



Environment-adaptive interaction primitives through visual context for human–robot motor skill learning

Yunduan Cui¹ · James Poon² · Jaime Valls Miro² · Kimitoshi Yamazaki³ · Kenji Sugimoto¹ · Takamitsu Matsubara¹

Received: 31 December 2016 / Accepted: 9 August 2018 / Published online: 18 August 2018
© The Author(s) 2018

Abstract

In situations where robots need to closely co-operate with human partners, consideration of the task combined with partner observation maintains robustness when partner behavior is erratic or ambiguous. This paper documents our approach to capture human–robot interactive skills by combining their demonstrative data with additional environmental parameters automatically derived from observation of task context without the need for heuristic assignment, as an extension to overcome shortcomings of the interaction primitives framework. These parameters reduce the partner observation period required before suitable robot motion can commence, while also enabling success in cases where partner observation alone was inadequate for planning actions suited to the task. Validation in a collaborative object covering exercise with a humanoid robot demonstrate the robustness of our environment-adaptive interaction primitives, when augmented with parameters directly drawn from visual data of the task scene.

Keywords Learning from demonstration · Physical human–robot interaction · Human–robot coordination

1 Introduction

The last decade has seen a significant increase in interest towards ‘social’ human-interactive robots (Vircikova et al. 2012), in both domestic and industrial environments. With predictions that approximately 1.3 million new industrial robots will be installed into factories worldwide between 2015 and 2018 (International Federation of Robotics 2015), there follows a strong research incentive to accelerate the rate at which robots can be incorporated into workspaces

alongside human workers. As an alternative to the manual tuning of parameters for each task in the rapidly increasing scope of activities that domestic and industrial robots will be expected to perform, demonstration learning provides robots the ability to synthesize their own operational parameters from observations of the activity to be conducted. Demonstration learning is well suited to humanoid robots, due to the intuitive correspondence between a teacher’s physical movements and what can be considered to be acceptable robot behaviour.

This paper presents the complete environment-adaptive interaction primitives (EaIPs) framework (Fig. 1) initially proposed in Cui et al. (2016), which in turn builds upon interaction primitives (IPs) (Ben Amor et al. 2014). IPs allow a robot to generate suitable collaborative actions by inferring parameters for Dynamic Movement Primitives (DMPs) (Ijspeert et al. 2013), following a brief observation of human partner movement. Groups of people routinely handle objects in a physically collaborative manner, such as the movement of bulky furniture items or laying out a large table-cloth. The motivation behind EaIPs is to enable robots participating in such tasks to incorporate additional information from the immediate environment as people instinctively do, instead of solely relying upon observations of their partner as is

Yunduan Cui and James Poon have contributed equally to this work.

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s10514-018-9798-2>) contains supplementary material, which is available to authorized users.

✉ Yunduan Cui
cuiyunduan@gmail.com

¹ Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma, Nara, Japan

² Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW, Australia

³ Faculty of Engineering, Shinshu University, Matsumoto, Nagano, Japan

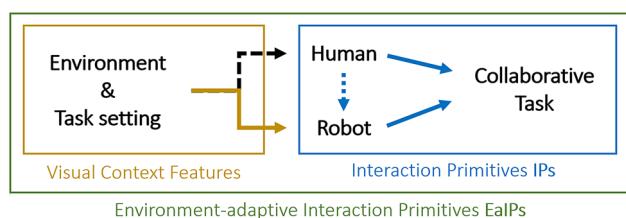


Fig. 1 Schematic overview of the complete EaIPs framework: an extension of IPs to consider additional environmental parameters from visual contextualization

the case for IPs. EaIPs expanded upon IPs by allowing for the consideration of these ‘environmental’ parameters, which are taken alongside partner observations during training and runtime. The effects of including environmental parameters are twofold; firstly, a faster robot response is made possible as an actionable level of confidence can be obtained in less time. Secondly, the system becomes hardened against partner observations that would not ensure safety or task compliance, as such observations are no longer the sole information source for robot action planning.

The main contribution of this paper is the combination of EaIPs with the automatic derivation of suitable environmental parameters. The driving motivation behind this step is to enable a more organic learning of human–robot collaborative behaviors, towards a complete interaction framework. Instead of relying upon parameters explicitly provided prior to execution which are tuned to the task at hand, as per Cui et al. (2016), the objective of this work is to generate suitable motor skills solely upon observations from scene and partner as would naturally occur within a human–human team. To draw such contextual parameters from the scene, we utilize a Convolutional Neural Network (CNN) (Lecun et al. 1998) based object detector to yield contextual information concerning the human–robot task. Here we obtain bounding boxes and class labels of detected objects the CNN has been trained upon; given image size properties about the object that serves as the focus of the interaction task, the EaIPs then also consider observations of its human partner for an action that caters to both information sources. We utilize YOLO (Redmon and Farhadi 2017) to obtain such information, as detailed in Sect. 4.2.1.

The advantage of utilizing CNNs as a preprocessing step is their ability to recognize a broad range of objects beyond the scope of IPs/EaIPs training data, thus greatly reducing the risk of overfitting due to the difficulty of obtaining substantial quantity of training data for human–robot interactions. The correlation of the bounding box to the object’s position and size to physical space help reduce the likelihood of unsuitable robot responses.

The remainder of the paper is structured as follows. Related work is outlined in Sect. 2. Details for IPs and EaIPs are available under Sect. 3. Our validation task is the col-

laborative covering of large objects with a plastic bag in both simulation (Sect. 4.1) and a Baxter humanoid robot (Sect. 4.2). Discussions of results and conclusions follow in Sects. 5 and 6.

2 Related work

It is common in the learning of more complex robot tasks to base actions from inferences drawn from training data, rather than to tune operating parameters manually. Depending on the scenario at hand this data is usually obtained by teleoperation of the robot such as by kinesthetic teaching e.g. Kormushev et al. (2010), Kronander and Billard (2014) or if possible, direct control of the robot which will be executing autonomous behaviors e.g. Goil et al. (2013), Soh and Demiris (2015) although the latter is often restricted to lower DOF systems such as ground vehicles. By gathering data consisting of desirable robot behavior and observations of a partner in a human–robot interactive exercise, an aim of learning from this data is the synthesis of suitable robot actions after anticipating outcomes of human actions to achieve fluent interaction, as is the case with both IPs and EaIPs.

Examples of partner anticipation include the work by Huang and Mutlu (2016), using Support Vector Machines to predict eye gaze and treat it as the human partner’s focus of attention, which formed the basis for subsequent robot activity. A reinforcement learning-based (Sutton and Barto 1998) approach to human intention estimation is investigated by Awais and Henrich (2013), where it is followed by a particle filter for probabilistic action selection. Taha et al. (2011) used Partially Observable Markov Decision Processes to infer the navigational intention of the user of a sensor-equipped robotic mobility aid, used to determine suitable assistive driving behaviors for the robot. Another example of user anticipation in the assistive robotics space is the work by Patel et al. (2014), utilizing Hierarchical Hidden Markov Models to predict a user’s intention to various levels of abstraction ranging from lower level activities such as ‘driving cardinal North’ to higher level activities such as ‘going to bed’.

DMPs have been used extensively for robot control due to their robustness, guarantee of convergence and ease of scaling spatially and temporally. As in interaction primitives, they are commonly used as the basis for complex robot behaviors which are then executed according to a higher-level framework above them; recent examples of their utilization in this manner include Fitzgerald et al. (2015), Mandery et al. (2016), Lioutikov et al. (2016) and Ude et al. (2010).

The collaborative manipulation of objects remains a challenging objective, as the nature of the object and any of its interactive counterparts being manipulated must be con-

sidered in conjunction with the behavior of the human partner. Sheng et al. (2015) uses reinforcement learning and observations of human behavior to keep a table held between the human and robot in a horizontal alignment, while Lawitzky et al. (2010) devised a framework that aimed to share the load between the two agents. As an alternative for robots with multi-modal sensory capabilities, Kruse et al. (2015) considered both the forces applied to a sheet of fabric held taut between the human and the robot as well as perceived deformations. The manipulation of deformable materials is still a challenging area of investigation due to a prohibitively large state space, although work is being done towards the modeling of smaller deformable items such as pieces of clothing e.g. Doumanoglou et al. (2014).

Convolutional Neural Networks are designed around the handling of structured input data such as images, as opposed to conventional Neural Networks which treat input variables with no particular regard to the potential for patterns within regions of input elements. This makes them particularly well suited to tasks such as object recognition and scene classification, as can be seen in recent works such as Krizhevsky et al. (2012), Simonyan and Zisserman (2014) and Szegedy et al. (2015). The direct application of deep learning to robot control remains challenging primarily due to the difficulties in accumulating requisite amounts of training data. As a result this remains a relatively recent initiative, e.g. in Pervez et al. (2017) where CNNs are used to regress robot action parameters. Other recent works include Finn et al. (2016), Pinto and Gupta (2016). Rather than serving as an oracle aimed at directly shaping robot actions, here a CNN serves as a mechanism to reduce uncertainty by examining the task scene for additional information, which in turn augments observations of the human partner for EaIPs.

3 Approach

3.1 Dynamic movement primitives

In the area of trajectory control and planning, Dynamic Movement Primitives (Ijspeert et al. 2013) were proposed to stably represent complex motor actions that can be flexibly adjusted without manual parameter tuning. In this work, DMPs are employed to encode trajectories of both human and robot movements. For each degree of freedom, a trajectory is defined as the following:

$$\ddot{y}(t) = \left(\alpha_y \left(\beta_y (g - y(t)) - \left(\frac{\dot{y}(t)}{\tau} \right) \right) + f(x_t) \right) \tau^2 \quad (1)$$

where α_y and β_y are constants, y , g are the state variable and target position of the trajectory, respectively. τ is a time constant and t is the time step. $f(x_t)$ is the forcing function

built by M Gaussian basis functions and a corresponding $M \times 1$ weights vector w :

$$f(x_t) = \frac{\sum_{i=1}^M \psi_i(x_t) w_i x_t}{\sum_{j=1}^M \psi_j(x_t)} = \phi(x_t)^T w, \quad (2)$$

x follows a canonical system: $\dot{x} = -\alpha_x x \tau$ where x is initialized as $x_0 = 1$.

To encode a T step trajectory $y = [y(t), \dot{y}(t), \ddot{y}(t)]_{t=1:T}^T$ by a weight vector w in the DMP, the forcing function that reproduces the sample trajectory from the t th step is firstly calculated according to Eq. (1):

$$f(x_t) = \frac{1}{\tau^2} \ddot{y}(t) - \alpha_y \left(\beta_y (g - y(t)) - \frac{\dot{y}(t)}{\tau} \right). \quad (3)$$

Expressing the DMP as $f = \Phi w$, with basis functions $\Phi = [\phi(x_1), \dots, \phi(x_T)]^T$ and $f = [f(x_1), \dots, f(x_T)]^T$, the weights vector w can be obtained via least squares regression:

$$w = (\Phi^T \Phi)^{-1} \Phi^T f. \quad (4)$$

3.2 Interaction primitives for collaborative human–robot tasks

Motivated by the desire of engaging in cooperative activities between human partners and robots using DMPs, interaction primitives (IPs) (Ben Amor et al. 2014) were proposed to extend DMPs to human–robot activities. After learning from demonstration data and maintaining a distribution over DMP parameters, IPs achieve human–robot cooperation through the following steps:

1. observe partial trajectory from human partners and identify the current phase of the interaction
2. compute the distribution over DMP parameters to control robot to cooperate with human partners based on the partially observed trajectory.

In the first step, Dynamic Time Warping (DTW) (Sakoe and Chiba 1978) is employed to estimate the phase of observed human movement. Given one partially observed human movement $[y_1^*, \dots, y_{T'}^*]^T$, and one full human reference movement observed during demonstration $[y_1, \dots, y_T]^T$, DTW measures the similarity between these two time series and yields the index t^* , reflecting the frame in the reference movement which produces minimal costs with respect to the query movement, i.e., $[y_1^*, \dots, y_{T'}^*]^T$ is close to $[y_1, \dots, y_{t^*}]^T$. The estimated phase of a partially observed human movement is therefore:

$$x^* = \exp \left(-\alpha_x \left(\frac{t^*}{T} \right) \tau \right). \quad (5)$$

In the second step, robot motor skills are predicted based on a partial observation of the human’s movement. We first prepare S sets of N DoFs trajectories that are temporally normalized to the same length T as the training samples:

$$Y = [Y_{human}, Y_{robot}] = \begin{bmatrix} \mathbf{y}_1^1 & \dots & \mathbf{y}_N^1 \\ \vdots & \ddots & \vdots \\ \mathbf{y}_1^S & \dots & \mathbf{y}_N^S \end{bmatrix} \quad (6)$$

where N is the total number of DoFs for both human and robot. Defining \mathbf{y}_i^j , \mathbf{w}_i^j and \mathbf{g}_i^j as the trajectory, weights vector and target position of the i th DoF in the j th demonstration respectively, $\theta^{[j]} = [\mathbf{w}_1^{jT}, \mathbf{g}_1^j, \dots, \mathbf{w}_N^{jT}, \mathbf{g}_N^j]^T$, $j = 1, \dots, S$ is the parameter vector for the DMPs as learned from $[\mathbf{y}_1^j, \dots, \mathbf{y}_N^j]$. Thus $p(\theta)$, the distribution among the parameter vector samples $\theta^{[j]}$, $j = 1, \dots, S$, follows:

$$p(\theta) = \mathcal{N}(\theta | \mu_\theta, \Sigma_\theta), \quad (7)$$

$$\mu_\theta = \frac{\sum_{j=1}^S \theta^{[j]}}{S}, \quad (8)$$

$$\Sigma_\theta = \frac{\sum_{j=1}^S (\theta^{[j]} - \mu_\theta)(\theta^{[j]} - \mu_\theta)^T}{S} \quad (9)$$

where $\theta = [\theta_{human}, \theta_{robot}]^T$ contains the parameter vectors of both human and robot.

After partially observing the human’s movement and estimating their phase x^* according to a reference movement via DTW, the trajectories $Y_{human}^* = [\mathbf{y}_1^*, \dots, \mathbf{y}_n^*]^T$ are resampled from the observed movement where $n < N$ is the DOF of the human’s movements. The unavailable trajectories of the robot Y_{robot}^* are set to $\mathbf{0}$. Defining $Y^* = [Y_{human}^*, Y_{robot}^*]$, the prediction of both human and robot’s parameter vector is represented by:

$$p(\theta | Y^*) \propto p(Y^* | \theta) p(\theta). \quad (10)$$

The likelihood $p(Y^* | \theta)$ is modelled by a Gaussian distribution over the forcing function:

$$p(Y^* | \theta) \sim \mathcal{N}(F^* | \Omega \theta, \sigma^2 I) \quad (11)$$

where F^* has two parts: $F_{human}^* = [f_1^*, \dots, f_n^*]^T$ is the observed forcing function of Y_{human}^* , its element is given by:

$$f_i^*(x_t) = \frac{1}{\tau^2} \ddot{y}_i^*(t) - \alpha_y \left(-\beta_y y_i^*(t) - \frac{\dot{y}_i^*(t)}{\tau} \right). \quad (12)$$

F_{robot}^* is the unavailable forcing function of robot and set as $\mathbf{0}$. The matrix $\Omega \theta$ contains the forcing function with relationship to $\tilde{\Phi}_t = [\phi(x_t)^T, \alpha_y \beta_y]$ over learning samples for $1 \leq t \leq t^*$:

$$\Omega \theta = \begin{bmatrix} \tilde{\Phi} & 0 & \dots & \dots \\ 0 & \tilde{\Phi} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & \dots & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{w}_N \\ \mathbf{g}_N \end{bmatrix} \quad (13)$$

with the $\tilde{\Phi}$ related to θ_{robot} in Ω being set to 0. σ^2 is the observation variance.

Given likelihood $p(\theta | Y^*)$, the $p(Y^*, \theta) = p(\theta | Y^*) p(\theta)$ is another joint Gaussian distribution:

$$p(Y^*, \theta) = \mathcal{N} \left(\begin{bmatrix} F^* \\ \theta \end{bmatrix} \middle| \begin{bmatrix} \Omega \theta \\ \mu_\theta \end{bmatrix}, \begin{bmatrix} A & \Sigma_\theta \Omega^T \\ \Omega \Sigma_\theta^T & \Sigma_\theta \end{bmatrix} \right) \quad (14)$$

with $A = \sigma^2 I + \Omega \Sigma_\theta \Omega$. The mean and variance of conditional distribution $p(\theta | Y^*)$ is therefore derived as:

$$\begin{aligned} \mu_{\theta | Y^*} &= \mu_\theta + \Sigma_\theta \Omega^T A^{-1} (F^* - \Omega \mu_\theta), \\ \Sigma_{\theta | Y^*} &= \Sigma_\theta - \Sigma_\theta \Omega^T A^{-1} \Omega \Sigma_\theta. \end{aligned} \quad (15)$$

After obtaining θ , the robot motor skills are operated by running DMPs with parameter vector θ_{robot} with estimated phase x^* .

3.3 Environment-adaptive interaction primitives

An issue with IPs is that in situations where the initial human partner observation $[\mathbf{y}_1^*, \dots, \mathbf{y}_{T'}^*]^T$ is too ambiguous for reliable DMP parameter inference. The system can either wait for more distinctive partner activity which may result in an unnaturally long pause before the robot’s response commences, or risk prematurely executing unsafe or otherwise undesirable actions.

To reduce uncertainty, EaIPs introduce environment parameters e representing features embodying task-critical environmental properties into IPs along with human observation trajectories $p(\mathbf{y}^*)$, i.e. computing a joint distribution $p(Y^*, \theta, e)$. Depending on context, the contents of e can vary widely from physical sizes of objects or perceived obstructions; these parameters can also be provided from models built on the same training data for the underlying IPs or from heuristics. However in this work we propose object detection data, to encourage generalization in tasks focusing on physical objects.

For recording trajectories with these environment parameters, a new training sample is defined as Y_e with an additional DOF for $E = [e^1, \dots, e^S]^T$:

$$Y_e = [Y_{human}, Y_{robot}, E] = \begin{bmatrix} \mathbf{y}_1^1 & \dots & \mathbf{y}_N^1 & e^1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{y}_1^S & \dots & \mathbf{y}_N^S & e^S \end{bmatrix}. \quad (16)$$

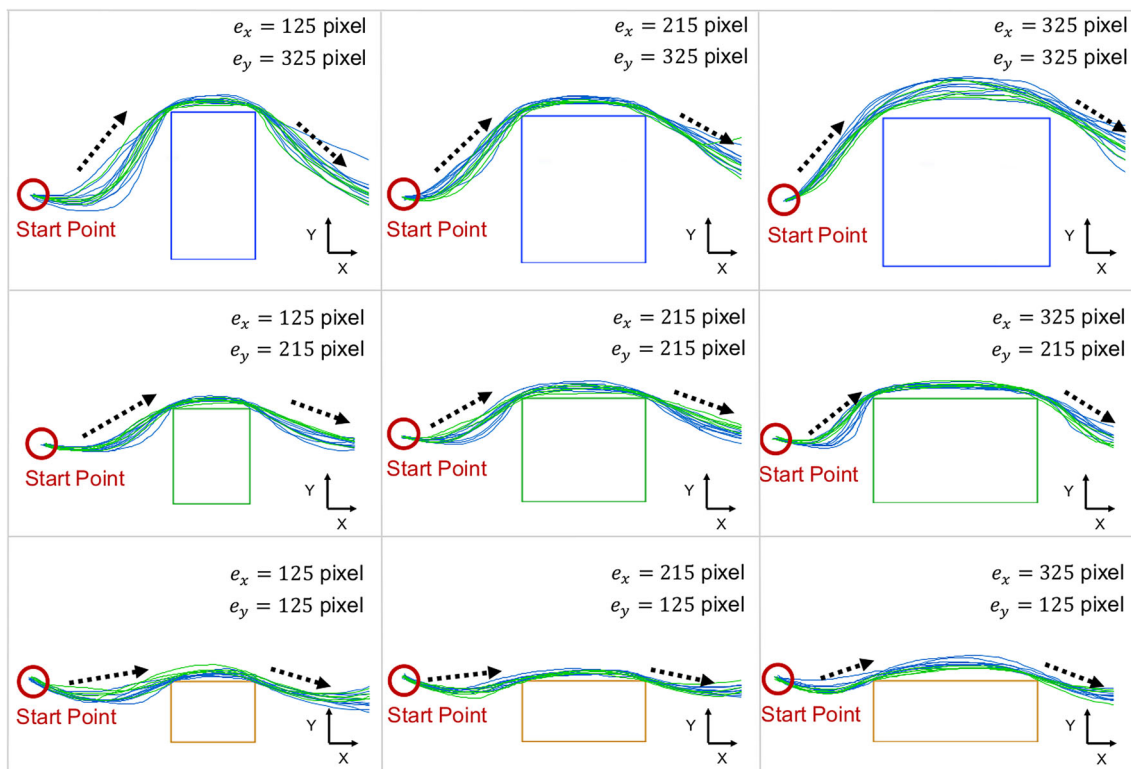


Fig. 2 Training data for simulation experiment of passing over rectangular objects. Training paths are shown in blue, and test paths are shown in green (Color figure online)

Applying DMPs to learn θ from Y_e , the environmental weight vector w_e is obtained from the likelihood $p(e|\theta)$ via least squares regression:

$$p(e|\theta) \sim \mathcal{N}(e|\theta w_e, \sigma_e^2 I). \tag{17}$$

Given t^* steps observing trajectories Y^* and environmental parameter e^* , we combine the observing forcing function and environmental parameter to $F_e^* = [f_1^*, \dots, f_N^*, e^*]^T$ and get a distribution similar to Eq. (14):

$$\mathcal{N}\left(\begin{bmatrix} F_e^* \\ \theta \end{bmatrix} \middle| \begin{bmatrix} \Omega_e \theta \\ \mu_\theta \end{bmatrix}, \begin{bmatrix} A_e & \Sigma_\theta \Omega_e^T \\ \Omega_e \Sigma_\theta^T & \Sigma_\theta \end{bmatrix}\right) \tag{18}$$

$$\Omega_e \theta = \begin{bmatrix} \tilde{\phi} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \\ w_e \end{bmatrix} \begin{bmatrix} w_1 \\ g_1 \\ \vdots \\ w_N \\ g_N \end{bmatrix}, \tag{19}$$

$$A_e = \begin{bmatrix} \sigma^2 & \dots & 0 & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \sigma^2 & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & \sigma_e^2 \end{bmatrix} + \Omega_e \Sigma_\theta \Omega_e. \tag{20}$$

The mean and variance of the distribution $p(\theta|Y_e^*)$ is then derived by plugging Eqs. (19, 20) into Eq. (15). The parameter σ_e in Eq. (20) is defined as the variance matrix of environmental parameters. σ_e should be suitably defined to capture the magnitude of the observed parameters’ noise.

4 Experimental results

We evaluate EaIPs in the task of covering large objects with a plastic bag collaboratively alongside a human partner. This is first done in a toy simulation of the exercise in Sect. 4.1, followed by experiments with a Baxter humanoid robot in Sect. 4.2.

4.1 Simulation results

Performing this task in simulation successfully is defined as neatly passing over a rectangular 2D object of varying size. We expand the setting from Cui et al. (2016) to nine objects with different width and height, i.e., two environmental parameters $e_x, e_y \in [125, 215, 325]$ pixels, are made available. As shown in Fig. 2, ten training trajectories (blue) and five testing (green) trajectories (each with 200 steps) are recorded from 2D mouse cursor movement across a GUI.

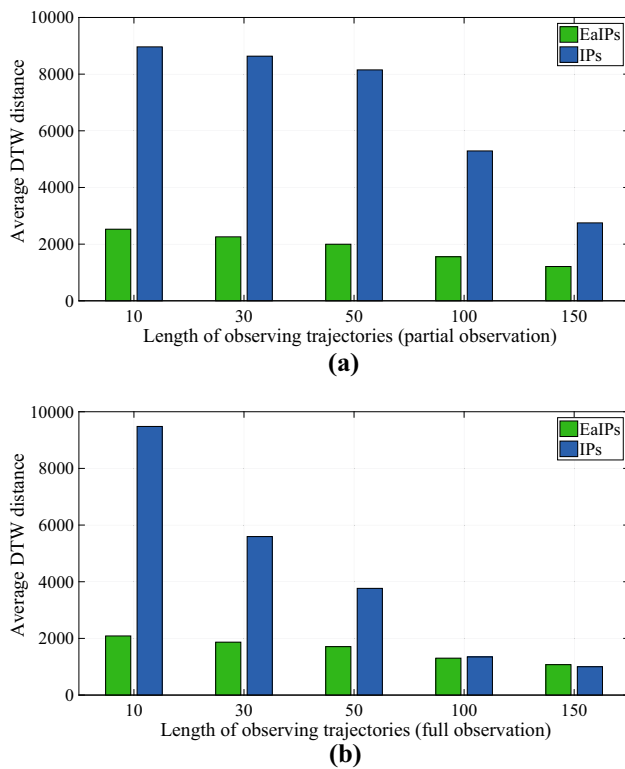


Fig. 3 Dynamic Time Warping distance (unitless) of IPs and EaIPs predictions to training samples in the simulation task. A smaller DTW distance indicates a better correlation between time series. **a** DTW distance, given observation of horizontal image axis only. **b** DTW distance, given observation of both image axes

The DTW distance is taken as an error metric to assess comparative performance in two different scenarios: a partial observation (horizontal movements only) and a full observation (both horizontal and vertical movements) over five testing trajectories with varying lengths of observed trajectories for each object. Results are shown in Fig. 3. According to the examples of both EaIPs and IPs with different length observation in two scenarios (Fig. 4), EaIPs yielded a much reduced DTW distance to training samples when given a short observation period while IPs only performing comparably when given lengthy (≥ 100 steps) observations along both horizontal and vertical image axes (Fig. 4b). When only provided with observations along the horizontal image axis (Fig. 4a) EaIPs were able to maintain similar performance to observing both axes due to the consideration of additional parameters, whereas the prediction accuracy of IPs was diminished. This result indicates that EaIPs can work with short ambiguous observations (even only along the horizontal axis), while IPs could not even if the partially observed action is more informative.

It can be seen in Fig. 5 that EaIPs are able to generate suitable trajectories even when provided with very little observation to cover objects with novel environmental

parameters ([400, 50] pixels in both axes). This demonstrates their ability to capture an underlying ‘style’ of the motion to be undertaken (Matsubara et al. 2011) and to accomplish the task when augmented by environmental parameters.

The performance of EaIPs with noisy and biased environmental parameters is further evaluated in an offline manner. 500 trajectories are planned by EaIPs with only ten steps along either axis provided as observation, while varying levels of 0-mean Gaussian noise are added to the environmental parameters as shown in Fig. 6. With a suitable setting of σ_e , the planned trajectories still outperform IPs. The average DTW distances are shown in Fig. 7. It can be seen that by setting a larger σ_e , the performance of EaIPs with uncertain environmental parameters approaches that of EaIPs with noise-free environmental parameters. These results indicate EaIPs’ ability to handle environmental observations with significant associated uncertainty.

4.2 Real robot experimental results

EaIPs are further evaluated with a Baxter research robot (Fig. 9). Section 4.2.1 details the CNN-based extraction of environmental parameters, followed by an overview of the experimental setup in Sect. 4.2.2.

4.2.1 Environmental parameters from object detection

For the real task, visual contextualization takes the form of the label and bounding box properties of the object serving as the focus of the interactive exercise; these serve as E in Eq. (16). This information is obtained from the YOLO v2 (You Only Look Once) (Redmon and Farhadi 2017) CNN-based object detector (Fig. 8). YOLO is particularly well suited to process scene information for EaIPs due to its capability to detect objects in real-time from a video stream, as its runtime complexity is far less than that of competing architectures such as VGG-16 (Simonyan and Zisserman 2014) while its bounding boxes and object labels possess some correlation to the physical space that EaIPs operate in. YOLO processes images as grids, with each grid cell providing estimates of class labels and bounding boxes of objects that may reside within/around it. These estimates are then combined into an overall probability map based on each cell’s likeliest label, from which final object detections are drawn.

Although suitable environmental parameters can indeed be also drawn from other information sources such as depth cameras or additional sensory apparatus, we maintain it is less cumbersome to use a vision-based object classifier. As training data is often available in a limited quantity for EaIPs modeling, models built for parameter regression upon such data would easily be overfitting, and it is also undesirable to have to craft heuristics for each task that EaIPs are to be applied to. Since classifiers such as YOLO can be exposed to

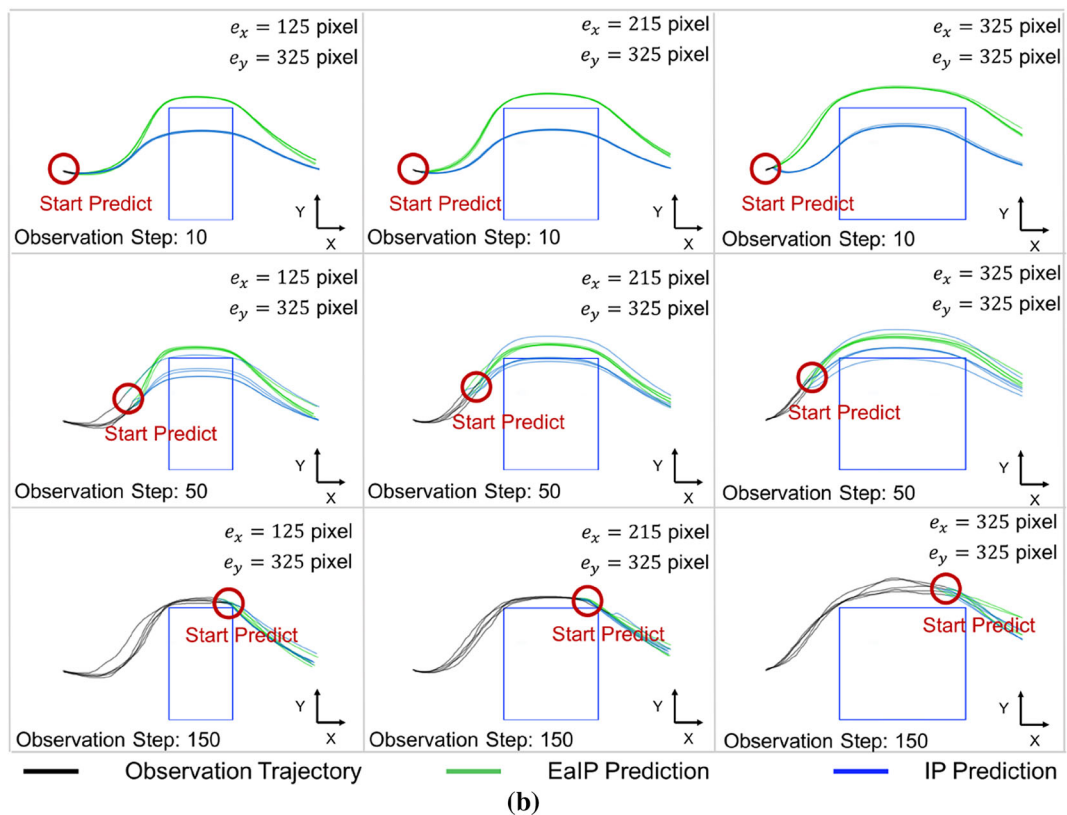
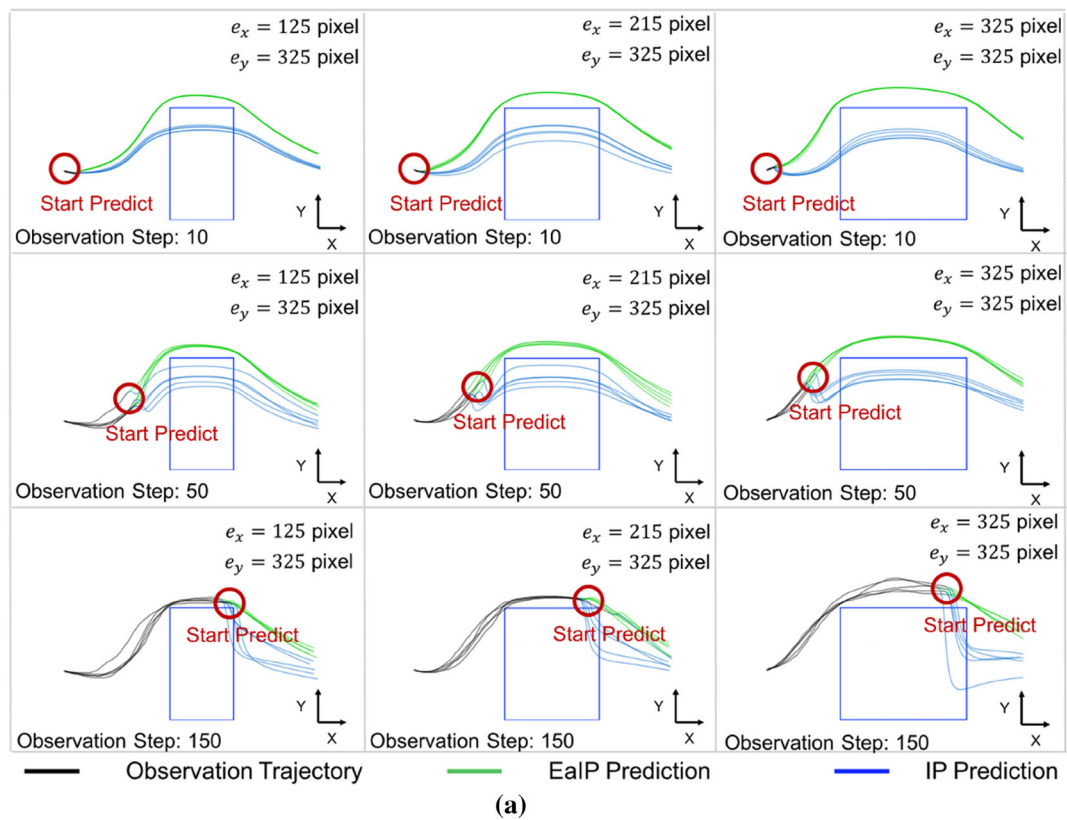


Fig. 4 Comparison of performance between IPs and EaIPs in the simulated task of passing over rectangular objects. **a** Given observation of horizontal image axis only. **b** Given observation of both image axes

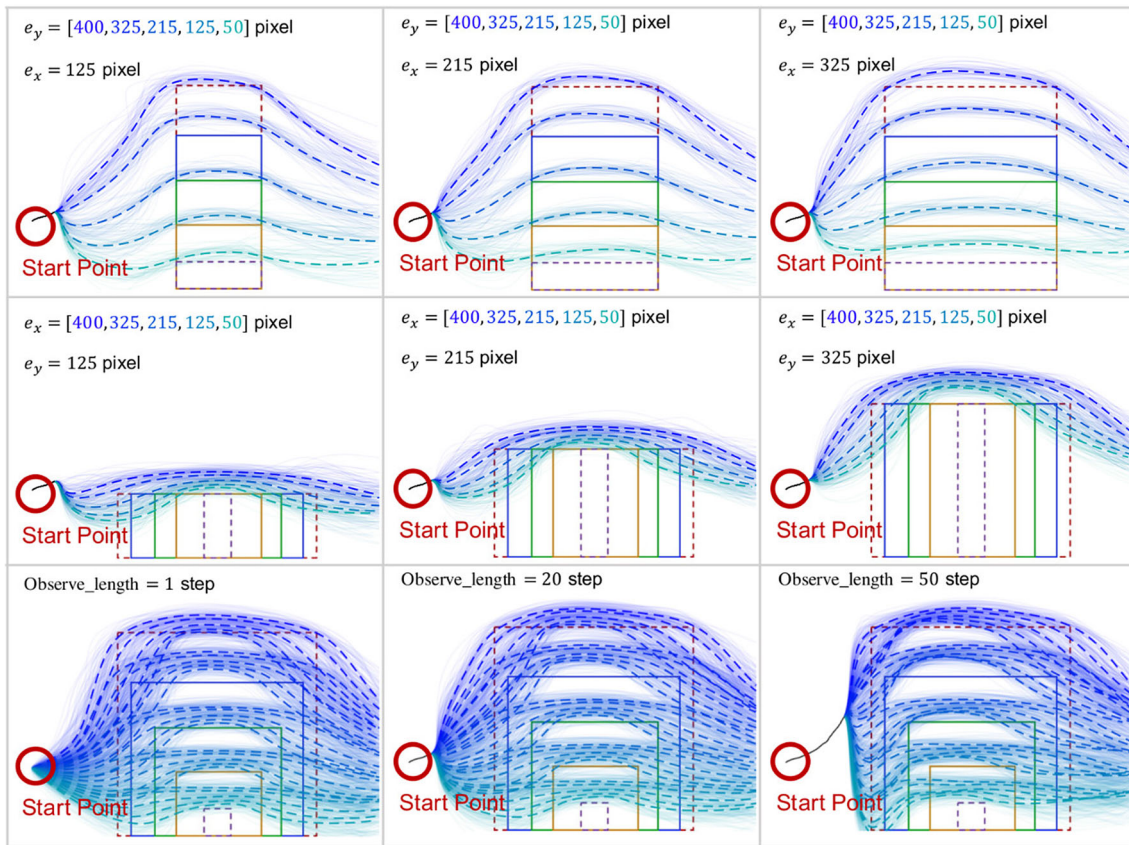


Fig. 5 Testing of EaIPs in passing over rectangular objects. Dashed trajectories indicate the mean of numerous solid trajectories generated for each object, with novel objects indicated by dashed borders

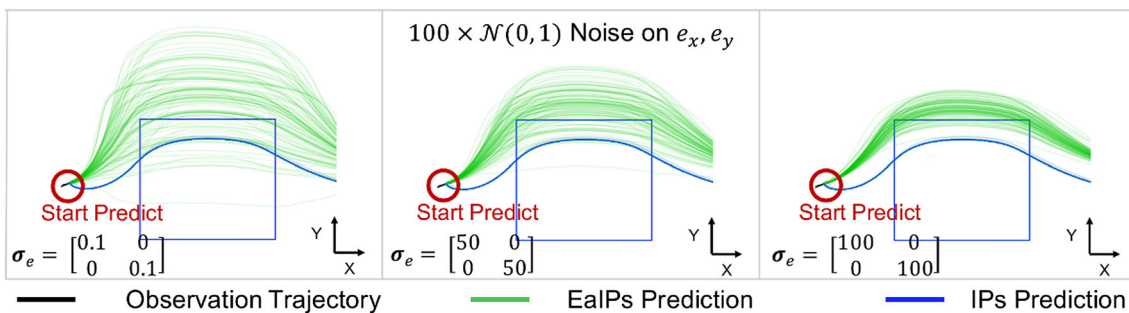


Fig. 6 Testing of EaIPs in passing over rectangular objects when varying levels of Gaussian noise are added to the environmental parameters

a far broader range of objects than in the EaIPs training set, its features immediately allow the EaIPs to generalize to new objects and can be used directly as environmental parameters without the need for task-specific modulation.

4.2.2 Baxter research robot experimental results

Three overhead Kinects provide visualization of the interaction scene from left, overhead, and right views. Following extrinsic calibration, their respective point cloud data can then be interpreted in the robot’s co-ordinate frame. Partner

observation [Y_{human} in Eq. (16)] consists of the Cartesian positions of both hands with green color gloves in the point cloud. The five dimensional environmental parameter vector E comes from YOLO processing RGB image data from the right-most Kinect (Kinect 3 in Fig. 9), and contains a numeric object class label (chair, stepladder, table and bookshelf), and the centroid and size of the estimated bounding box in image space to capture some correlation to physical space. To perform its share of the activity, the robot held a point along the opening of a large bag stationary in its right gripper, while its left swept over the object to be covered while holding another

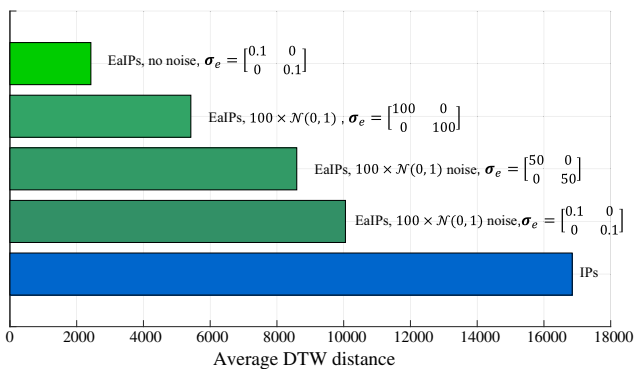


Fig. 7 Dynamic Time Warping distance (unitless) of EalPs’ prediction results to training samples in the simulation task, when Gaussian noise is added to the environmental parameters

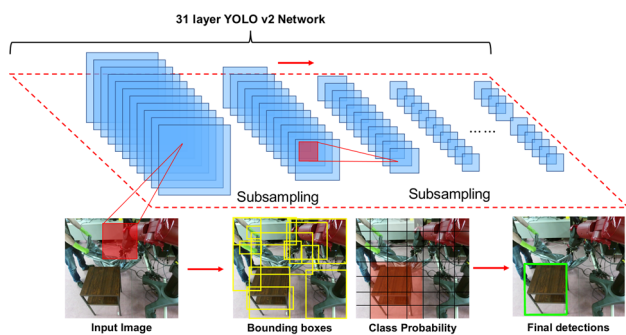


Fig. 8 Deep-learning CNN topology for object detection. YOLO’s network yields object label and bounding box estimates

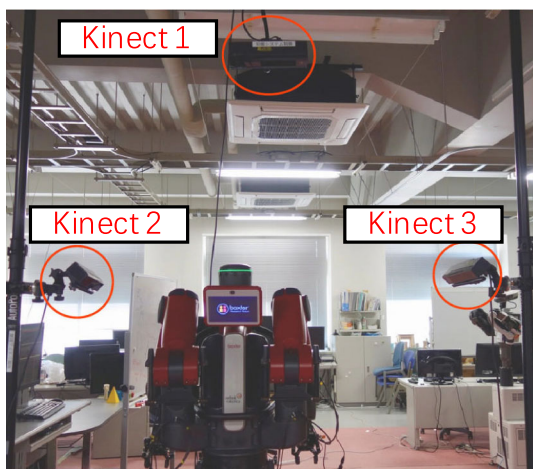


Fig. 9 NAIST Baxter platform with 3 overhead Kinect V2 sensors (red ellipses) (Color figure online)

part of the bag’s lip. The generated robot action in each step is defined as a three dimensional vector including the baxter’s left arm end-effector’s position. The training objects used in our experiment are shown in Fig. 10. Training data was generated by kinesthetic operation (Argall et al. 2009) of the robot’s left arm in tandem with the interaction partner, with



(a) (b)



(c) (d)

Fig. 10 Training object set. **a** Stepladder, **b** folding table, **c** office chair, **d** brown bookshelf



(a) (b)



(c) (d)

Fig. 11 Test object set. **a** Yellow bookshelf, **b** end-table, **c** 4-legged chair, **d** 4-legged table (Color figure online)

ten samples recorded for each training object. Data communication and logging was managed via the Robot Operating System (Quigley et al. 2009) middleware.

Fig. 12 Workflow of the real experiment (Color figure online)

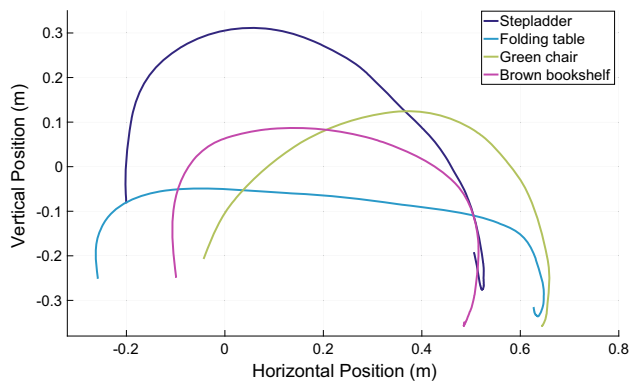
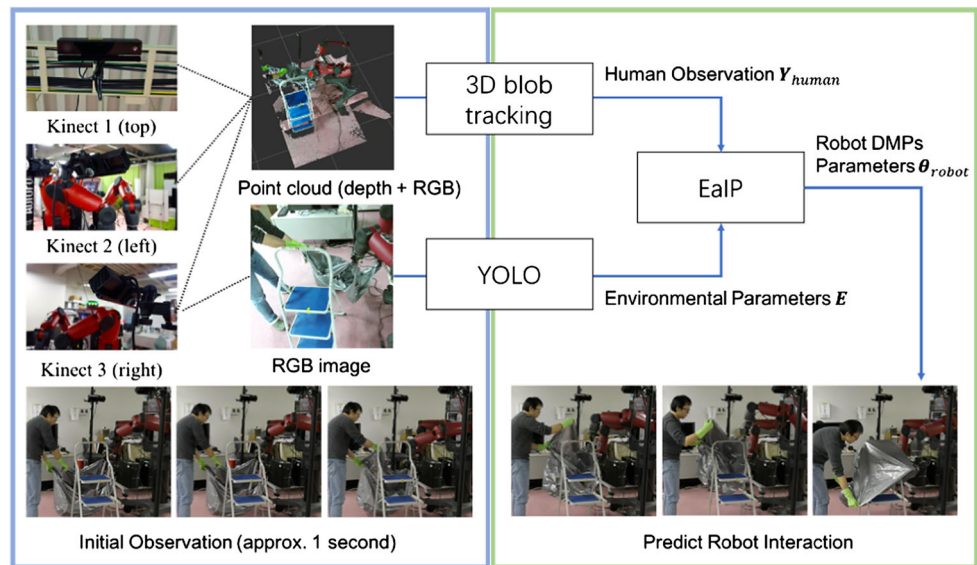


Fig. 13 Mean training data of the Baxter's left end-effector, from ten samples for each training object

To ensure object recognition, a 31-layer YOLO network was trained upon 300 manually labelled RGB images of the four training objects in Fig. 10 as seen by Kinect 3.

EaIPs were run¹ with four new objects (Fig. 11) and with partners that were not present during the collection of training data. As shown in Fig. 12, the workflow of this experiment is to first observe the human action Y_{human} via RGB-XYZ point cloud data, and extract E from the RGB scene perceived by Kinect 3. EaIPs then generate a collaborative robot action. Figure 14 shows the left end-effector paths from both IPs and EaIPs from a frontal view of the robot's YZ plane, for all eight objects when given a 1 s observation period (approximately 10% of average task duration). It can be seen that EaIPs better preserve the path structure from training samples (Fig. 13), since IPs' paths possess very little variation between objects. The paths from IPs (Fig. 14a) resulted in collisions with the

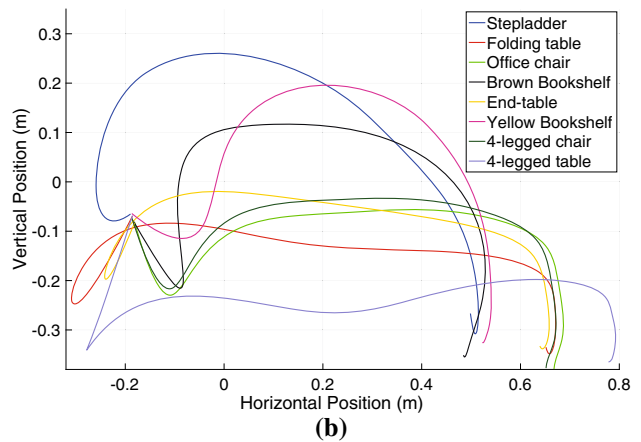
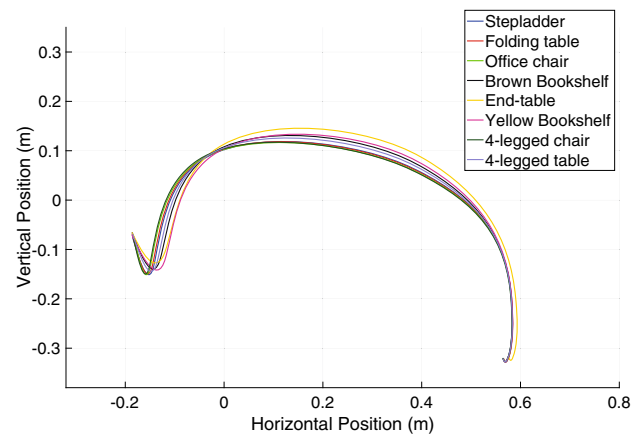


Fig. 14 Baxter left end-effector paths from IPs and EaIPs. **a** Paths from IP, **b** paths from EaIP

¹ Video available at <https://youtu.be/x872ILZ9MEc>.

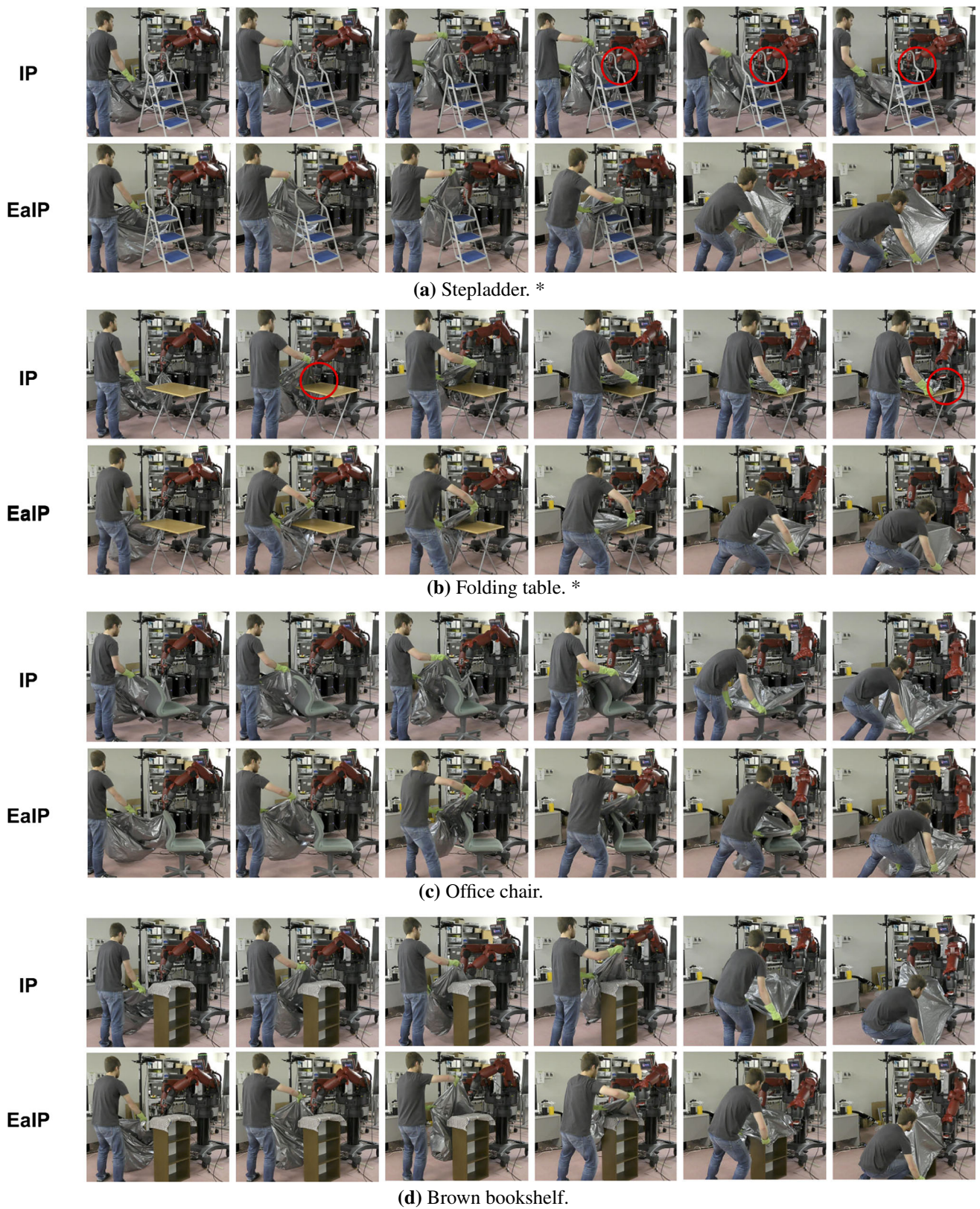


Fig. 15 Behavior of IPs and EalPs on the training set. * indicates the object resulted in collision (red ellipses) (Color figure online)

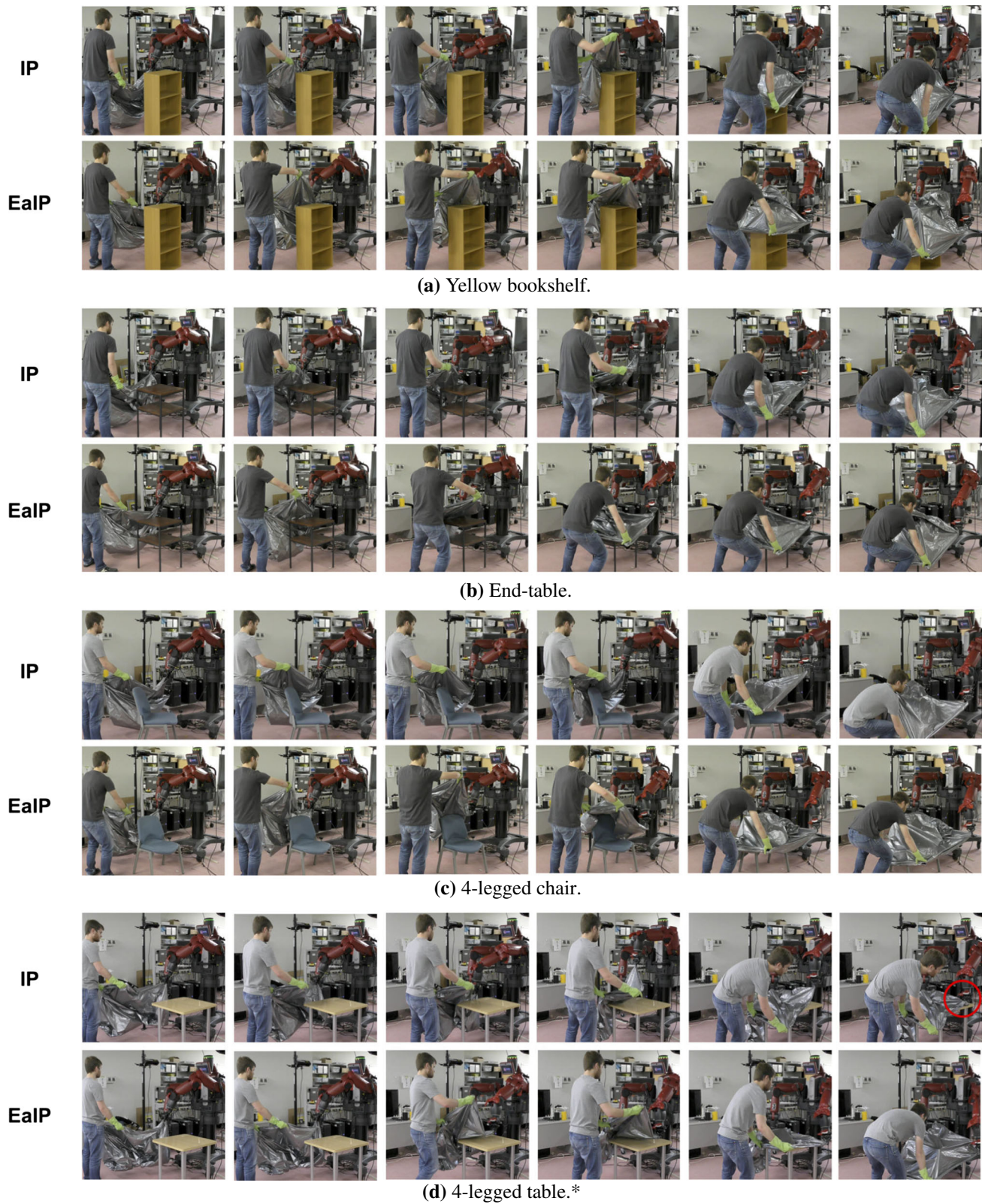


Fig. 16 Behavior of IPs and EalPs on the novel object set. * indicates the object resulted in collision (red ellipses) (Color figure online)

Table 1 Classification Accuracy (CA) of YOLO in real experiment, for 0–50 and 0–100% task duration

Object	CA (first 50%)	CA (100%)
Stepladder	186/186 (100%)	293/371 (78.98%)
Folding table	194/194 (100%)	337/388 (86.86%)
Office chair	134/134 (100%)	258/267 (96.63%)
Brown bookshelf	130/130 (100%)	211/261 (80.84%)
Yellow bookshelf	193/193 (100%)	335/386 (86.79%)
End-table	224/224 (100%)	382/448 (85.27%)
4-legged chair	185/185 (100%)	309/370 (83.51%)
4-legged table	137/137 (100%)	220/274 (80.29%)

three larger objects as shown in Figs. 15 and 16, whereas EaIPs were able to cover all eight objects successfully.

The detection and generalization abilities of the trained YOLO network are investigated with 1968 RGB images recorded from Kinect 3 during experimentation. As seen in Table 1, the classification accuracy (CA) decreases from 100% to around 80% during the second half of the task duration when the plastic bag partially obscures the objects. However since these classification failures happen long after the observation period had ended, they would have no negative impact on EaIPs' performance. Hence the object detection component utilized here is sufficiently reliable for the scope of this exercise.

5 Discussion

In the real experiment, paths sufficiently close to the 100% partner observation can be obtained by EaIPs given as little as 10% of partner observation according to Fig. 17 while IPs would require nearly 70% of partner observation to reach a comparable RMSE as shown in Fig. 17b. The consideration of environmental parameters can thus greatly enhance the robustness of EaIPs, which is beneficial in cases where observed partner behaviour would result in unsafe actions being planned. Even if the full partner observation in Fig. 15a were to be provided as an observation to IPs, it would have still resulted in unsafe robot motion as shown in Fig. 17a due to similarity with training samples with smaller objects.

It is possible to run inferences continuously throughout the duration of the interaction, however as seen in Fig. 17a, paths from EaIPs do not significantly change as more information becomes available while IPs may benefit as the partner's movement becomes more defined. However depending on other conditions, such as those which would be suitable for representation in EaIPs' environmental parameter set, an extended observation of potentially ambiguous partner movement may still be inadequate for ensuring success even

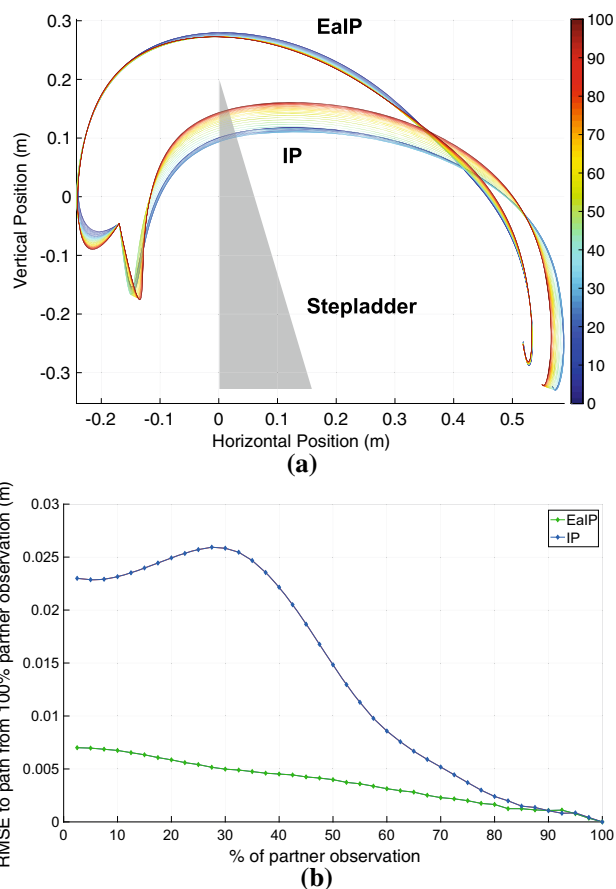


Fig. 17 Top: convergence of a-posteriori paths given partner observations of varying duration for IPs and EaIPs in the stepladder covering exercise (Fig. 15a). Bottom: RMSE to their respective 'final' paths generated under 100% partner observation. For both figures, a minimum observation of 10 steps (<3%) is taken to avoid severely premature inferences. **a** Paths from IPs and EaIPs under increasing partner observation (% in the colorbar), with the stepladder visualized in grey, **b** RMSE of the above to IP/EaIPs' respective final paths (Color figure online)

if the partner's behavior is indeed suitable from their side of the task.

The ability of the EaIPs framework to leverage increasingly robust object detection methodologies from the vision community (Stallkamp et al. 2012; Google 2017) can allow for improved adaptability towards novel objects not seen in training data. At present this is particularly important in demonstration-based systems, as training data would otherwise be inhibitive expensive to obtain in adequate volumes. Our results show that even with a limited quantity of training data, the use of a vision-based object detector was sufficient for the planning of robust robot actions. Whereas our earlier work relied upon ground-truth features to conceptually validate EaIPs, here we present the framework in its entirety; capable of modeling human–robot object-focused tasks from observations of partner and scene.

For future work we intend to utilize the object class label to a greater extent and further leverage bounding box information (e.g. 3D bounding boxes from distributed RGB cameras) and introduce the depth information in object detection in order to handle EaIPs for higher level task abstraction that require more accurate environmental information, e.g., human–robot cooperation in a dynamical work space with several objects occluding each other. The nature of the object in question would serve as a clearer discriminating factor than partner behavior as in the work by Maeda et al. (2017), which solely relies upon differences in partner observations to determine a response. Vision-based approaches to partner observation (Cao et al. 2017) may allow for these kinds of collaborative activities without the need for 3D sensing, as task parameters can be drawn through similar means. Removing reliance upon hardware such as the Kinect will allow for more versatile hardware configurations, as well as operation in traditionally less robot-friendly conditions such as open sunlight.

6 Conclusion

This work presents an extension to the interaction primitives framework that enhances fluency in physical human–robot interaction. Task parameters from a CNN object detector, consisting of class labels and bounding boxes in image space, allow for the complete EaIPs modeling of a collaborative human–robot task from observations of partner and scene. The correlation of these augmentative parameters to physical space make them naturally better suited to this problem than for example, abstract dimensionality reduction features which would easily be overfitting.

Experimental results in the joint task of sweeping a large plastic bag over bulky objects, in both simulation and with a humanoid Baxter robot, show an increased robustness to ambiguity in partner activity compared to interaction primitives. We aim to develop this framework further to consider visual partner observation, as well as the leveraging of object label information for more complex interactive activities.

Acknowledgements We gratefully acknowledge support from the New Energy and Industrial Technology Development Organization (NEDO), and UTS' FEIT HDR Students Research Collaboration Experience. We also thank Mr. Haifeng Han and Mr. Guillaume Fournet for their assistance with experimentation.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5), 469–483.
- Awais, M., & Henrich, D. (2013). Human–robot interaction in an unknown human intention scenario. In *International conference on frontiers of information technology (FIT)* (pp 89–94).
- Ben Amor, H., Neumann, G., Kamthe, S., Kroemer, O., & Peters, J. (2014). Interaction primitives for human-robot cooperation tasks. In *IEEE international conference on robotics and automation (ICRA)* (pp. 2831–2837). IEEE.
- Cao, Z., Simon, T., Wei, S., & Sheikh, Y. (2017). Realtime multi-person 2D pose estimation using part affinity fields. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1302–1310).
- Cui, Y., Poon, J., Matsubara, T., Valls Miro, J., Sugimoto, K., & Yamazaki, K. (2016). Environment-adaptive interaction primitives for human–robot motor skill learning. In *IEEE-RAS international conference on humanoid robots (Humanoids)* (pp. 711–717).
- Doumanoglou, A., Kargakos, A., Kim, T. K., & Malassiotis, S. (2014). Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning. In *IEEE international conference on robotics and automation (ICRA)* (pp. 987–993).
- Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., & Abbeel, P. (2016). Deep spatial autoencoders for visuomotor learning. In *IEEE international conference on robotics and automation (ICRA)* (pp. 512–519).
- Fitzgerald, T., Goel, A., & Thomaz, A. (2015). A similarity-based approach to skill transfer. In *Women in robotics workshop at the robotics: Science and systems conference*.
- Goil, A., Derry, M., & Argall, B. D. (2013). Using machine learning to blend human and robot controls for assisted wheelchair navigation. In *IEEE international conference on rehabilitation robotics (ICORR)* (pp. 1–6).
- Google. (2017). Google cloud vision api. <https://cloud.google.com/vision>
- Huang, C. M., & Mutlu, B. (2016). Anticipatory robot control for efficient human–robot collaboration. In *ACM/IEEE international conference on human–robot interaction (HRI)* (pp. 83–90).
- Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P., & Schaal, S. (2013). Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Computation*, 25(2), 328–373.
- International Federation of Robotics. (2015). *World robotics 2015 industrial robots*. Technical report, International Federation of Robotics, Germany.
- Kormushev, P., Calinon, S., & Caldwell, D. G. (2010). Robot motor skill coordination with em-based reinforcement learning. In *2010 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3232–3237).
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 25, pp. 1097–1105). Curran Associates, Inc. <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- Kronander, K., & Billard, A. (2014). Learning compliant manipulation through kinesthetic and tactile human–robot interaction. *IEEE Transactions on Haptics*, 7(3), 367–380.
- Kruse, D., Radke, R. J., & Wen, J. T. (2015). Collaborative human–robot manipulation of highly deformable materials. In *IEEE international conference on robotics and automation (ICRA)* (pp. 3782–3787).

- Lawitzky, M., Mortl, A., & Hirche, S. (2010). Load sharing in human–robot cooperative manipulation. In *IEEE international conference on robot and human interactive communication (RO-MAN)* (pp. 185–191).
- Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.
- Lioutikov, R., Kroemer, O., Maeda, G., & Peters, J. (2016). *Learning manipulation by sequencing motor primitives with a two-armed robot* (pp. 1601–1611). Cham: Springer.
- Maeda, G., Ewerton, M., Neumann, G., Lioutikov, R., & Peters, J. (2017). Phase estimation for fast action recognition and trajectory generation in human–robot collaboration. *The International Journal of Robotics Research (IJRR)*, 36, 1579–1594.
- Mandery, C., Borrás, J., Jochner, M., & Asfour, T. (2016). Using language models to generate whole-body multi-contact motions. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 5411–5418).
- Matsubara, T., Hyon, S. H., & Morimoto, J. (2011). Learning parametric dynamic movement primitives from multiple demonstrations. *Neural Networks*, 24(5), 493–500.
- Patel, M., Miro, J. V., Kragic, D., Ek, C. H., & Dissanayake, G. (2014). Learning object, grasping and manipulation activities using hierarchical hmms. *Autonomous Robots*, 37(3), 317–331.
- Pervez, A., Mao, Y., & Lee, D. (2017). Learning deep movement primitives using convolutional neural networks. In *IEEE-RAS international conference on humanoid robots (humanoids)* (pp. 191–197).
- Pinto, L., & Gupta, A. (2016). Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *IEEE international conference on robotics and automation (ICRA)* (pp. 3406–3413).
- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., et al. (2009). ROS: An open-source robot operating system. In *ICRA workshop on open source software, Kobe* (Vol. 3, p. 5).
- Redmon, J., & Farhadi, A. (2017). Yolo9000: Better, faster, stronger. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1063–6919).
- Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1), 43–49.
- Sheng, W., Thobbi, A., & Gu, Y. (2015). An integrated framework for human–robot collaborative manipulation. *IEEE Transactions on Cybernetics*, 45(10), 2030–2041.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. CoRR [arXiv:1409.1556](https://arxiv.org/abs/1409.1556).
- Soh, H., & Demiris, Y. (2015). Learning assistance by demonstration: Smart mobility with shared control and paired haptic controllers. *Journal of Human-Robot Interaction*, 4(3), 76–100.
- Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32, 323–332.
- Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction*. Cambridge: MIT Press.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1–9).
- Taha, T., Miro, J. V., & Dissanayake, G. (2011). A pomdp framework for modelling human interaction with assistive robots. In *IEEE international conference on robotics and automation (ICRA)* (pp. 544–549).
- Ude, A., Gams, A., Asfour, T., & Morimoto, J. (2010). Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Transactions on Robotics*, 26(5), 800–815.
- Vircikova, M., Smolar, P., & Sincak, P. (2012). Current trends in human–robot interaction: Towards collaborative & friendly machines. In *12th scientific conference of young researchers*.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Yunduan Cui was born in China in 1990. He is currently undertaking a research assistant professor at Nara Institute of Science and Technology, Japan. He received Ph.D. in information science from Nara Institute of Science and Technology, Japan in September 2017, M.E in computer science from Doshisha University, Japan in September 2014, and the B.E. in Electronic Engineering from Xidian University, China in 2012. His research interests are machine learning and control theory of robotics, especially reinforcement learning in robot control.



James Poon received the Bachelor of Mechanical and Mechatronic Engineering from the University of Technology Sydney, Australia in 2013. Since 2014 he has been a Ph.D. student in UTS' Centre for Autonomous Systems lab. His research interests include assistive robotics, learning from demonstration and human–robot collaboration.



Jaime Valls Miro received his B.Eng. and M.Eng. in Computer Science (Systems Engineering) from the Valencia Polytechnic University (UPV, Spain), in 1990 and 1993 respectively. He received his Ph.D. in robotics and control systems from Middlesex University (UK) in 1998, and worked in the underwater robotics industry as a software and control systems analyst until 2003. In 2004 he joined the Centre for Autonomous Systems in UTS (Australia), where he is currently an Associate Professor. His areas of interest span across the field of robotics, most notably modelling sensor behaviours for perception and mapping, computational Intelligence in HRI—assistive robotics in particular, and robot navigation.



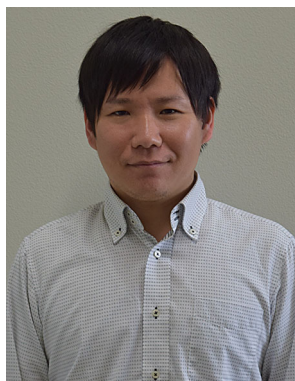
Kimitoshi Yamazaki was born in Japan in 1979. He received B.E., M.E., and Ph.D. degrees from University of Tsukuba in 2002, 2004, and 2007, respectively. From 2006 to 2007, he was a JSPS research fellow. From 2007 to 2012, he was a project assistant professor in the University of Tokyo. Since 2010, he is also a researcher of PRESTO, JST. Since 2012, he has been an assistant professor in Shinshu University. His current research interests are in robot vision, and motion planning. He is a member

of RSJ, SICE, JSME and IEEE.



Professor Kenji Sugimoto received the M.S. and Ph.D. degrees from Kyoto University in 1982 and 1989, respectively. After working with Mitsubishi Electric Corporation, he became an Assistant Professor with Kyoto University in 1985. Then he became an Associate Professor with Okayama University and Nagoya University. Since 1999, he has been a Professor with the Nara Institute of Science and Technology. His current research interests include control theory and system science. Prof.

Sugimoto is a member of IEEE, SICE, and ISCI.



Takamitsu Matsubara received his B.E. in Electrical and Electronic Systems Engineering from Osaka Prefecture University, Osaka, Japan, in 2003, M.E. in Information Science from the Nara Institute of Science and Technology, Nara, Japan in 2005, and a Ph.D. in Information Science from the Nara Institute of Science and Technology in 2007. From 2005 to 2007, he was a research fellow (DC1) of the Japan Society for the Promotion of Science. He is currently an Associate Professor

at the Nara Institute of Science and Technology and a visiting researcher at the ATR Computational Neuroscience Laboratories, Kyoto, Japan. His research interests are machine learning and control theory for robotics.