5	4/5

For each target mode, five trials are taken with the via-point layout randomly arranged

effort from the human to convey the changed intention and get the robot to collaborate accordingly.

Table 3 gives more results about adaptation under different configurations. In this experiment, a human supervisor has his/her own intended task mode in mind, and intervenes by physically moving the robot motion if he/she thinks that the robot is not following the correct task mode. All combinations of the robot initial mode (R Mode) and the human intention (H Mode) are tested. The metric is the success rate of the collaboration. A collaboration is considered as a success if: (1) the robot identifies the human intention and follows the guidance when the task constraint is fulfilled; (2) the robot follows its own intended motion when the human guidance violates the task constraints. The results demonstrate that the proposed framework allows the robot to understand the human intended target and adapt its motion accordingly throughout almost all of the test cases.

Some additional insights regarding the behavior of our approach can be elicited from Fig. 24. This figure overlaps the layout of the workspace and the corresponding cost evaluation, with the dimensions of mode, time and Z-axis collapsed. It is clear that the peaks of the cost coincide with the key objects in the scene. Moreover, steep cost gradient is visible due to the high consistency of the demonstration behavior around these objects, especially the two intermediate spots. They are automatically identified as critical and discriminative frames. Passing either of them will lead to very strong constraints, preventing the follow-up motion to switch to the other modes, unless if such switching is compatible to the constraint (for example, switching between modes 1 and 2).

In all, this experiment showcases a task in which the instantaneous sensory reading is not sufficient to determine the desired action, which implies that imitation through multi-mode behavior cloning is not directly applicable. The proposed framework, leveraging the extracted demonstration modes and cost-to-go functions, provides a way to evaluate the sensory feedback and infer the intended task mode. With a prior upon the dynamical mode transition combined, a mixed

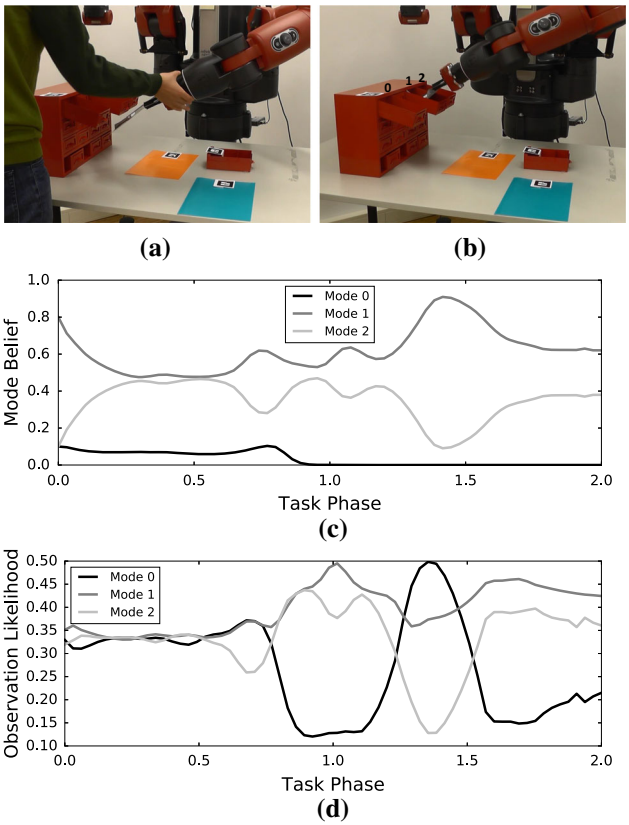


Fig. 22 Reject to human intervention of guiding the delivery to an unlikely goal: the robot holds a low belief about the mode of reaching mailbox-0 since it has passes the orange area. **a** Intervention. **b** Adaptation. **c** Estimated mode belief. **d** Evolution of observation likelihood (Color figure online)

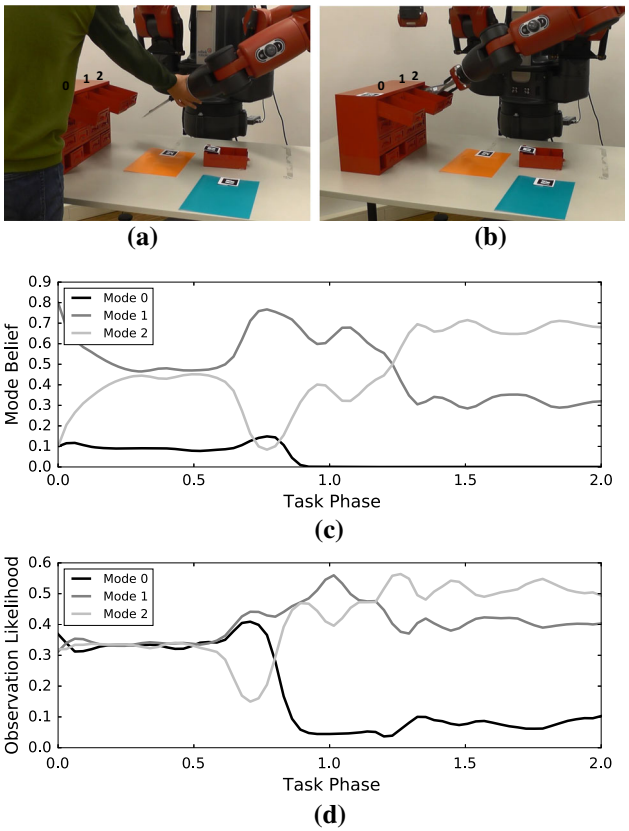


Fig. 23 Yielding to the external perturbation: the robot collaborates by adjusting the motion (mail location-orange area-mailbox 1) to an alternative target mailbox-2. The prior of mode 1 is not completely dominant against mode 2. **a** Intervention. **b** Adaptation. **c** Estimated mode belief. **d** Evolution of observation likelihood (Color figure online)

behavior emerges: the robot can automatically decide when and where to collaborate with/reject human interventions based upon constraints learned from the demonstrations.

7 Conclusions and future work

This paper presents an approach to learn ensemble cost-to-go function models for robotic motion adaptation under human supervision. We demonstrate how the principle of ensemble models allows for rapid learning of a strong model by aggregating a group of simple IOC models. These models are naive but are much more efficient to train than directly tackling the original and much harder problem. We propose the use of quadratic parameterization as a proper candidate, leading to efficient learning and LQ-like control synthesis for demonstrations with multiple latent styles. The proposed framework is further extended by introducing an extra model over these latent states. The dynamical estimation enables to alter learned modes to realize online motion adaptation through inferring human intentions. Our analysis and experiments validate the framework in robotic tasks involving human interventions.

Table 3 Results of task adaptation under human intervention for different configurations: an adaptation is marked as a success if the robot (R) follows the human (H) intended task mode under the intervention and deliver the mail to the correct target

	R Mode 0	R Mode 1	R Mode 2
H Mode 0	5/5	5/5	5/5
H Mode 1	5/5	5/5	4/5
H Mode 2	4/5	5/5	4/5

For each target mode, five trials are taken with the via-point layout randomly arranged

The framework demonstrates its capability of generalizing to untrained task configurations. This is enabled by the adopted task-parameterized feature. Generally, the generalization capability depends on the feature design. The random subspace embedding can be subject to various choices about the decision boundary and feature selection, capturing data

Fig. 24 Contour of the learned cost-to-go functions with the time and Z axes collapsed. The areas with dense contours indicate the demonstrations are locally consistent hence some of the regions will be discriminative for differentiating motion modes

structures beyond the axis-aligned grid used in this paper. We refer to Criminisi et al. (2012) for more details.

A quadratic cost function demands a feature space in which an Euclidean distance serves as an effective norm. Task-relevant features can be designed to fulfill this requirement. For instance, forward kinematics can be used to project the raw joint positions to a task-relevant feature space, e.g., the robot end-effector or manipulated object pose. One can also introduce features based on robot dynamics for adding more complexities such as inverse dynamics control. Indeed, choosing a proper task-relevant feature entails a manual design. This is definitely one of the most phenomenal problems, not only in IOC, but also in general AI and machine learning. To put it in perspective, this framework is not straightforwardly applicable to extremely high-dimensional demonstrations (e.g., visual pixels) since the statistics are nontrivial and hard to be handcrafted. The issues might be partially resolved by parameterizing the cost models with complex features whose representations can be jointly learned from the demonstration data. Such a representation learning strategy that embeds high dimensional features has demonstrated its success in the control synthesis with a well-defined goal of the task (Watter et al. 2015). It will be interesting to also incorporate this idea into imitation learning.

The proposed framework is more restrictive in terms of the assumed dynamics and cost form, comparing with a traditional MDP. However, these constrains are natural for the interested robotic tasks, which often feature a control-affine dynamics with a continuous action space and a criterion penalizing control effort for the motion smoothness. Also, as pointed in Dvijotham and Todorov (2010), traditional MDPs can be approximately resolved under the linearly-solvable system in the similar way of relaxing an integer programming into a linear programming.

The framework requires the task-dependent mode transition as prior knowledge. One open question is how the hybrid system can be estimated from data. Maximizing the model flexibility of both latent dynamics and state optimality might raise a *severe non-identifiable* issue (Frigola et al. 2014). It is worth to investigate proper regularization or learning the discrete components alone.

Here, the ensemble method is based on tree and bagging techniques. A bagging based ensemble alleviates overfitting by smoothing over multiple predictions. Hence, the approach is robust to noisy demonstrations. Moreover, tree-based techniques generally scale well to a large dataset. One of the limitations is that the framework might face difficulties in selecting model parameters to learn from a limited number of demonstrations. The boosting scheme might be a better choice in this case, since it aggregates for improving the predictive power while the goal of bagging is variance reduction. Unlike the tree-based bagging, however, it is now unclear in what form the weak models can relate to a meaningful IOC problem that can be efficiently solved. Also, the standard boosting often aggregates the decisions through majority vote, which might be problematic for obtaining a continuous cost.

Another direction to explore is how the learned models can be used as priors to steer the posterior trajectory optimization. Models with a similar form as the one proposed here have been applied to probabilistic trajectory planning, where dynamical constraints are nontrivial or even model-free (Calinon et al. 2012; Kobilarov 2012). In light of that, the present framework, as a generative model, can benefit the downstream control synthesis in terms of its exploration, refinement, generalization and ultimately, integration with learning from human demonstrations.

Acknowledgements This work is partially funded by Swiss National Center of Robotics Research and national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013 and the doctoral Grant (ref. SFRH/BD/51933/2012) under IST-EPFL Joint Doctoral Initiative.

Appendix: Proof for the MaxEnt approximation of probabilistic model with quadratic cost-to-go

Substituting the Gaussian passive dynamics and the quadratic cost-to-go function, we have:

$$P(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) = \frac{e^{-\frac{1}{2}\|\boldsymbol{x}_{t+1}-f(\boldsymbol{x}_t))\|_{\Sigma_0^{-1}} - \frac{1}{2}\|\boldsymbol{x}_{t+1}-\boldsymbol{\mu}\|_{\Lambda}}}{\int e^{-\frac{1}{2}\|\boldsymbol{x}'_{t+1}-f(\boldsymbol{x}_t)\|_{\Sigma_0^{-1}} - \frac{1}{2}\|\boldsymbol{x}'_{t+1}-\boldsymbol{\mu}\|_{\Lambda}} d\boldsymbol{x}'_{t+1}} \tag{16}$$

The corresponding log-likelihood can be written as

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = & -\frac{1}{2}(\boldsymbol{x}_{t+1} - f(\boldsymbol{x}_t))^T \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}_{t+1} - f(\boldsymbol{x}_t)) \\
& -\frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu}) \\
& -\log \int \underbrace{e^{-\frac{1}{2}(\boldsymbol{x}'_{t+1} - f(\boldsymbol{x}_t))^T \boldsymbol{\Sigma}_0^{-1}(\boldsymbol{x}'_{t+1} - f(\boldsymbol{x}_t))}}_{\leq 1 \text{ and positive}} \\
& \times e^{-\frac{1}{2}(\boldsymbol{x}'_{t+1} - \boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{x}'_{t+1} - \boldsymbol{\mu})} d\boldsymbol{x}_{t+1} \\
\geq & \underbrace{-\frac{1}{2}(\boldsymbol{x}_{t+1} - f(\boldsymbol{x}_t))^T \Sigma_0^{-1}(\boldsymbol{x}_{t+1} - f(\boldsymbol{x}_t))}_{\text{Independent of } \boldsymbol{\mu} \text{ and } \boldsymbol{\Lambda}} \\
& -\frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu}) \\
& +\frac{d}{2}\log(2\pi) + \frac{1}{2}\log|\boldsymbol{\Sigma}_0| \\
& \underbrace{-\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Lambda}^{-1}|}_{-\log[\int e^{-\frac{1}{2}(\boldsymbol{x}'_{t+1} - \boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{x}'_{t+1} - \boldsymbol{\mu})} d\boldsymbol{x}'_{t+1}]} \\
= & -\frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu})^T \boldsymbol{\Lambda}(\boldsymbol{x}_{t+1} - \boldsymbol{\mu}) \\
& -\frac{1}{2}\log|\boldsymbol{\Lambda}^{-1}| + \text{const} \\
= & \hat{\mathcal{L}}(\boldsymbol{\mu}, \boldsymbol{\Lambda})
\end{aligned}
\tag{17}
$$

where d denotes the state dimension. The exponential from the passive dynamics (the third line of the equation) can be considered as a positive coefficient that is always less than one. Replacing the coefficient with one results in a simple integral of Gaussian function (the exponential of negative cost-to-go function, line 7), which is always larger than or equal to the integral involving passive dynamics. We can obtain a lower bound of the original likelihood by instead subtracting this simplified integral. The MaxEnt estimation $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_{t+1}^i$ and $\boldsymbol{\Lambda}^{-1} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{x}_{t+1}^i - \boldsymbol{\mu})(\boldsymbol{x}_{t+1}^i - \boldsymbol{\mu})^T$ happens to be the optimal solution to the likelihood lower-bound $\hat{\mathcal{L}}$. And the gap shrinks as noise magnitude $\|\boldsymbol{\Sigma}_0\| \to \infty$, with the approximation degenerating to the MaxEnt formulation.

References

Abdolmaleki, A., Lau, N., Paulo Reis, L.,& Neumann, G. (2016). Contextual stochastic search. In *Proceedings of the 2016 on genetic and evolutionary computation conference companion, GECCO '16 companion* (pp. 29–30). New York, NY: ACM.

Akgun, B., Cakmak, M., Yoo, J. W.,& Thomaz, A. L. (2012). Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *Proceedings of the ACM/IEEE international conference on human-robot interaction (HRI)* (pp 391–398). New York, NY.

Bagnell, J. A. D. (2015). An invitation to imitation. Tech. Rep. CMU-RI-TR-15-08, Robotics Institute, Pittsburgh, PA.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Calinon, S. (2015). Robot learning with task-parameterized generative models. In *Proceedings of the international symposium of robotics research (ISRR)*.

Calinon, S., Pervez, A.,& Caldwell, D. G. (2012). Multi-optima exploration with adaptive Gaussian mixture model. In *Proceedings of the international conference on development and learning (ICDL-EpiRob)*. San Diego, CA.

Calinon, S., Bruno, D.,& Caldwell, D. G. (2014). A task-parameterized probabilistic model with minimal intervention control. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 3339–3344).

Criminisi, A., Shotton, J., & Konukoglu, E. (2012). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, *7*(2–3), 81–227.

Dvijotham, K.,& Todorov, E. (2010). Inverse optimal control with linearly-solvable mdps. In *Proceedings of the international conference on machine learning (ICML)* (pp. 335–342).

Englert, P., Paraschos, A., Peters, J.,& Deisenroth, M. P. (2013). Model-based imitation learning by probabilistic trajectory matching. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 1922–1927).

Ewerton, M., Neumann, G., Lioutikov, R., Ben Amor, H., Peters, J.,& Maeda, G. (2015). Learning multiple collaborative tasks with a mixture of interaction primitives. In *IEEE international conference on robotics and automation* (pp. 1535–1542).

Finn, C., Levine, S.,& Abbeel, P. (2016). Guided cost learning: Deep inverse optimal control via policy optimization. In *Proceedings of the international conference on machine learning (ICML) abs/1603.00448*.

Frigola, R., Chen, Y., & Rasmussen, C. E. (2014). Variational Gaussian Process state-space models. In *Proceedings of neural information processing systems (NIPS)*.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42.

Kalakrishnan, M., Pastor, P., Righetti, L.,& Schaal, S. (2013). Learning objective functions for manipulation. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 1331–1336).

Kalman, R. E. (1964). When is a linear control system optimal. *Journal of Basic Engineering*, *86*, 51–60.

Kappen, H. J., Gmez, V., & Opper, M. (2012). Optimal control as a graphical model inference problem. *Machine Learning*, *87*(2), 159–182.

Khansari, M., Kronander, K.,& Billard, A. (2014). Modeling robot discrete movements with state-varying stiffness and damping: A framework for integrated motion generation and impedance control. In *Proceedings of robotics: Science and systems (RSS)*.

Kobilarov, M. (2012). Cross-entropy motion planning. *International Journal of Robotics Research*, *31*(7), 855–871.

Kukliski, K., Fischer, K., Marhenke, I., Kirstein, F., aus der Wieschen, M. V., Sølvason, D., Krüger, N.,& Savarimuthu, T. R. (2014). Teleoperation for learning by demonstration: Data glove versus object manipulation for intuitive robot control. In *Ultra modern telecommunications and control systems and workshops (ICUMT), 2014 6th international congress on* (pp. 346 – 351). IEEE.

Levine, S., & Koltun, V. (2012). Continuous inverse optimal control with locally optimal examples. In *Proceedings of the international conference on machine learning (ICML)*.

Levine, S., Popovic, Z., & Koltun, V. (2011). Nonlinear inverse reinforcement learning with gaussian processes. In *Proceedings of neural information processing systems (NIPS)* (pp. 19–27). Curran Associates, Inc.

Monfort, M., Liu, A., & Ziebart, B. D. (2015). Intent prediction and trajectory forecasting via predictive inverse linear-quadratic regulation. In *Proceedings of the twenty-ninth AAAI conference on artificial intelligence, AAAI'15* (pp 3672–3678). AAAI Press.

Nehaniv, C. L., & Dautenhahn, K. (2002). The correspondence problem. In K. Dautenhahn & C. L. Nehaniv (Eds.), *Imitation in animals and artifacts* (pp. 41–61). Cambridge, MA: MIT Press.

Nikolaidis, S., Ramakrishnan, R., Gu, K., & Shah, J. (2015). Efficient model learning from joint-action demonstrations for human-robot collaborative tasks. In *Proceedings of the ACM/IEEE international conference on human-robot interaction (HRI)* (pp. 189–196). New York, NY: ACM.

Pomerleau, D. A. (1991). Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, *3*(1), 88–97.

Ratliff, N., Bagnell, J. A. D., & Zinkevich, M. (2006). Maximum margin planning. In *Proceedings of the international conference on machine learning (ICML)*.

Rozo, L., Bruno, D., Calinon, S.,& Caldwell, D. G. (2015). Learning optimal controllers in human-robot cooperative transportation tasks with position and force constraints. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS) Hamburg, Germany* (pp. 1024–1030).

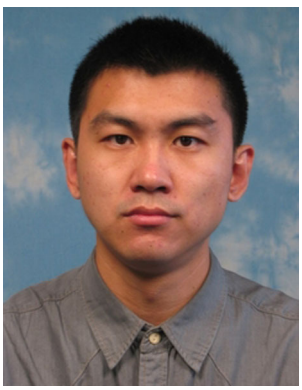
Todorov, E. (2009). Compositionality of optimal control laws. In *Proceedings of neural information processing systems (NIPS)* (pp. 1856–1864). Curran Associates Inc., USA.

Watter, M., Springenberg, J. T., Boedecker, J., & Riedmiller, M. A. (2015). Embed to control: A locally linear latent dynamics model for control from raw images. CoRR abs/1506.07365.

Wulfmeier, M., Ondruska, P., & Posner, I. (2015). Maximum entropy deep inverse reinforcement learning. CoRR abs/1507.04888.

Yin, H., Alves-Olivera, P., Melo, F. S., Billard, A., & Paiva, A. (2016). Synthesizing robotic handwriting motion by learning from human demonstrations. In *Proceedings of international joint conference on artificial intelligence (IJCAI)*.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., & Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *Proceedings of the national conference on artificial intelligence (AAAI)* (pp. 1433–1438).



Hang Yin is a student of the Swiss Federal Institute of Technology in Lausanne (EPFL) and the Instituto Superior Técnico (IST), University of Lisbon. He received a B.Eng. in Mechanical Engineering, and a M.Sc. in Mechatronics from Shanghai Jiao Tong University. He is now working in the LASA laboratory (EPFL) and the GAIPS group (IST) for his Ph.D. study. His research interests are robot motion representation and control, machine learning and their application in human-robot interaction.

Francisco S. Melo is an associate Professor at Instituto Superior Técnico, University of Lisbon, and a researcher in the GAIPS group of INESC-ID. He received his Ph.D. in Electrical and Computer Engineering at Instituto Superior Técnico in 2007. Since then he held appointments in the Computer Vision Lab of the Institute for Systems and Robotics, in Lisbon, and in the Computer Science Department of Carnegie Mellon University, in the U.S.A. His research addresses problems within machine learning, particularly on reinforcement learning, planning under uncertainty, multi-agent and multi-robot systems, developmental robotics, and sensor networks.

Ana Paiva is a research group leader of GAIPS at INESC-ID and a Full Professor at Instituto Superior Técnico, University of Lisbon. She is well known in the area of Intelligent Agents and Multi-agent Systems, Artificial Intelligence, Human-Robot Interaction and Affective Computing. After her PhD in the UK, she returned to Portugal where she created a group on intelligent agents and synthetic characters (GAIPS). Her research is focused on the affective elements in the interactions between users and machines. She served as a member or a chair of numerous international conference and workshops, in particular she was co-chair of ACII in 2007, local chair for AAMAS in 2008 and co-PC-Chair of HRI in 2016. She has (co)authored over 200 publications in refereed journals, conferences and books. She is a member of the Scientific Advisory Committee of Science Europe since 2016.

Aude Billard is Professor of Micro and Mechanical Engineering, and the head of the LASA Laboratory at the School of Engineering at the Swiss Federal Institute of Technology in Lausanne. She received a M.Sc. in Physics from EPFL (1995), a M.Sc. in Knowledge-based Systems (1996) and a Ph.D. in Artificial Intelligence (1998) from the University of Edinburgh. She was the recipient of the Intel Corporation Teaching award, the Swiss Natio -nal Science Foundation career award in 2002, the Outstanding Young Person in Science and Innovation from the Swiss Chamber of Commerce and the IEEE-RAS Best Reviewer award in 2012. Aude Billard served as an elected member of the Administrative Committee of the IEEE Robotics and Automation society (RAS) for two terms (2006–2008 and 2009–2011) and is the chair of the IEEE-RAS Technical Committee on Humanoid Robotics. Her research interests focus on machine learning tools to support robot learning through human guidance. This extends also to research on complementary topics, including machine vision and its use in human-robot interaction and computational neuroscience to develop models of motor learning in humans.