



REVIEW

Review of the techniques used in motor-cognitive human-robot skill transfer

Yuan Guan | Ning Wang | Chenguang Yang

Bristol Robotics Laboratory, University of the West of England, Bristol, UK

CorrespondenceChenguang Yang, Bristol Robotics Laboratory, University of the West of England, Bristol, UK.
Email: cyang@ieee.org**Funding information**

Engineering and Physical Sciences Research Council, Grant/Award Number: EP/S001913

Abstract

A conventional robot programming method extensively limits the reusability of skills in the developmental aspect. Engineers programme a robot in a targeted manner for the realisation of predefined skills. The low reusability of general-purpose robot skills is mainly reflected in inability in novel and complex scenarios. Skill transfer aims to transfer human skills to general-purpose manipulators or mobile robots to replicate human-like behaviours. Skill transfer methods that are commonly used at present, such as learning from demonstrated (LfD) or imitation learning, endow the robot with the expert's low-level motor and high-level decision-making ability, so that skills can be reproduced and generalised according to perceived context. The improvement of robot cognition usually relates to an improvement in the autonomous high-level decision-making ability. Based on the idea of establishing a generic or specialised robot skill library, robots are expected to autonomously reason about the needs for using skills and plan compound movements according to sensory input. In recent years, in this area, many successful studies have demonstrated their effectiveness. Herein, a detailed review is provided on the transferring techniques of skills, applications, advancements, and limitations, especially in the LfD. Future research directions are also suggested.

1 | INTRODUCTION

Similarly to the appearance of the robot discussed here, the world's first robot prototype, Unimate, was officially born in 1959. Only two years later, Unimate1900 series robots had been used as the first batch of automated mass production tools in factories. Since then, with a vision for the future, humans have embarked on a long journey, but at a surprisingly fast speed, to explore the potential of robots. Unquestionably, today, 60 years later, the boom in the robotics market is clear, behind which has been the active development of robotics technologies and artificial intelligence. So far, robots with skill learning have generally appeared in industrial manufacturing [1, 2], logistics [3], field robotics [4], surgery [5] and other fields. In addition to these common areas, robots are also expanding into nursing [6], human–robot collaboration [7], autonomous vehicles [8] and transfer learning [9] (i.e. sharing knowledge between agents with similar but discrepant mechanisms). It is not difficult to see that robots are continuously

finding new ways to impact people's lives, and robotics technologies have gradually begun to develop to enter for the trend of lightweight and collaborative areas, which could potentially allow more people to benefit from the safety and efficiency brought about by robots in the future.

In manufacturing, robotic surgery and various other robot applications, traditional robot programming is oriented toward specific skills, and engineers focus on the realisation of each task by hard-coding everything that needs to be executed. Traditional robots, while performing the task of 'bottle-to-cup water pouring', need to pay attention to the factors involving the position of the bottle/cup mouth and path planning. However, the task requirements in the actual scene are far from simple. When encountering challenges like obstacles, moving objects, and environmental changes, robots struggle to handle these adaptability challenges. These pre-programmed skills take engineers a long time to develop, and they may not be compatible with different task environments, as such demand is beyond their capacity [11]. Once the developed system starts to be

used, the robot's performance will not be enhanced any further, and any improvement will depend entirely on the engineer's knowledge of a specific task. In order to solve these problems, modularised skills, improved adaptation ability, and higher robot cognitive level are necessary. Moreover, in an era where robots are expected to perform a variety of personalised tasks, developing a useful skill transfer framework which is capable of endowing a robot with human-like motor and reasoning skills is their destiny.

The term learning from demonstration (LfD) or programming by demonstration (PbD) is sometimes also referred to as imitation learning or apprenticeship learning. Whatever it is called, it is an efficacious technique for skill transferring and reducing the complexity of search spaces for a robot trying to learn a skill [12]. Research into LfD started about 40 years ago; as its name suggests, LfD allows robots to learn a skill by analysing the skill performed by the demonstrator and then imitating the skill. This transforms the tiresome and complicated (even sometimes unmanageable) skill pre-programming process into a demonstrating (i.e. teaching) process, which can be implemented by laymen (i.e. people who do not have programming skills), and resembles the method of human–human interaction and social learning. LfD could be categorised as a supervised learning approach, assuming the experts' demonstrations are (sub-)optimal. The observed demonstrations direct the robot policy search to a local optimal area where refinement could be further applied.

Overall, these kinds of technologies that replace traditional programming have a broad range of applications; for example, remote underwater vehicle operation [13], reusable surgical skills [14], autonomous services robotics [15] and physical collaborative tasks [16], etc. Most importantly, LfD is ideal for highly customised/personalised tasks, as it reduces the time and labour cost of re-programming. A typical LfD human–robot skill transfer method involves three main steps, which are (1) demonstration, (2) modelling and learning, and (3) skill extraction and reproduction. Herein readers are guided in recapping the different outcomes, in recent years, that have made breakthroughs in the above three steps.

In [17], four primary issues were summarised that deal with imitation learning/LfD problems. Two issues raised, 'what to imitate' and 'how to imitate', are generally of interest to robotics researchers, and many solutions have been proposed and tested. When imitators learn a skill, a good learner knows clearly on a high-level what they want to imitate. The world we observe is composed of countless amounts of low-level and high-level information. Only a small portion of this information is essential in imitation. For example, when imitating table-wiping skills, agents need to pay attention to hand position and press strength, while speed information is usually irrelevant. The fundamental reason for a human being able to continuously improve his/her skill performance, based on observing his/her own skill completion, is the understanding of performance and the corresponding systematic evaluation metric. Then, the question of 'how to imitate' explores how to maximise the metric. This implies that imitators need to have sufficient high-level cognitive abilities to solve these problems.

The term 'cognition' refers to all psychological skills that can help learning, understanding, reasoning, and perception. Decision-making is a higher level cognitive process, and is also a consideration that focused on here. Artificial intelligence like machine learning (ML), deep learning (DL) and other data-driven technologies can achieve reasoning, planning, learning, decision-making and other functions to a certain extent [18]. After Gibson's theory of affordance [19], research on affordance learning also began. Affordance explained the mapping between objects, their affordable behaviours, and the effects of behaviours, enabling skill transfer and learning from a unique viewpoint [20]. Robots in the new era require more personalised skills and a more user-friendly programming interface. So far, most of the research into LfD has been limited to specific scenarios and skills. Due to the inadequate cognitive/decision-making capabilities of robots, most applications are not suitable for frequently changing task scenarios because they cannot reduce the load of reprogramming well (i.e. reprogramming is needed when the task is changed). In some current application scenarios, robots need to possess a great variety of skills and be able to evolve into novel ones by reorganising existing skills, rather than acquiring them from the human teacher each time. In addition to the realisation of learning individual (motor) skills, the text herein will also motivate readers to review and understand some research into cognitive skill transfer and LfD dealing with challenges in complex task scenarios, high-level decision-making and action-sequence planning.

A detailed overview of the critical techniques, advancements and problems, and developmental trends of robot–human skill transfer (i.e. in particular, LfD) in both high-level and low-level perspectives is provided. Section 3 introduces the works in LfD to realise low-level individual (motor) skill. Section 4 focuses on the high-level aspects which introduce cognition/decision, complex compound skills. Section 5 extracts the current works and shortcomings, putting forward constructive suggestions for future developments. Finally, Section 6 provides a summary.

2 | PRELIMINARIES

As one of the most comprehensive topics in robotics, human–robot skill transfer involves a lots of work of which some is solved and some remains unsolved. Figure 1 depicts the main issues that have been long attracting researchers' attention. This can be simplified by understanding the big picture using a few perspectives, as shown below.

- **Demonstration:** When tackling skill transfer problems, the first thing that needs to be specified is the formation of a skill. Depending on the modal choice, different sensors may be applied as parts of the demonstration interface. Through three main kinds of interface, the expert/failure demonstrations are captured. Each interface has its pros and cons, which are discussed later.
- **Learning low-level motor primitives:** Temporal alignment, motion trajectory segmentation, and recognition may be the

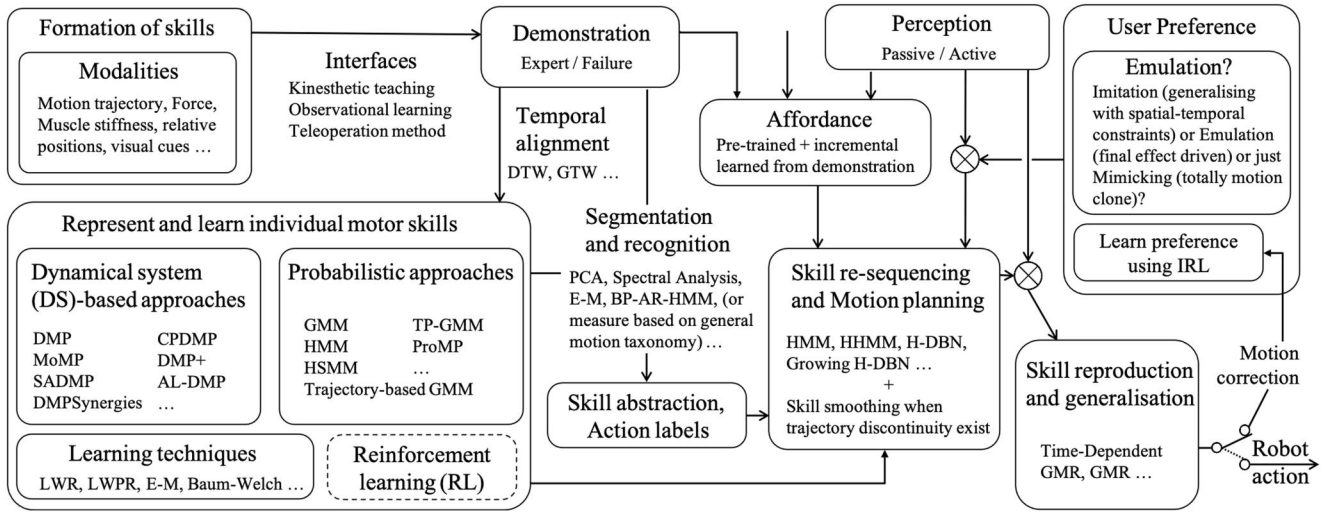


FIGURE 1 Overview of skill transfer problems. AL-DMP, Arc-length DMP; BP-AR-HMM, Beta-process Autoregressive Hidden Markov Model; CPDMP, Compliant Parametric DMP; DMP, Dynamic movement primitive; DTW, Dynamic Time Warping; E-M, Expectation-maximisation; GMM, Gaussian mixture model; GMR, Gaussian mixture regression; GTW, Generalised Time Warping; H-DBN, Hierarchical Dynamic Bayesian Network; HHMM, Hierarchical Hidden Markov Model; HMM, Hidden Markov model; HSMM, Hidden semi-Markov model; IRL, Inverse reinforcement learning; LWPR, Locally weighted projection regression; LWR, Locally weighted regression; MoMP, Mixture of Movement Primitive; PCA, Principal component analysis; SADMP, Style Adaptive DMP; TP-GMM, Task-parameterised GMM

prerequisites for learning individual sub-tasks, as people usually demonstrate a complex sequence of skill primitives. Two main types of primitive representation approaches can now be found: the dynamical system-based approach and the probabilistic approach. Each model usually has its own dedicated learning techniques that have been proved to work effectively. Sometimes, reinforcement learning can be additive to model learning, as it refines a skill that could then be performed by a robot, which will have a higher success rate.

- *High-level symbolic reasoning:* Being able to perform high-level reasoning is an improvement to robot cognition. From that, robots automatically recognise skill primitives and give them labels. The task goal is usually predefined as it reduces the system complexity. However, modern industrial robots are expected to realise the task goal by themselves so that conventional robot programming can truly be replaced with advanced autonomous methods. Some techniques for primitives re-sequencing and motion planning already exist; however, learning that task planning skill with object affordance seems to be more intuitive since it allows to plan from high level to low level.
- *Skill reproduction and generalisation:* Techniques employed for regenerating skills can be different, depending on the model used in the primitive learning stage. Perception forms a crucial role when generalisation is necessary. Feedback terms may be added to the motion planning module in order to achieve online adaptation. It is certainly the case that active perception would improve perception accuracy; however, computational modelling of active perception skill is only in its preliminary research stage.
- *Incremental learning:* To further refine a skill performance, correction is a means to provide additional information for robots to learn a user preference based on inverse

reinforcement learning. Allowing emulation is an important user preference that differs from imitation in being mainly motion-driven or effect-driven. Again, affordance would definitely facilitate the emulation process, which should make researchers pay attention.

3 | LOW-LEVEL MOTOR SKILL FORMULATION

3.1 | Overview of LfD

LfD is a skill-learning process for a robot that does not require people to do hands-on hard-coding programming during learning. Instead, it provides other interfaces that are friendly to users for demonstrating the skill, which makes the interaction more intuitive, as shown in Figure 2. Performing demonstrations requires an interface (i.e. a medium); there are three main categories of interfaces for demonstrating a skill. Methods such as kinesthetic teaching and observational learning are commonly used as they are fundamental to routine human social learning. Methods like teleoperation, which are not applicable in skill transfer between humans, efficiently work between humans and robots. It is worth mentioning that some methods, like trial-and-error learning [21], are not in the scope of ‘demonstrating’, and hence cannot be classified as LfD.

Kinesthetic teaching is an approach in which the teacher directly interacts with the robot via physical contact and manipulates each degree of freedom to complete a task—for example, holding a robot arm to open the door [22]. This kind of direct contact interaction retains the teacher’s haptic perception while demonstrating, but the performance level is

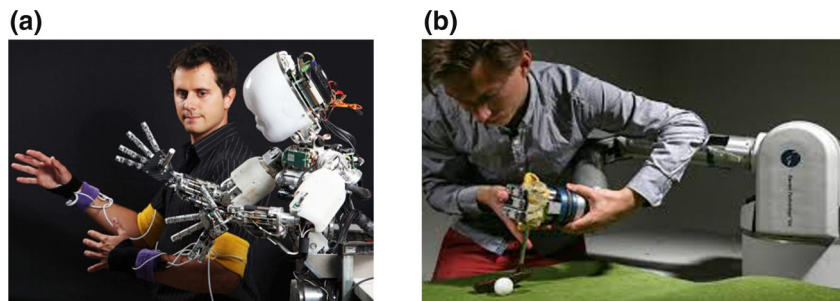


FIGURE 2 Human-robot skill transfer [10]: (a) learning via teleoperation; (b) kinesthetic teaching

hugely dependent on the teacher's operation level. In other words, for a remarkably complex behaviour (such as manipulating a humanoid robotic arm with redundancy [23]), it is difficult for the teacher to take into account the coordinated movement of all degrees of freedom, which reduces the expertise level of the skill performance to a certain extent. Demonstrators often just hold the end-effector of the manipulator to roam in its workspace and feel his/her motion is under constraint, which clearly shows that its drawback is its inadequacy for a highly dynamic task.

The **observational learning** approach primarily exploits computer vision systems or systems involving other modalities, which map the pose of humans to robots by estimating human joint states and calculating them according to the corresponding kinematic algorithms. In this process, a binocular camera with a depth sensor [24, 25] and markers [26] can be used. Other motion-recording systems like gyroscopes and accelerometers can also be used as a substitute for vision. Through observational learning, the pose of the entire person's arm will be considered, not just the end-effector (i.e. hands), which maximises the dexterity of robot skills. However, the drawback of this method is that no interactive force feedback can be obtained.

The **teleoperation method** requires the teacher to control the leader device (e.g. manipulator/joystick/exoskeleton/haptic device, etc.) [27], and consequently control the movement of the follower mechanism. Compared to kinesthetic teaching, this method has to use kinematics to map the configuration at both ends. Moreover, not all teleoperated systems are equipped with a force/tactile feedback system [28], so the teacher loses haptic perception while manipulating, which will degrade the performance of the skill; even if the system is based on a haptic device, the time-delay intervention will also lead to resonating, shaking and other problems [29].

The **correspondence problem** [30] describes the effect of the discrepancies in the embodiment (i.e. nature and working mechanism) of humans and robots on accomplishing the same skill. Different agents utilise different perceptual systems and physical mechanisms to interact with the environment or the other agents, which makes the sensory capabilities and motor capabilities among them different. From a technical aspect, for example, a robot might use depth-imaging sensors [31] as its visual sensor, which do not capture as much detail as human eyes do; and a 3 degree-of-freedom (DoF) robot may not be

suitable for a 6 DoF skill, which is easy to achieve by humans. To successfully evaluate a learnt skill, it is necessary to make sure both that agents share the same, or at least a similar, visual and motor capability [32]. However, the requirements of those capabilities for learning skills are task-oriented. As benefiting from contacting robot, kinesthetic teaching can simplify the correspondence problem as humans adapt their skill based on their own experience and cognitive skill.

3.2 | What is a skill?

The composition of skills mainly involves the problem of modalities or "what to imitate". The development of robotic skills usually progresses with demand. Robots were initially used in industrial production, and the task goal was principally to control the pose (i.e. position and orientation) of the robot. After that, pure position control cannot meet the force-related and contact-rich task scenarios, such as mopping and polishing. Researchers then began to consider encoding force information into the skill [33]. Later, some used bioelectric signals such as surface electromyography (sEMG) to match the human muscle activation level to the robot impedance, thereby enhancing the adaptability of the robot [27, 34].

In LfD, the motor skills of robots are usually subdivided into various skill primitives. When performing complex tasks, the robot needs to find the most suitable primitive among the existing skill library and arrange them in the correct order. In addition, there are many uncertain factors in the environment, and it is particularly necessary to perceive and select appropriate sensory information, which greatly reflects the robot's symbolic reasoning ability. Does the robot need to perceive and analyse all modalities in this high-level decision-making process? If not, how can humans quickly extract the key information needed to complete the task from the sensory information? Can such cognitive skills be transferred to robots technically? Problems like these have not been able to be solved well so far.

Lopes et al. [35] proposed a computational model of social learning. The article points out the essential difference between imitation and emulation. They suggested using the "strategy weighting triangle" to explain the "what to imitate" problem. In their theory, imitation refers to fully understanding and copying the demonstrator's behaviour and intentions, while emulation refers to the understanding of the effect of the demonstrator's behaviour and then achieving this effect with

action more open (i.e. observed actions or actions that were never-before-seen). An example of emulation can be described as this, the demonstrator grabs the object away from the desktop and put it down to another position, while the robot (i.e. emulator) realises the goal is to change the location of the object, so it pushes the object to the goal position instead of grasping it. The following baseline preference is to minimise the energy consumed by the behaviour. Through the trading-off between the three, the learnt skills may be better evaluated.

Figure 3 is a visualisation of the theory of Lopes et al. Non-social behaviour performed using a passive role intends to minimise the energy consumption. Imitation and emulation as two common imitative behaviours are the motion/action copier and goal/final effect copier, respectively. Points located in the middle of the triangle then indicate the nature of the imitator. From the above points of view, skills are comprised of single low-level motor primitives and high-level cognition/symbolic reasoning, although it is very challenging to model and integrate high-level reasoning into robot skills.

3.3 | Represent and learn individual motor skills

In order to facilitate the computational modelling and reuse ability, motor skills are often modelled into various primitives at a trajectory level. The article [36] summarises and introduces the algorithms for implementing skills learning at the trajectory level. A survey [37] broadly summarised the different technologies (including LfD and other techniques) that will be used in robot skill transfer learning and list the applications in different scenarios. Recently, some new review articles on LfD conclude the advances based on multiple categorising methods [38] and applications in the robotic assembly domain [39]. Meanwhile, readers can also obtain information from some classic LfD review articles, such as [40–42].

A simple recording and replay software will satisfy the behavioural clone, which is a brute-force method. However, skill synthesis and encoding allow generalisation in different situations. This kind of skill modelling technique seeks an abstracted and generic expression of the skill, which usually reduces the total number of feature points of the skill and generalises/modifies it at an abstraction layer. Generally speaking, skills can be described in two broad ways, namely, dynamic system-based approaches and probabilistic approaches.

3.3.1 | Dynamic system (DS)-based approaches

DS-based methods generally seek to encode the dynamic attractor landscape in the state space (i.e. typically position and velocity) of the demonstrations.

Dynamic movement primitive (DMP) is one of the most popular techniques to model skills at the trajectory level, which was first officially proposed in 2002 [43]. It was then updated and described in detail in 2013 [44]. The essence of DMPs is a second-order non-linear dynamic system, which contains a

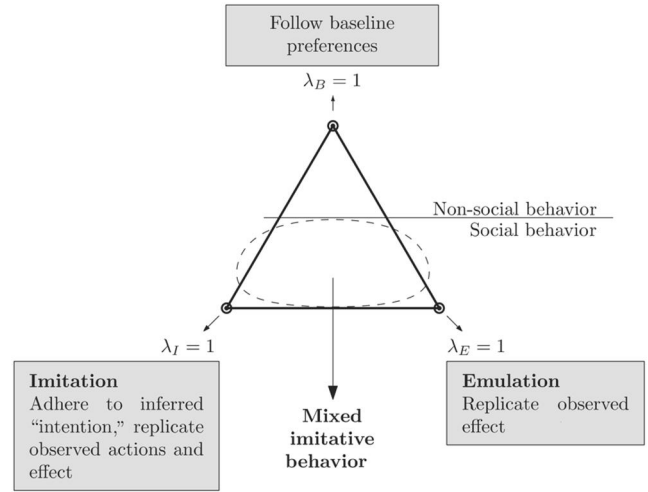


FIGURE 3 A computational model describing the relationship between imitation and emulation in social learning [35]

spring and damper. The original DMPs (e.g. non-periodic ones) are presented with the following set of differential equations in a first-order notation:

$$\tau \dot{v} = \alpha_v (\beta_v (g - x) - v) + f, \tag{1}$$

$$\tau \dot{x} = v \tag{2}$$

Since DMPs are orienting to the trajectory, thus we have the trajectory x and its rate of change v . g is the set point of the system and τ is a time constant which relates to the movement duration. α_v and β_v are positive constants that relate to the spring constant and the damping coefficient of the second-order system. f is a non-linear forcing term that is driven by a state parameter s .

$$f(s) = \frac{\sum_i \omega_i \psi_i(s)}{\sum_i \psi_i(s)} (g - x_0), \tag{3}$$

$$\tau \dot{s} = -\alpha s \tag{4}$$

where i is the index of basis, x_0 is the initial position of the trajectory and α is a constant. The state parameter s as stated in Equation (4), which monotonously reduces to zero as time passes, forms an indicator of the task completion. In other words, $s \rightarrow 0$ as time $t \rightarrow \infty$, which makes the forcing term $f(s)$ go to zero when the task is about to complete. While the forcing term is zero, Equations (1) and (2) become a standard spring-damper system that always leads x towards g . The personalised pattern of a trajectory is encoded in the forcing terms as stated in Equation (3) by using a set of weighted Gaussian kernel basis functions ψ . The learnt pattern is stored

as the weights ω . The scaling term $g - x_0$ helps to adapt the skill to a new initial position, while maintaining the global shape of the trajectory. The high popularity of DMPs confirms some of their advantages; the four main features are discussed below.

- DMPs can encode not only discrete (non-periodic) skills, but also rhythmic (periodic) skills, which is implemented by replacing the system with a limit cycle dynamic system. Nakanishi et al. [45] successfully realised biped walking locomotion of a humanoid robot with rhythmic DMPs. However, the walking trajectory in the experiment was generated by a state machine controller, and the walking demonstration of a human would cause failure. This well reflects that the physical or sensory discrepancy between imitator and demonstrator will greatly affect the success rate of skill transfer.
- DMPs can be easily extended to multi-dimensional systems by sharing the same canonical system (Equation 4) and using different transformation systems for each trajectory. In [46], DMPs are separately modelled for each joint of the manipulator by establishing multiple non-linear functions and transformation systems. In [47], motion trajectory and force trajectory are encoded into a single skill.
- Inherent generalisability is another advantage of DMPs, which can be implemented straight forwardly. The task performance duration can be adjusted by changing the temporal scaling factor τ to get a more rapid or slow state change, and the start/goal position can also be adaptable while maintaining the overall shape that is stored in the learnt weights ω [48].
- Because of the spring-damper system, DMPs are very stable and robust. They are also very resistant to small external disturbances. For large constraints like someone obstructing the robot arm while it is moving, a feedback term can be added into the system to adjust the behaviour online [49].

Despite DMPs showing many kinds of advantages in skill learning, they also have certain disadvantages [44]. The system cannot be used when the start and goal positions are too close. Moreover, there is a mirror effect when generalising the movement to a specific goal. To this end, [50] proposed a modified DMP formulation, as shown in Equation (5). This equation is then a replacement of Equation (1).

$$\tau \dot{v} = K(g - x) - Dv - K(g - x_0)s + Kf(s), \quad (5)$$

K and D are the stiffness coefficient and damping coefficient, respectively, which are somehow equivalent to α_v and β_v . The left of Figure 4 intuitively describes the DMPs that operate based on a canonical system and a virtual spring-damper attached to the start and goal. The canonical system, as a virtual timer, controls the phase s according to the time t , and the virtual spring-damper driven by s produces a virtual force at

each time and attracts the system state space values to the equilibrium.

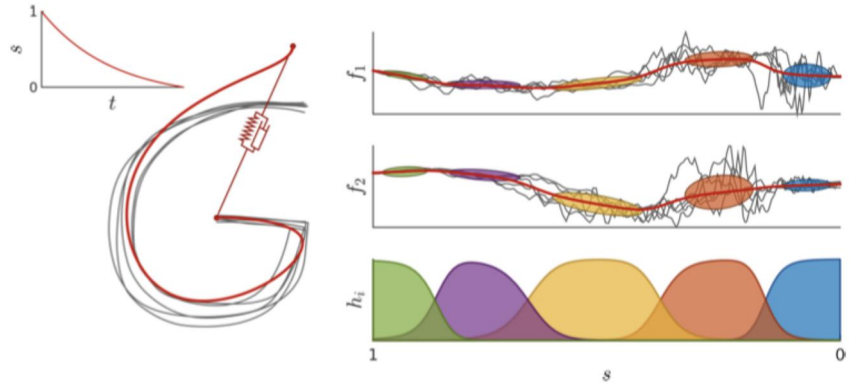
Other works that approach a better performance in using DMPs are as described below. By adding an external signal to rhythmic DMPs and introducing a set of additional dynamic systems for the temporal scaling factor, a smoothly changed and speed-adjustable rhythmic skill with synchronisation is obtained [52]. Kober et al. [53] extend the DMPs by involving external variables to each DoF, so that the perceptually coupled motor primitives are obtained.

Based on the insight that humans utilise a small number of motor primitives to generalise a large number of motions according to different environment stimuli (perceptual information or task parameters), gating networks can be used to further extend the generalisability of the skill. A gating network-based model Mixture of Movement Primitives (MoMP) [54] is proposed, which outputs a weighted sum of the old motor primitives to form a new movement. In MoMP, the gating network is triggered by an augmented state, which is associated with meta-parameters (i.e. the hitting position, velocity and orientation of the racket in a table tennis task). In order to learn the meta-parameter in that table tennis hitting task, two methods can be used, which are (1) analytically predict and convert to joint space using inverse kinematics; (2) episodic reinforcement learning approach (Cost-regularised Kernel Regression (CrKR) [55]). Summing up MP candidates some time may not be able to deliver a good policy (and may be even worse than a single primitive) [56]. The weights (i.e. the responsibility of a motor primitive to produce a new movement) for each motor primitive can be updated through reinforcement learning methods such as the method inspired by Relative Entropy Policy Search (REPS) [57].

The authors of [58, 59] introduce the idea of Query q to DMPs, which was inspired by MoMP to extend the generalisability. $q \rightarrow [\omega, \tau, g]$, is the key concept, and finds the mapping between queries (i.e. goal of primitives) and the learnt parameters. Then, for any given novel query q , the robot can generate a motion plan that is more human-like. The work [60] proposed DMPs that further refine the query system, encoding each style directly into the attractor landscapes (i.e. the forcing term f). The authors of [61] point out that the importance of the movement styles which can be utilised to accomplish specific tasks like shooting ball in Small Size eague and table tennis. ‘Style Adaptive DMPs’ (SADMPs) are presented that merge the weights of different styles to adapt to the changes in goal position.

Inspired by some findings in biology, Rückert et al. [62] provide a parameterised version of DMPs called ‘DMPSynergies’. This technique parameterises the basis function in the forcing term, so that, in multi-task learning, the synergies knowledge in each task can be shared and speed up the learning process. Very recently, another parameterised version of DMPs was proposed in [63], which is the Compliant Parametric DMPs (CPDMPs). This uses the Parametric Hidden Markov Model (PHMM) to encode the forcing term of the modified DMPs so that $f(s)$ becomes $f(\theta_m, s)$ where θ_m can be any high-dimensional variable that affects the shape of the

FIGURE 4 Dynamic movement primitives in conjunction with Gaussian mixture model [51]



trajectory. For example, θ_m can be the position of an obstacle. After training of the PHMM parameters based on multiple demonstrations and Expectation Maximisation algorithm [64], the system is then ready to generate a trajectory that avoids obstacles. In addition, a force-feedback coupling term is introduced into the transformation system so that the robot is able to handle the external forces.

DMP Plus [65] has been designed to increase the trajectory reproduction precision by adding bias to each Gaussian basis kernel ψ_i and truncating the kernel. In [66], the exponentially decaying canonical system Equation (4) is replaced by a linearly decayed one. The system shows comparable results and reduced the user's time expansion while teaching the robot. The authors of [67] developed a new DMPs representation called arc-length DMPs (AL-DMPs) that decouples the temporal speed and spatial shape of motion trajectories. This was done by representing the DMPs differential equation using the derivatives of arc-length rather than time. It is worth pointing out that AL-DMPs well solve the temporal scaling and time alignment problems. Gams et al. [68] studied the DMPs in a bimanual scenario where two arms are tightly coupled. They modified the DMPs to be cooperative DMPs by adding complementary coupling terms to each transformation system.

Stable estimator of dynamic systems (SEDS) is complementary to DMPs [69, 70]. A SEDS model uses Gaussian mixture model (GMM) to encode the attractor landscape as the joint probability of position and velocity. The learning outcome becomes an optimisation problem for finding GMM parameter values that minimise the trajectory error concerning all the demonstrations, such that the system is globally asymptotically stable. Similar to DMPs, high stability always allows converging to the goal. While DMPs allow for a single demonstration encoding, a SEDS is suitable for multiple demonstrations learning. A drawback of SEDS is that it assumes the dynamic system is time-invariant, which is not as versatile as DMPs, where the system dynamic could change with time as the forcing term changes. Another drawback is that the fixed attractor normally does not like to make the states vector (i.e. position and velocity) go away from the attractor, which may distort the overall shape of the motion. To this end, the Control Lyapunov Function-based dynamic movements

(CLFDM) approach was designed [71], which is also guaranteed to be global asymptotically stable. The difference between the two models is that the CLFDM uses regression techniques like Gaussian mixture regression (GMR) to model an unstable motion according to demonstrations; however, it learns a Lyapunov function to control the stability at runtime during task reproduction/generalisation by solving the problem of constrained convex optimisation.

3.3.2 | Probabilistic approaches

Probabilistic approaches seek the help of probability theory to encode a spatial or temporal pattern using joint probability density, while a method called probabilistic movement primitives (ProMP) can be slightly different, and is introduced below.

The GMM is one of the most popular models for encoding a trajectory. The term GMM is also referred to as mixture of Gaussian (MoG), and was first proposed to describe multimodal probability distributions [73], which is then utilised to encode the complex shape of a trajectory of low or high dimensions [74]. The idea of a GMM is as shown as below:

$$p(\xi_j) = \sum_{k=1}^K \pi_k \mathcal{N}(\xi_j; \mu_k, \Sigma_k), \quad (6)$$

$$1 = \sum_{k=1}^K \pi_k, \quad (7)$$

where $p(\xi_j)$ is a probability density function of a point j on the trajectory ξ_j ; there would be K number of Gaussian components in total, π_k is the prior knowledge of the k -th Gaussian. \mathcal{N} denotes the symbol of a Gaussian distribution. μ_k and Σ_k are the mean and covariance of the k -th Gaussian, respectively. Put simply, a GMM encodes the trajectory as the K -modal probability map that shows what spatial position a point is most likely occur at in the trajectory space. An example of GMM encoding uses single demonstration one-shot learning,

that can be seen in the left of Figure 5, where a 3D trajectory is encoded into a four-modal Gaussian map. The uniqueness of a DMP is the dynamic equation-based representation, which means that the frequently used forcing term of DMPs composed of weighted Gaussian kernel can be replaced by a GMM [75]. As shown in the right of Figure 4, GMM models the joint probability of forcing terms f and phase s , which is more compact when the trajectories have more fluctuations.

Hidden Markov model (HMM) and **hidden semi-Markov model (HSMM)** are both alternatives to GMM [76, 77]. The key difference between an HMM and GMM is that the HMM considers the transition probability between each states (i.e. each Gaussian modal). One can think of an HMM model as a GMM with latent variables (not directly observable) changing over time. An HMM encodes the state duration implicitly in the probability of no state transition, which may be inaccurate, while an HSMM explicitly defines the state duration using a log-normal distribution (\mathcal{LN}) with mean μ_i^D and covariance \sum_i^D , as shown in Equation (10). The parameters for modelling an HSMM (Θ^{HSMM}) based on K state components can be seen in the Equations (8) and (9).

$$\Theta^{HMM} = \{\{a_{ij}\}_{j=1}^K, \Pi_i, \mu_i, \sum_i\}_{i=1}^K, \quad (8)$$

$$\Theta^{HSMM} = \{\{a_{ij}\}_{j=1, j \neq i}^K, \Pi_i, \mu_i, \sum_i, \mu_i^D, \sum_i^D\}_{i=1}^K, \quad (9)$$

$$p(D_i)^{HSMM} = \mathcal{LN}(\mu_i^D, \sum_i^D) \quad (10)$$

D_i is the duration of the i -th state, a_{ij} is the state transition probability between state i and j , and Π_i is the prior of the state i , indicating that the probability of the initial state is i . The parameters of an HMM (Θ^{HMM}) would be similar to the HSMM, where duration terms are removed and j can be equal to i . Figure 6 visualises an example of encoding using HSMM, however an HMM example will not be shown here since HSMM has a similar structure but without state duration probability. It can be seen easily from the figure that six Gaussian components are used for encoding, where state transitions (i.e. arrows between states) are enabled with transition probabilities. Close to each state, a state duration probability density function is calculated, shown as small bell-shaped functions. Based on all of the above, state sequence probabilities, which describe the conditional probability of seeing a point in a certain state given the time variable, are plotted on the timeline.

Trajectory-based GMM is a method that explores not only the static feature but also the dynamic features (i.e. utilising the relationship between the trajectory and its derivatives, e.g., position, velocity, acceleration and jerk) [80]. As shown in Equation (11), ξ_t can be calculated through Euler approximation using spatial position (i.e. it can be any dimensions) of three executive time steps, where Φ_t is the matrix that relates to the Euler approximation. By stacking data of all the time

steps a generic function (Equation 12) can be obtained, where Φ is a large sparse matrix.

$$\xi_t = \begin{bmatrix} \mathbf{x}_t \\ \dot{\mathbf{x}}_t \\ \ddot{\mathbf{x}}_t \end{bmatrix} = \Phi_t \begin{bmatrix} \mathbf{x}_t \\ \mathbf{x}_{t+1} \\ \mathbf{x}_{t+2} \end{bmatrix}, \quad (11)$$

$$\xi = \Phi \mathbf{x} \text{ by stacking all the } t \text{ from } 1 \text{ to } T \quad (12)$$

Suppose multiple trajectories are obtained, information can be encoded either in a GMM, HMM or HSMM. Taking GMM, for example, a likelihood of trajectory ξ given state s can be computed where s is a sequence of indicators (i.e. one for each time step) that denotes which Gaussian component contributes to each time step, as shown in Equation (13).

$$p(\xi | s) = \prod_{t=1}^T \mathcal{N}(\xi_t | \mu_{s_t}, \sum_{s_t}), \quad (13)$$

where μ_{s_t} and \sum_{s_t} are the mean and covariance of the state (i.e. which Gaussian component) s_t at time step t . The maximum likelihood then can be performed with the help of Cholesky and QR decompositions [81] to solve the motion generation problem. As shown in Figure 7, the trajectory-GMM works well in a four-demonstration, 18 Gaussian components situation, which synthesises the trajectories to form a complex one that contains a diversion in the middle, allowing to go in either direction. This method does not require time/spatial aligned demonstrations.

Gaussian process regression (GPR) is a generic method that computes with brute force the high-dimensional trajectory distribution [82]. It finds the correlation between each degree-of-freedom and stores the trajectory distribution information into a high-dimensional positive definite covariance matrix. The trajectories can be regenerated or resampled by conditioning. Figure 8 illustrates the process of using a GPR method in modelling a 1D trajectory. This method results in a large covariance matrix and means, which may be easy to use. However, its performance depends entirely on the demonstration quality and it is prone to spatial/temporal variations. A tremendous amount of demonstrations may be needed to synthesise and infer a smooth trajectory.

Task-parameterised solutions are well summarised in [83], which clearly shows three types of solutions, namely, GPR with Trajectory Models database; Multi-Streams approach; and Parametric Gaussian Mixture Model (PGMM/PHMM). Generally speaking, the task-parameterised model includes information of task parameters (i.e. offset positions and transformations) as query points for each demonstration data point. Suppose the task parameter in each demonstration is fixed, the issue can be treated as a GPR problem [82] solving with techniques like GMM and Gaussian mixture regression. This allows the motion to be inferred based on novel task query in real time; however, GPR cannot handle

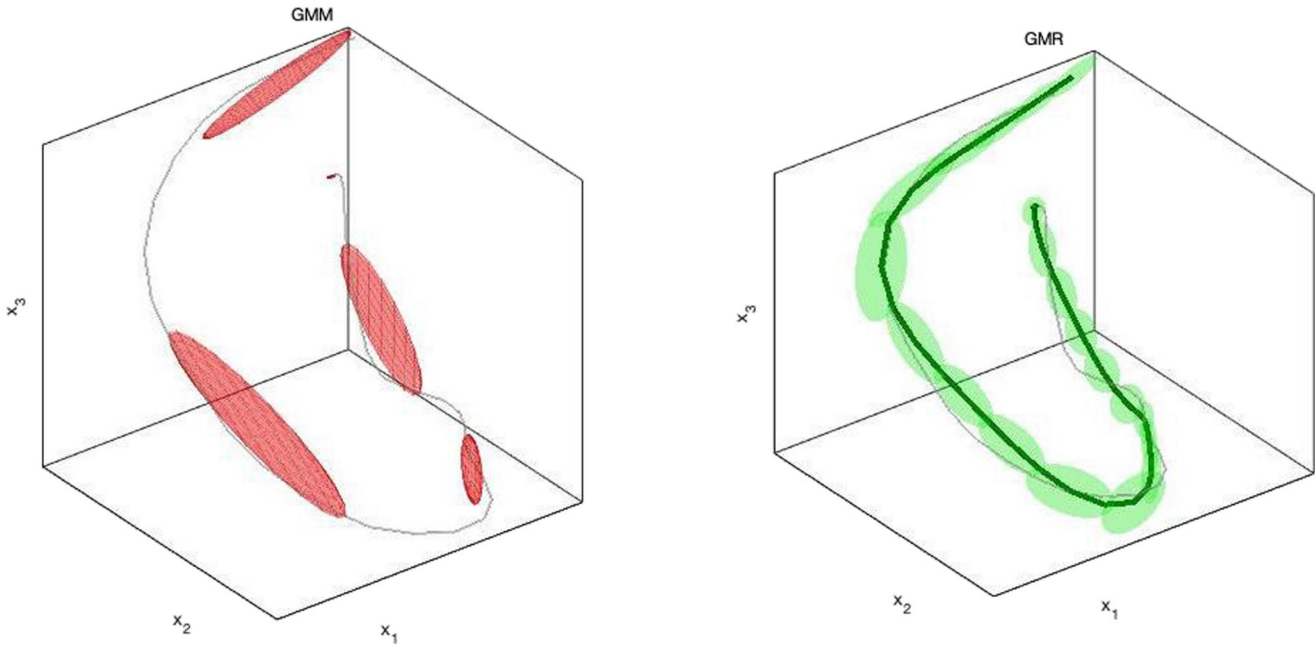


FIGURE 5 (left) Example of Gaussian mixture model motion encoding with five Gaussian components; (right) Gaussian mixture regression trajectory generation and its covariance [72]

query points that are too far from the demonstrated ones. The Multi-Streams approach utilises a different strategy that observes motion at multiple different frames, and trains each model separately, which, however, may raise a computational limitation issue [84]. The parametric hidden Markov model (PHMM) takes all the demonstrated motion and query into a single model, while PGMM holds the same strategy but does not take state transition into account [85]. PGMM/PHMM can be problematic sometimes as the covariances are not parameterisable (i.e. constants) and lead to a local minimum solution in Expectation-Maximization (E-M) learning. An improved version is presented in [83] based on PHMM, which makes a fusion of the main features of the above three models. Task parameterisation can be used in conjunction with trajectory-GMM model, where the motion derivatives are introduced.

Probabilistic movement primitives (ProMP) [86] uses a similar idea as DMPs to add an artificial clock that maps time to a phase variable with an arbitrary non-linear function. Locally weighted regression can also be applied to learn a set of basis function weights. The uniqueness of a ProMP model is that it further encodes the trajectory using GPR in the weighting space (to find a distribution of the weights), where abstraction is carried out twice to get a compactly represented model, which yields a number of fruitful properties to use, and it is easy to perform temporal/spatial generalisation and online modifications. It estimates a trajectory shape using a weighted sum of basis function, as $\mathbf{x}_t = \Phi_t \boldsymbol{\omega} + \boldsymbol{\epsilon}_x$, where Φ_t is a matrix of time-variant basis function and $\boldsymbol{\omega}$ is the weight that encodes the spatial information; $\boldsymbol{\epsilon}_x$ is the Gaussian noise which belongs to a zero mean covariance matrix Σ_x .

$$p(\mathbf{x}_t | \boldsymbol{\omega}) = \mathcal{N}(\mathbf{x}_t | \Phi_t \boldsymbol{\omega}, \Sigma_x), \quad (14)$$

$$p(\boldsymbol{\omega} | \Theta_\omega) = \mathcal{N}(\boldsymbol{\omega} | \boldsymbol{\mu}_\omega, \Sigma_\omega) \quad (15)$$

Once the weights are learnt, a probability map of the trajectory can be obtained as in Equation (14). To learn the weightings from multiple demonstrated trajectories, the parameter $\Theta_\omega = \{\boldsymbol{\mu}_\omega, \Sigma_\omega\}$ of a Gaussian distribution of the weights for all demonstrations can be learnt. Then Equation (15) can be used to derive the weights. An example of application of ProMP can be seen in [87] for table tennis, where the authors also studied the probability distribution in task space and robot joint space, so that the task space trajectory can be obtained from a ProMP model trained in the joint space. ProMP model has a flexible generalisation ability through probability conditioning, and it can also be used for action recognition, which is not so easily accomplished using DMPs.

3.4 | Learning techniques and reproduction of the skill

3.4.1 | Locally weighted regression

Locally weighted regression (LWR) [88] is a super-fast algorithm combining the simplicity of linear least squares regression and the flexibility of non-linear regression, which was originally proposed in [89]. It allows doing one-shot learning with the idea of doing linear regression locally on a non-linear problem. An extended version called locally weighted projection regression (LWPR) was introduced in [90] to reduce the

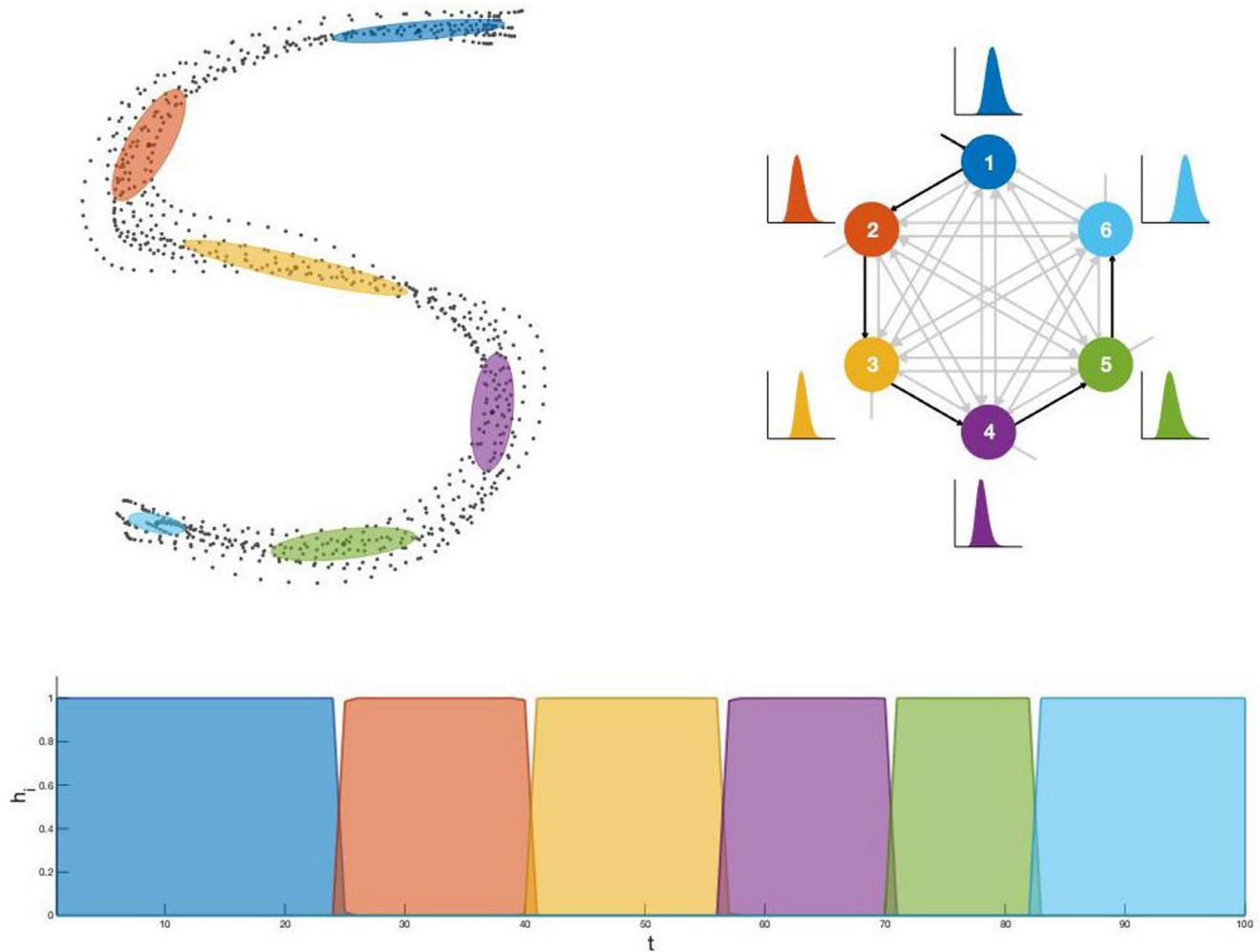


FIGURE 6 (left) Example of hidden semi-Markov model motion encoding with six Gaussian components; (right) state transition and state duration probability; (bottom) timeline of state sequence probabilities [78, 79]

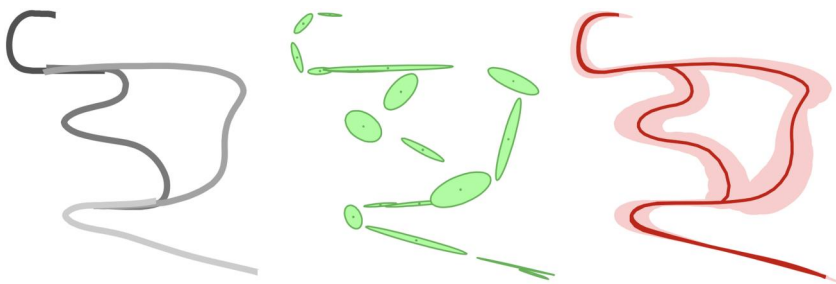


FIGURE 7 (left) Four demonstrated trajectories; (middle) Gaussian mixture model (GMM) encoding of the trajectories with 18 Gaussian components; (right) trajectory inference of the trajectory-based GMM and its covariance with two possible paths in the middle [80]

redundant complexity using a partial least-square. LWR is commonly used for learning of DMPs, and some recently published examples can be seen in [91, 92] for target reaching and pouring with obstacle avoidance tasks, respectively.

3.4.2 | Expectation-maximisation

E-M is an algorithm for learning the parameters of GMM- and HMM-based models [93]. It is an iterative algorithm that involves

two steps, E-step and M-step. More intuitively, the idea of E-M is to update the lower bound function of the objective function, then the lower bound function is maximised, which implicitly (indirectly) maximises the overall object function. Note that, the E-M algorithm may not work efficiently when the GMM or HMM contains too many components [94], which increases the dimensionality in solving E-M learning and significantly adding the computational complexity to the system. A variant of E-M called the Baum-Welch algorithm can be seen in [64], and an example of using it to train an HMM can be seen in [10].

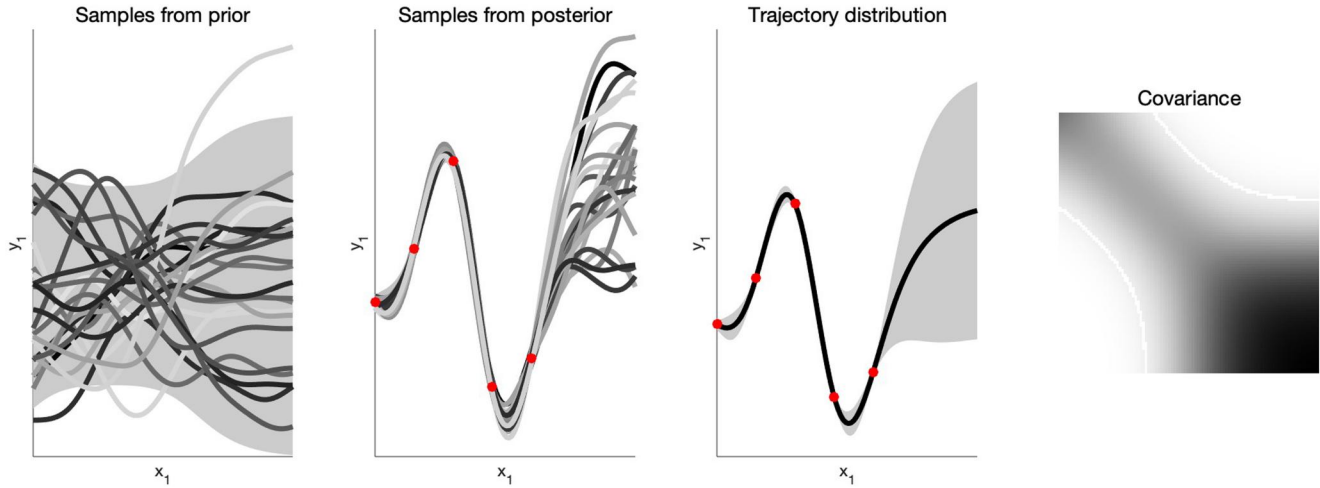


FIGURE 8 A 1D example of Gaussian process regression [80]

3.4.3 | Gaussian mixture regression

GMR is a commonly used technique that is frequently used in combined with GMM [95], HMM [10] and DMPs [51, 96, 97]. An excellent review on different regression algorithms can be found in [98]. Comparing GMR with LWR or LWPR, it computes the regression function from the (multivariate) joint density function. For a robot trajectory, the temporal value often acts as the model input and the other data as the output values, and by finding the conditional probability map of trajectory given time, $\mathcal{P}(x_{trajectory} | t)$, the infinitely differentiable (non-discontinuous) spatial trajectory can be obtained [99]. Another interesting example usage of GMR can be found in [70], which uses a GMM to encode $\mathcal{P}(x_{trajectory}, \dot{x}_{trajectory})$ and derive velocity at different spatial positions with GMR. Time-dependent GMR [100] is another variant usage that has been proven to be useful; it takes time, position and velocity together as variables for computing the joint probability density, and then estimates $\mathcal{P}(x_{trajectory}, \dot{x}_{trajectory} | t)$. In the right of Figure 5, the computed trajectory means and covariances can be visualised. In [101], not only the motion dynamics is encoded, the joint probabilities between position and force, and joint stiffness are also encoded using HSMM, then GMR is used to generate motion with the desired contact force, joint stiffness and velocity at each position.

3.4.4 | Reinforcement learning

The idea of reinforcement learning in robot policy learning is to find the cost and reward function for a robot to optimise its action, hence obtaining better outcomes. Conventional trial-and-error [21] requires robots to search in a very large action space, which means the training takes a very long time and is not friendly to a real robot. With a combination of LfD, the action space would be significantly confined to a local optima area, which reduces the training time. In other words, the success rate of reproducing the

learnt skills after one-shot learning may not be assured a high value with known and unknown reasons, which reminds us of the effectiveness of reinforcement learning (RL) and makes it a non-replaceable tool. In a ball-in-the-cup skill transfer study [102], the manipulator cannot succeed in reproducing the skill at all by just using a single demonstration because some spikes in acceleration cannot be reproduced correctly by the motor.

Inverse reinforcement learning (IRL) [103] has been proposed to extract reward functions based on perceiving actions and environment, as shown in Figure 9. In some scenarios, such as in minimally invasive surgery (MIS), expert surgeons are continuously optimising their actions through evaluation. However, even the surgeons themselves cannot fully explain what their evaluating policy is. This, again, reminds us of using IRL. Recently, Li et al. [104] presented their inspiring work on implementing IRL in robotic surgery procedures to derive a policy for surgery skill evaluation.

Most works in robot reinforcement learning are time-consuming and based on the real system. Some studies have tried to train a robot in a physical simulator, which may significantly reduce the time spent on the real system as it searches in a submanifold of space of policy and high-dimensional sensory input. However, it is quite common that the robot produces catastrophic failures after transfer to real scenarios, and that is why people often use model-free RL in the real system. Domain Randomisation (DR) is a useful technique that allows to fill in the reality gap (i.e. mismatch between real world and simulations) by randomising the simulation scenes, which embed the uncertainty of the world into the learnt parameters to form a more robust policy. Examples can be seen in [105, 106], which utilise DR to train for a balancing task and ball-in-a-cup task. Peng et al. [107] also developed a dynamic randomisation technique to randomise the unknown dynamics property of objects to link a Sim-Real.

In summary, DS-based methods are often used for direct learning control because of the characteristics of dynamic equations. Users can access and improve the robot system's

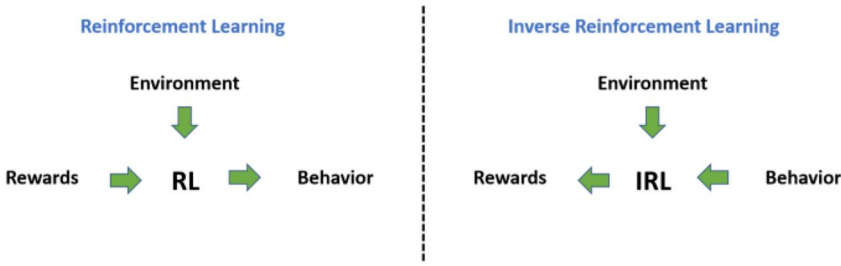


FIGURE 9 An intuitive comparison between reinforcement learning (RL) and inverse reinforcement learning (IRL)

stability and complete the task stably by online regulation. However, the method based on probability theory is more flexible and suitable for the target/via-point generalisation. It assumes that the user has an effective robot controller. Therefore, this method is usually used for task space trajectory planning rather than directly learning motor control variables. Nevertheless, this also facilitates the robot task planning in the task space at a high level based on cognitive skills.

3.5 | Temporal alignment

As for multiple demonstrations and complex skill demonstrations, demonstration differences are commonly seen in both spatial and temporal values. Demonstrators may not be able to or not necessarily produce temporal and spatial aligned motions. Spatial difference is totally dependent on the demonstrator; however, for temporal difference, there are some models that can get rid of these issues; for example, the AL-DMP and trajectory-based GMM as introduced in the previous sections. Trajectory-based methods solve this issue by pushing the model complexity to a relatively high level, which is not always ideal. As other methods do not provide a built-in mechanism for temporal alignment, commonly used techniques are introduced in this section.

Dynamic time warping (DTW) was first proposed in [108], and tackles speech recognition problems, as speech and robot motion share a similar trajectory structure using time series, and DTW is also well applied in the robotics domain. In [109], the authors use DTW and E-M algorithm to achieve a method that synthesises multiple demonstrations and computes the time-aligned trajectories at the same time. The outcome of this method is to compute a synthesised trajectory \mathbf{z} as a reference with a set of time mappings τ for each demonstration. The model of their work is based on Equation (16)

$$\begin{bmatrix} \mathbf{x}^1(\tau_t^1) \\ \vdots \\ \mathbf{x}^M(\tau_t^M) \end{bmatrix} = \begin{bmatrix} I \\ \vdots \\ I \end{bmatrix} \mathbf{z}_t + \mathcal{N}\left(0, \begin{bmatrix} \mathbf{R}^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{R}^M \end{bmatrix}\right) \quad (16)$$

where $\mathbf{x}(\tau_t)$ is the demonstrated trajectory with its time mapping at time t , and M is the total number of demonstrations. \mathcal{N} denotes the symbol of Gaussian distributions; here, the mean is zero and covariance matrices \mathbf{R} are the weighting which indicates the contribution of each demonstration in

shaping the reference trajectory \mathbf{z} . Open parameters are \mathbf{z} , \mathbf{R} and τ . Then a Kalman smoother is established, and optimises results via E-M. The E-step compute a Gaussian distributions \mathbf{Z} for the reference \mathbf{z} given \mathbf{R} and τ , and the M-step updates \mathbf{R} and τ given \mathbf{Z} by maximising the likelihood, then closes until they converge. Another extension of DTW, Generalised Time Warping (GTW), can be found in [110] which allows to work with high-dimensional, multi-modality data efficiently using a Gauss-Newton algorithm. DTW often causes non-continuous trajectories with very large acceleration jumps, instead of giving a constraint on the velocity or acceleration, and [111] proposes the Local Time Warping (LTW) method to solve that problem by optimisation using local information.

4 | COMPOUND SKILLS AND HIGH-LEVEL COGNITIVE REASONING

In the previous section, the learning outcomes of a robot are mainly defined as low-level policies, which can be utilised for a robot to perform a single task element. However, to endow the robot with more capability in handling people's routine work, achieving human-like high-level reasoning is required. High-level skills are the premise of combining low-level skills to complete more complex compound skills. Research in this area usually involves skill segmentation, labelling, recognition, and planning/sequencing. The significance of these studies is notable, considering that if the learnt skill primitives cannot be arranged correctly, the robot will not be able to complete more complex tasks autonomously.

Meanwhile, the robot can use this learnt high-level reasoning skill knowledge and produce novel skills through some generative frameworks. This is just like the learning process of humans, which is well reflected in the learning of musical instrument performance. For example, skilful violinists can compose and play a piece of never-before-seen sheet music almost immediately, as that is just resequencing of all the learnt 'skill primitives'. In this example, composing involves high-level decision-making, while playing involves resequencing and joining smoothly the learnt motor skills primarily. Learning of such complicated high-level cognitive skills is no different from learning the interaction between the labels of environmental conditions, agents and objects. The following section reviews previous studies and the current works in techniques of skills segmentation, recognition, motion taxonomy, joining, and sequencing; and then, an inspiring topic of

affordance-learning used in robotics is introduced, which tries to solve imitation in an affordance-based method.

4.1 | Skill segmentation and recognition

Segmentation of a continuously performed complex compound skill (i.e. computer assembly) can be solved in various kinds of ways. Supervised segmentation, or manual segmentation, is intuitive and precise, but is also time consuming. However, achieving a perfect generic framework for segmentation can be extremely challenging since the segmentation problems are often task relevant. The most straightforward, efficiently worked unsupervised method is based on stopped motion by assuming each cut-off point has zero velocity.

In [112], a segmentation algorithm based on spectral analysis is introduced. In this study, the target trajectory for conducting segmentation was the temporal varying 3D-position of hand. It is noteworthy that the affinity matrix of spectral analysis is computed using Gaussian kernel function based on spatial difference and temporal difference, which is considered to be less intuitive. This method has a drawback of manual selection for parameters (i.e. standard deviation term of Gaussian kernel and the total number of clusters), where, in a more complex trajectory, it is difficult to take all kinds of motor pattern into account, and people do not want to specify the number of sub-trajectories manually.

Meier et al. [113] modified the DMPs to be a discretised Kalman filter version to achieve movement segmentation with online movement recognition. At each time step, the E-M algorithm was used to update the optimal parameter estimation for each DMP in the library. The DMPs with the maximum of likelihood are then the best predicted motion, which can be done online with increasing confidence. An issue with this is that the estimated optimal DMP temporal scaling factor can be sometimes unreasonably large or small. The auto-segmentation is done by monitoring the drop of likelihood value. This method assumes all skill primitives are stored in a library and the segmenting target joins each primitive with no spacing, which largely constraints its effectiveness.

Another method based on Beta-process Autoregressive Hidden Markov Model (BP-AR-HMM), which is less straightforward but has good performance, was proposed by Emily et al. [114, 115]. For the details of this algorithm, refer to their works; however, the process is briefly summarised below. A beta process (BP) is seen as the conjugate of the Bernoulli process (BeP), which describes the parameter of BeP and implies an infinite number of features/modes (i.e. infinite possible patterns of a segment). Suppose multiple compound demonstrations (i.e. time series) are given, each demonstration can share modes in the infinite features set, which is described as different subsets of all the features and is encoded in a binary vector f_i , where f_i indicates which mode has occurred in series i . This suggests that features can be shown multiple times in the same or a different time series, which benefits the process of multiple demonstration during teaching. Then, a Dirichlet

distribution is used to derive the transition probability vector $\pi^{(i)}$ with finite parameter for HMM, which helps to compute the modes of $z^{(i)}$ for the next time step based on all accessible knowledge. Finally, the problem is solved by defining VAR dynamics for observation $y^{(i)}$. Notice that $\pi^{(i)}$ has finite parameters, which suggests that the total number of modes does not need to be specified. The structure of BP-AR-HMM can be seen in Figure 10, and application examples can be seen in [116, 117].

The above-mentioned methods worked fine in certain situations and performed with a key concept of similarity. In other words, it attempted to maximise the similarity/likelihood. Other works shown below tackle segmentation in another way based on indicative events. Konidaris et al. proposed a segmentation method based on changepoint detection [118]. The idea would be finding the point where a coming time step cannot be fit into the same model as before. This method can only be used with the prerequisite that all of the model is stored as the prior or just using a simple model like linear/quadratic trajectories.

Zhe et al. presented their Sensorimotor Primitive Segmentation algorithm based on triggering events [119]. This method utilises Bayesian online changepoint detection (BOCPD) [120] as a technique with multi-modal sensory data including robot state, environment/object state and tactile sensory signal based on a BioTacs sensor [121]. With these setups, the results show that it is very effective for using such multi-modal sensory signals, especially for using a tactile signal within a contact-rich task.

4.2 | Skill taxonomy and skill library

In Section 3, it was shown that DMPs and probabilistic approaches can store the skill pattern into each primitive. That actually facilitates the classification process. The motivation of classification is that, from the result of motor skill auto-classifiers, one would be able to implement motion prediction and choose the reactive skill from the pre-constructed skill library to solve problems. DMPs are mainly designed for skill generation and not yet for classification, although they show some capability of classifying [122]. Probabilistic models like an HMM would be fascinating in classification. Motions can be easily put into a category with its model parameter value, for example, an HMM parameter of Θ^{HMM} . Motion is classified by finding a set of parameters Θ^{HMM} to maximise the likelihood of seeing a motion \mathbf{x} , as shown in Equation (17).

$$\Theta^{HMM} = \underset{\Theta^{HMM}}{\operatorname{argmax}} p(\mathbf{x} | \Theta^{HMM}) \quad (17)$$

All of the classified motion models that are fulfilled by personalised characteristics encoding construct a skill library for use in specific application domains. However, such classified motion is not automatically labelled with physical meaning.

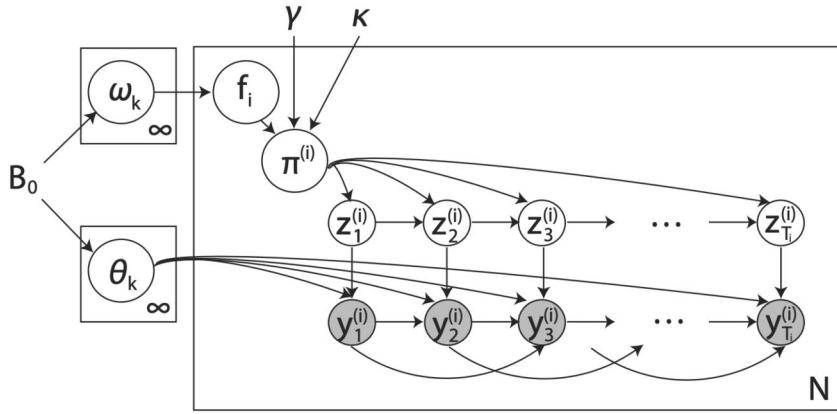


FIGURE 10 The illustrative structure of beta-process autoregressive hidden Markov model [115]

In other words, similar motions in a skill library, which is constructed in LfD, can be used interchangeably. However, skills like ‘push’ may not necessarily hold similar spatial shape in the task space (e.g. push could in any direction with any speed). This increases the difficulty in assigning a symbolic label to motion primitives based on current LfD skill representation models, whereas symbolic labels can be very helpful in planning a compound motion at high level. To this end, would it be very helpful to derive a better motion taxonomy methodology for generic usages, by finding the most effective and most explicable sensory modalities?

Aksoy et al. [123] proposed a Semantic Event Chain (SEC) tool for object–action relation, as shown in Figure 11 (a). In SEC, objects in the scene are segmented and semantic scene graphs are established based on the object’s spatial relations in four modes. With SEC as the basis, the most fundamental manipulations in life are studied in [124]. This research summarises a manipulation ontology tree which contains three fundamentally different types of manipulations and six manipulation goal categories, as shown in Figure 11(b). The authors further explored the importance of motion trajectory in motion taxonomy and discovered a positive answer.

Different from the above, a motion taxonomy for manipulation embedding with similarity metric was proposed in [125]. In this taxonomy, four main aspects are involved for encoding motions to binary vectors, which are, interaction type, object structural outcome, motion trajectory, and active descriptor, which indicates whether a tool is used to actively manipulate or not. The motion trajectory analysis is based on principal component analysis (PCA); interestingly, this binary encoding allows to compute motion differences using weighted Hamming distance with tuneable weight (i.e. trajectory profile may influence the taxonomy more than the interaction type). The calculated distances between motions can help to cluster similar motions, as shown in Figure 12, which is produced by a visualisation tool (t-SNE) [126].

In addition to all the above, other inspiring works can be seen in [127–129], which present motion taxonomy in different perspectives. Additionally, motion taxonomy could be beneficial for extending skills in affordance-based learning, which is introduced in Section 4.4.

4.3 | Skill sequencing and joining primitives

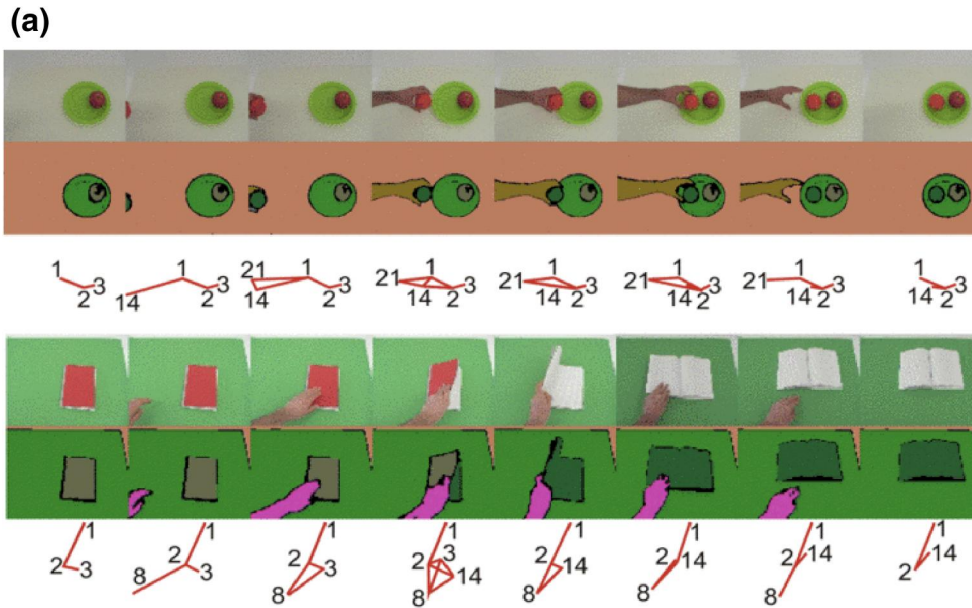
Enabling robots to solve problems in a complex environment and perform a series of actions requires the robot to know the meaning of the label behind an action. A metric for inferring task constraints and goals can be essential.

Ekvall et al. [130] presented their framework for complex task planning and resequencing the learnt sub-tasks. The essence of this framework is to express complex task demonstrations as many series of states/effects (i.e. effects mainly for absolute/relative position changes). Based on all the demonstrations, task priority constraints can be found, hence the possible skill sequence can be found to achieve an effect. The learning framework also allows incremental learning, which updates the knowledge as new demonstrations arrive. Another area of work that trains the robot incrementally with the help of vocal comments under a LfD framework can be seen in [131]. HMM is commonly used in high-level skill planning and recognition [132, 133], as it encodes the transition between each state in a time-series. The Hierarchical Hidden Markov Model, formally defined as Hierarchical Dynamic Bayesian Network (H-DBN), in multiple levels is presented in [134]. Another version of H-DBN, Growing H-DBN, can be seen in [135], and has two levels for encoding high-level abstraction and low-level motor skills, respectively.

Nevertheless, one can always define the action labels manually and join all the primitives in a predefined sequence. Joined primitives can be problematic while they are reproduced, because of the discontinuity. Problems can be solved by refining the connection between two primitives. For example, third-order DMP formulation allows a smooth acceleration profile [136] or adapting the primitive model parameters to combine primitives as one skill [137].

4.4 | Affordance-based learning

Gibson et al. [19] first proposed their theory about affordance a few decades ago. It is used to describe the relationship between objects/agents, actions and effect after actions. For example, a larger force of a hand will push a ball away along a



(b)

Type	Name	Description	Goal Category	Abbrev.	Examples
1	Hand-Only Actions	Hand alone acts on target object or Hand grasps tool and acts on target object	Rearranging	r	E.g. stir sugar into coffee, or push an object to free a space
			Destroying	d	Cut dough into pieces Chop wood
2	Separation Actions	Hand manipulates one object to either destroy it or to remove it from another object.	Break	b	Break off, rip-off, but also uncover
			Take-Down	t	Take down an object from another one
3	Release determined actions	Hand manipulates object and combines it with second, target object. (Many times this requires a grasp.) Combination of the objects is determined by the way the hand releases the grasped object.	Hide	h	Cover an object with another one
			Construct	c	Build a tower

FIGURE 11 Taxonomy based on semantic event chain

direction for a long distance, while a lower force can only produce a short movement. Examples can be seen in [138], where a robot learnt the rolling capability of a toy car and learnt the tool selection skill by searching the affordance of objects in the scene for achieving an effect. Figure 13 shows that affordance can usually be used in three ways. As action, object and effect are correlated, inference of one component can be feasible when other two components are known. Do et al. [140] proposed a deep network-based framework for affordance recognition training, which is called ‘Affordance-Net.’ As shown in Figure 14, with the object detected, its affordance attribution inference can be accomplished (with the colour-augmented area). Based on that, robot WALK-MAN is able to grab a bottle and pour towards a pan.

LfD and affordance learning can be used as a conjunction; in [139], affordance is learnt with Markov Chain Monte Carlo and Bayesian network [141]. In addition, a robot is able to perform task abstraction based on the action–object state demonstration, hence, the robot learns the policy of optimally finishing the task. Learning of the policy is achieved by adopting a variant of Bayesian inverse reinforcement learning [142, 143]. Based on the idea of affordance, Kroemer et al. [144] present a kernel-based approach that works together with DMPs, which improves the action planning due to better perception. In their work, affordance of object subparts is learnt (i.e. a particular object shape is affordable for specific actions and effects). Kernel basis function is used for computing subpart similarity in shape, and kernel logistic

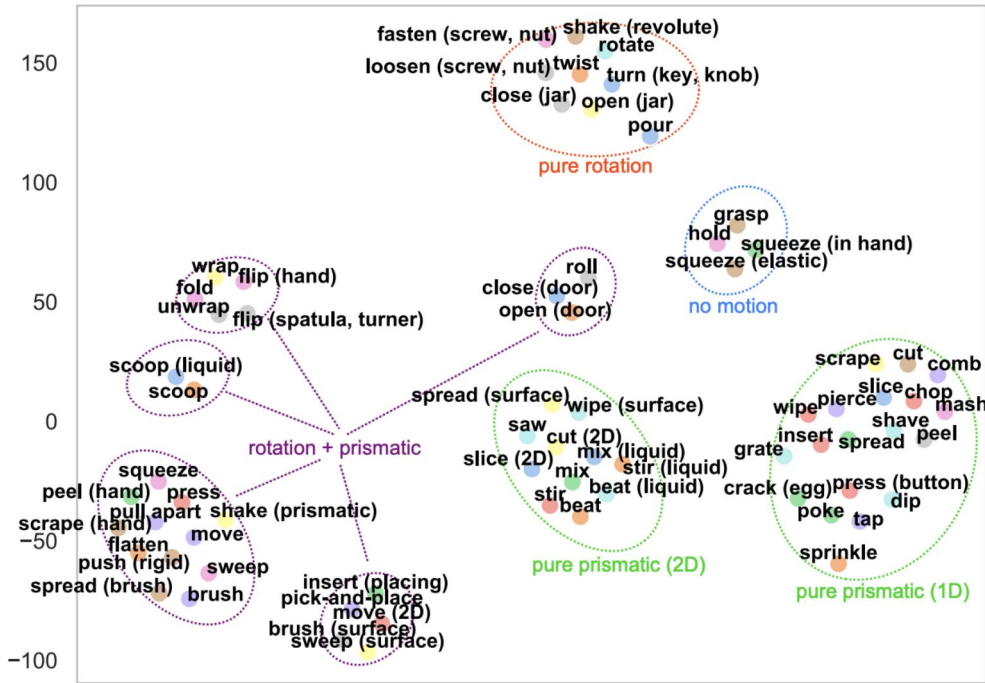


FIGURE 12 A motion taxonomy: visualised by t-SNE [125]

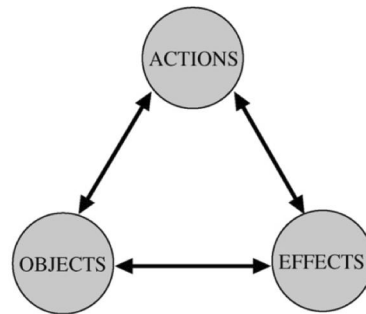
regression is used to choose a DMP. Another interesting example can be seen in [145, 146], where potential object grasp positions are learnt via image processing with machine learning or estimated by a Fuzzy Gaussian Mixture Model (FGMM).

5 | CURRENT WORKS AND OPEN ISSUES

5.1 | Collaborative skills

Collaboration between robots and humans is remarkably useful but challenging, even just for a human-like hand-over task. Achieving collaborative or interactive skills requires a robot to have an accurate human motion recognition and estimation module, which also takes safety factors into account. Collaboration is the consequence of reaction to an action or an active promotion to the overall task goal. It is relatively easy to train a robot to collaborate in a supervised approach, where action primitives are manually labelled. However, an unsupervised approach is more preferable because of the exemption of labelling work, so that robots can observe human co-workers all day long or actively involved in the task as a co-worker and discriminatively learn the intention–action pairs.

HMM, a widely used time-series model which is capable of encoding both temporal and spatial patterns, could be a way to approach collaboration. In [147], a Mimetic Communication Model, as shown in Figure 15, was proposed to realise collaboration control. The motions of humans and robots are recognised as continuous HMMs (CHMMs), where human motion is used to trigger interaction state changes in a discrete HMM (DHMMs) and select an appropriate reference motion primitive



inputs	outputs	function
(O, A)	E	Predict effect
(O, E)	A	Action recognition & planning
(A, E)	O	Object recognition & selection

FIGURE 13 Put affordances into use [139]

for the robot. Then the reference trajectory is modified online to adapt to human behaviours, and meanwhile, humans will also adapt to robot behaviour online (i.e. do not need a computational model for that, i.e. the human's reaction. However, in the future, it is interesting to compute the model of the human's adaptive behaviour to influence the human indirectly).

Another fascinating example of a collaborative skill transfer framework is proposed in [148], where ProMP is used. For achieving a single task collaborative skill, the motion demonstration of both humans (with P DoF) and robots (with Q DoF) are encoded into a single ProMP model using M number of demonstrations as described in the previous section. Then



FIGURE 14 AffordanceNet: objects detected with the affordance attribution attached [140]

the probability distribution of a motion pattern (i.e. $p(\omega; \Theta)$ where ω is the weight of basis functions that encodes a motion pattern, Θ is a hyperparameter) is a multivariate distribution where part of its dimensions relate to human motion, and the rest are for robot motion. In the inference stage, by providing a human motion y^* , the likelihood $p(\omega | y^*; \Theta)$ is obtained by conditioning. To generate a robot motion, the weights are integrated in the posterior distribution. This interactive skill also has a mixture representation if multiple subtasks exist, which allows a non-linear correlation between tasks. The process is relatively straightforward since it utilises the idea of Gaussian mixture, which was covered in the previous section. All the key concepts of this method are depicted in Figure 16. Apart from skill learning for collaboration, perception accuracy is also an important factor that influences performance as perception is directly related to the inference of people's intent. Su et al. [149] trained a model-free Deep Convolutional Neural Network (DCNN) for recognition of surgeon gestures based on the information from a kinect camera, inertial measurement unit (IMU) and EMG sensors.

5.2 | Learning to emulate

Start with questions: 'If you need to get an object right in front of you, would you choose to grab by your hand or grasp with another grasping tool? If the object is out of reach of the robot's end-effector, would the robot be able to use a grasping tool in its reach to extend its capability of achieving the goal without a human's explicit demonstration or being told to perform like this?' As a human, reasoning about a task goal is relatively easy with knowledge about the affordances, which is a result of training that happens every day. Whiten et al. [150] presented a taxonomy of social learning effects that classifies the emulation within copying. Emulation is first distinguished from imitation by the observation of children that can achieve a goal using idiosyncratic means which were never seen previously [151]. It suggested an approach for engineers of using a different way to re-look at imitation learning problems in an emulation aspect. Normally, the term emulation means to copy the end-state of a series of actions or copy the final effect of a series of actions (Figure 17).

Despite affordance learning being an alternative of emulation that facilitates the generalisation of skills, emulation can be a very unique way to produce novel skills that forms an alternative to conventional imitation learning. A good example of emulation

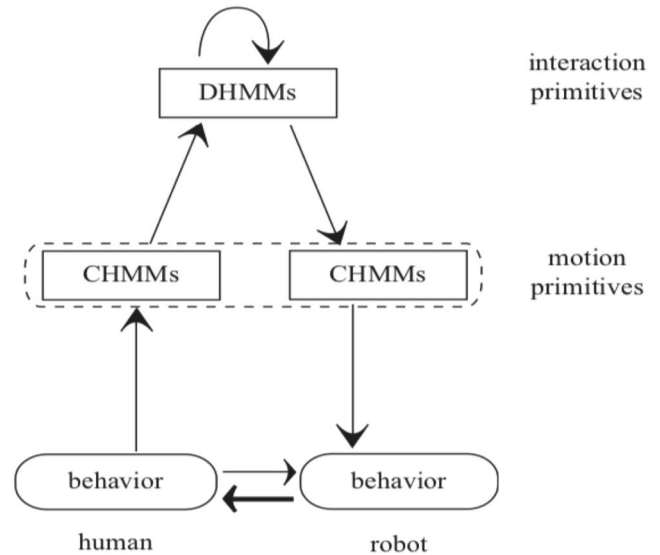


FIGURE 15 Mimetic communication model [147]

learning in robot skill learning can be seen in [152], where skills like pick-and-place can be done with moving obstacles. Comparing this to the previous works, where obstacle positions are assumed to be fixed after calculating an optimal trajectory, it contains an emulation module which feeds an estimated parameter (i.e. relates to the Gaussian representation of obstacles) back to the system using a utility function. At each time step, the solution would be the trajectory that minimises the utility function, which is actually minimising the error between the learnt demonstration trajectory and the generated trajectory with obstacle avoidance (i.e. satisfying both task and state constraints). Apart from that method, where the task constraint and state constraint are manually defined, affordance-based methods could help to select those constraints based on learnt high-level knowledge, which would definitely be a research direction worth attempting in the future.

5.3 | Active perception

Perception provides information for an agent to learn the context of the environment. To the best of the authors' knowledge, robot perception is commonly used in a passive manner, while humans often take an active role in perceiving information for useful modalities. For example, perceiving a circular object does not necessarily mean that is a ball, it could

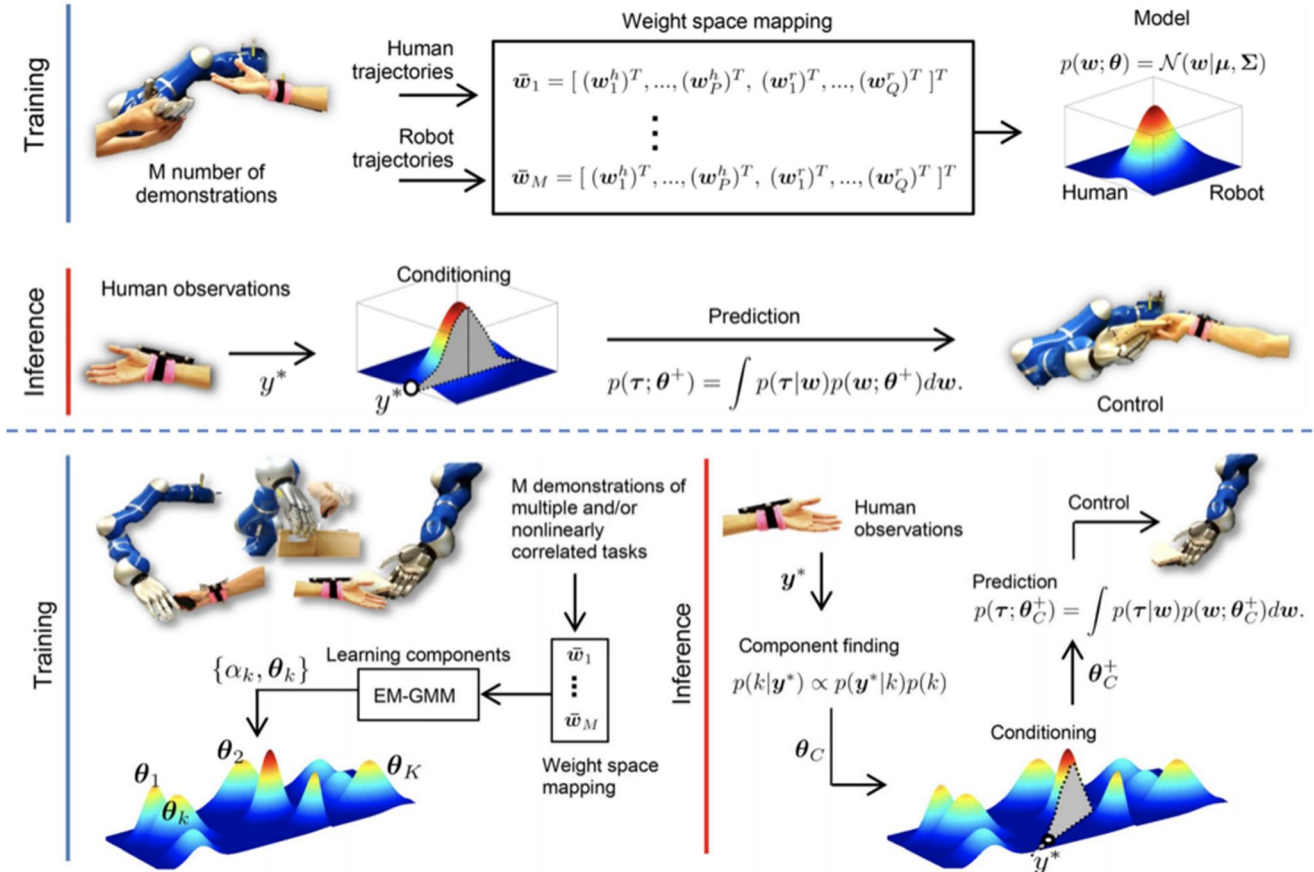


FIGURE 16 Key concept of a probabilistic movement primitives (ProMP)-based collaborative skill realisation: (top) single task interactive ProMP training and inference key process; (bottom) multi-task mixture of interactive ProMP training and inference key process [148]. CHMM, continuous HMM; DHMM, discrete HMM; HMM, Hidden Markov model

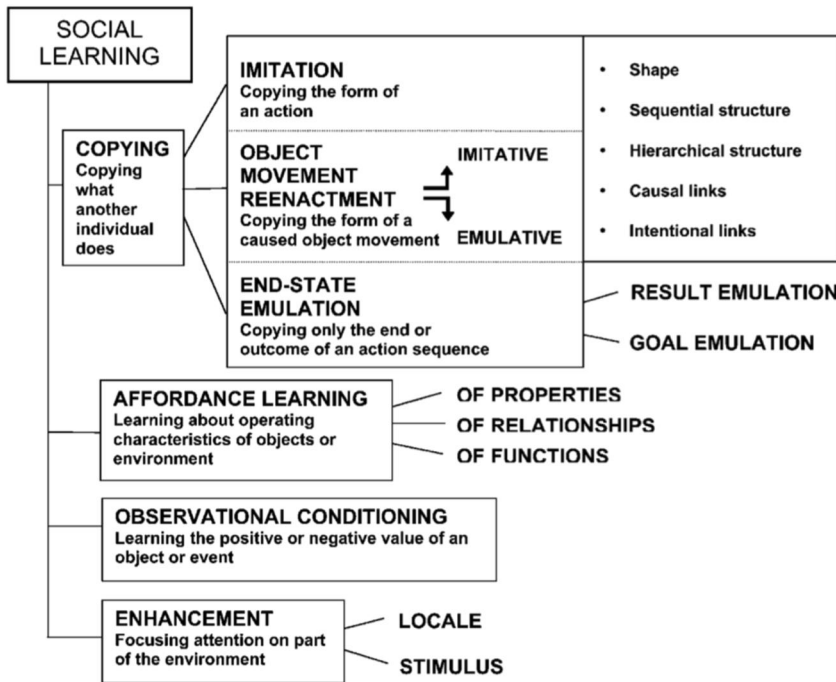


FIGURE 17 A taxonomy of social learning effects [150]

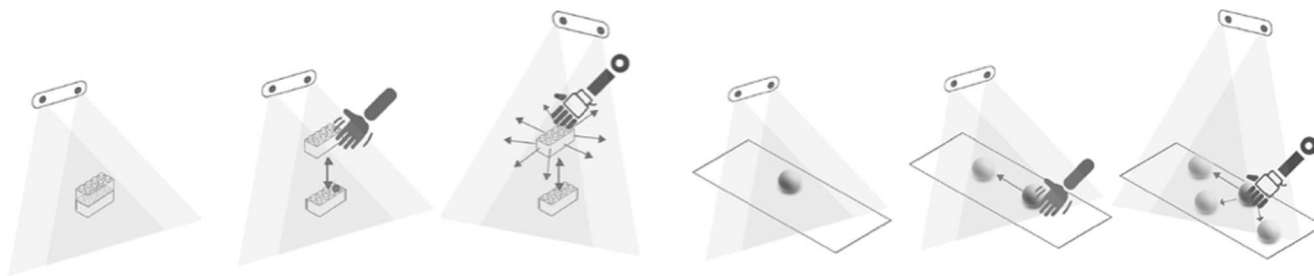


FIGURE 18 Illustration of the use of active perception to obtain more information [153]

also be a piece of paper in a circular shape, and people usually actively move their viewpoints to get a better perception. Figure 18 intuitively demonstrates this problem. Some articles have also pointed out the importance of being an active perceiver in an experiment with animal subjects [153]. The essence of active perception is defined as setting up goals based on the current belief about the state of the world and performing behaviours that may achieve it [154]. One of the most informative sensory modality in perception is vision. The information embedded in images is not fully extractable in one go. The goal of active perception is to know the context well, adopting a top-down attention strategy in the vision system has been proven to be efficient [155, 156], and this is a feature selection strategy that starts from the highest level.

In [157], an active sensing control method is proposed and tested in a more plane simultaneous localisation and mapping (SLAM) application. In their work, an online gradient descent strategy is used to shape a B-spline as an optimal path for perception. The results showed that relying on a calculated optimal path, which was compared to a straight path, the maximum estimation uncertainty can be reduced. This work is being extended to more complex scenarios [158]. As seen in the previous sections, by adding perceptual coupling to the skill model, the skill will be adaptive to the changes. Imagine if better perception could be achieved by a robot as a skill, robots would be more human-like, which may attract the attention of increasing numbers of researchers.

5.4 | Learning from failure

Another interesting topic would be learning policies from failures as they are often ignored in conventional LfD. Shiarlis et al. [159] proposed an algorithm called Inverse Reinforcement Learning from Failure (IRLF) based on IRL. Using IRLF, policy can be derived from both successful demonstrations and failure demonstrations, even if the failure trajectory is very similar to the successful one. Common LfD methods aim to maximise the similarity between the generalised trajectory and the demonstrations; when failures occur, robots can also minimise the possibility of reproducing such a mistake. Based on this idea, [160] proposed a framework that a human demonstrate failure actions, followed by a robot's exploratory trials that make the policy diverge from the failures, which was tested using a real robot.

5.5 | Incremental learning from correction

As discussed above, the action may not be perfectly reproduced after the learning process, hence the skill refinement process would be very helpful. Although a huge amount of state–action pairs may have been used for training, due to the complexity and uncertainty of the environment, robots still produce errors. For example, actions may be generalised to an unexpected goal. In [161], an interactive algorithm named confidence-based autonomy was proposed. This method the for expert's corrective demonstration to incrementally update the trained policy. Additionally, they used the corrective demonstration as a replacement of the policy that would be generalised under the specific state.

However, there can be another way for correction learning that has been rarely tried; that is, to seek changes in both low-level dynamics and high-level goals. A robot may continuously monitor its dynamic changes, if the human interacts with the robot, most likely an error has occurred in either low-level motion or the high-level decision, otherwise, refinement is required. By abstracting these human intentions or preference during the correction process, and putting this knowledge into use, would the performance be getting better?

Very recently, a fascinating learning outcome from a demonstration framework was proposed [162], where robots and humans share the role of controlling motion. The human correction is analysed to learn the human's preference. For example, the human may want the robot to hold fragile objects (i.e. slow down the motion), which is not a predefined preference that a robot would take into account. The robot needs to be aware of its lack of ability to explain the human's intention by having less confidence, and then the reason about how to behave to meet people's requirements, which is potentially solved in [162, 163].

5.6 | Open issues

There remain many open issues for human–robot skill transfer that are difficult to answer or close. On a technical level, humans arrange tasks through high-level cognition and use low-level motor muscle memory to generalise certain behaviours, which is one of the most commonly agreed facts. However, whether computers can have the same cognitive abilities as humans may remain controversial for a long time in

the future. Whether or not it exists, a generic framework or solution to arrange tasks to solve problems through cognition needs everyone's joint efforts to find out. An agents' emulation behaviour is actually the production of a kind of 'creativity'. However, whether this kind of 'creativity' is necessary to serve the development of human society is also a very interesting topic. In terms of law and ethics, the skill transfer framework is based on an individual's experience or a small number of individuals' experiences. More investigations may be needed to discover if it is suitable to be applied in real industrial applications.

6 | CONCLUSION

A comprehensive review has been provided herein on the topic of techniques used in human–robot skill transfer, to enable readers to capture an overall bigger picture of issues that may be met in research. Both a dynamic system-based model and a probabilistic model can be used to model individual skill primitives. With a different model, various learning and generalisation techniques could be used. To allow reactive and complex task performance, high-level reasoning is the most important aspect. HMM-based methods are shown to be very effective in high-level knowledge learning. Despite that, active perception should definitely be studied for a compact computational model to facilitate the perception accuracy in collaborative skill performing. Affordances-based learning is then a useful tool for tackling motion/intention recognition and action planning at a high level. Finally, the importance of incremental learning for robot skill refinement has been emphasised. Based on the introduction given here, it is hoped that researchers will have a clear idea of the structure of robot skill learning and transfer this knowledge and explore for more potential topics to further extend robot capability.

ACKNOWLEDGEMENT

This work was partially supported by Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/S001913.

ORCID

Yuan Guan  <https://orcid.org/0000-0001-9040-3169>

Chenguang Yang  <https://orcid.org/0000-0001-5255-5559>

REFERENCES

- Duque, D.A., Prieto, F.A., Hoyos, J.G.: Trajectory generation for robotic assembly operations using learning by demonstration. *Robot. Comput. Integrated Manuf.* 57, 292–302 (2019)
- Tang, Z., et al.: Assembly sequence planning based on graduated non-convexity and concavity procedure. In: *Proceeding of the 11th World Congress on Intelligent Control and Automation*, pp. 4933–4938. IEEE, Shenyang, China (2014)
- Niechwiadowicz, K., Khan, Z.: Robot based logistics system for hospitals-survey. In: *IDT Workshop on Interesting Results in Computer Science and Engineering* (2008)
- Silver, D., Bagnell, J.A., Stentz, A.: Learning from demonstration for autonomous navigation in complex unstructured terrain. *Int. J. Robot. Res.* 29(12), 1565–1592 (2010)
- Osa, T., et al.: Trajectory planning under different initial conditions for surgical task automation by learning from demonstration. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6507–6513. IEEE, Hong Kong, China (2014)
- Chen, T.L., Kemp, C.C.: Lead me by the hand: evaluation of a direct physical interface for nursing assistant robots. In: *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 367–374. IEEE, Osaka, Japan (2010)
- Rozo.Castañeda, L., et al.: Learning collaborative impedance-based robot behaviours. In: *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pp. 1422–1428. Bellevue, Washington, USA (2013)
- Kuderer, M., Gulati, S., Burgard, W.: Learning driving styles for autonomous vehicles from demonstration. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2641–2646. IEEE, Seattle, WA, USA (2015)
- Zhang, F., et al.: WAGNN: A Weighted Aggregation Graph Neural Network for robot skill learning. *Robot. Autonom. Syst.* 103555 130, (2020)
- Calinon, S., et al.: Learning and reproduction of gestures by imitation. *IEEE Robot. Automat. Mag.* 17(2), 44–54 (2010)
- Kober, J., Peters, J.: Learning motor primitives for robotics. In: *2009 IEEE International Conference on Robotics and Automation*, pp. 2112–2118. IEEE, Kobe, Japan (2009)
- Billard, A., et al.: Survey: robot programming by demonstration. In: Siciliano, B., Khatib, O., (Eds.) *Handbook of Robotics*, vol. 59 Springer, Berlin, Heidelberg (2008)
- Havoutis, I., Calinon, S.: Learning from demonstration for semi-autonomous teleoperation. *Aut. Robot.* 43(3), 713–726 (2019)
- Yang, D., et al.: Learning from demonstration: dynamical movement primitives based reusable suturing skill modelling method. In: *2018 Chinese Automation Congress (CAC)*. IEEE, pp. 4252–4257. Xi'an, China (2018)
- Mühlig, M., Gienger, M., Steil, J.J.: Interactive imitation learning of object movement skills. *Aut. Robot.* 32(2), 97–114 (2012)
- Evrard, P., et al.: Teaching physical collaborative tasks: object-lifting case study with a humanoid. In: *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pp. 399–404. IEEE, Paris, France (2009)
- Nehaniv, C.L., Dautenhahn, K.: Like me?—measures of correspondence and imitation. *Cybern. Syst.* 32(1–2), 11–51 (2001)
- Russell, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ (2002)
- Gibson, J.J.: *The Ecological Approach to Visual Perception: Classic Edition*. Psychology Press Hove, East Sussex (2014)
- Şahin, E., et al.: To afford or not to afford: a new formalisation of affordances towards affordance-based robot control. *Adapt. Behav.* 15(4), 447–472 (2007)
- Whitehead, S.D., Ballard, D.H.: Learning to perceive and act by trial and error. *Mach. Learn.* 7(1), 45–83 (1991)
- Racca, M., et al.: Learning in-contact control strategies from demonstration. In: *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 688–695. IEEE, Daejeon, Korea (South) (2016)
- Wrede, S., et al.: A user study on kinesthetic teaching of redundant robots in task and configuration space. *JHRI.* 2(1), 56–81 (2013)
- Xu, Y., et al.: Robot teaching by teleoperation based on visual interaction and extreme learning machine. *Neurocomputing.* 275, 2093–2103 (2018)
- Li, J., et al.: An incremental learning framework to enhance teaching by demonstration based on multimodal sensor fusion. *Front. Neurobot.* 14, 55–55 (2020). <https://doi.org/10.3389/fnbot.2020.00055>
- Tang, T., et al.: Teach industrial robots peg-hole-insertion by human demonstration. In: *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 488–494. IEEE, Banff, Canada (2016)
- Yang, C., et al.: A dmeps-based framework for robot learning and generalisation of humanlike variable impedance skills. *IEEE ASME Trans. Mechatron.* 23(3), 1193–1203 (2018)

28. Okamura, A.M., et al.: Haptics for robot-assisted minimally invasive surgery. In: Kaneko M., Nakamura Y. (eds) *Robotics Research*, vol. 66. pp. 361–372. Springer, Berlin, Heidelberg (2010)
29. Lawrence, D.A.: Stability and transparency in bilateral teleoperation. *IEEE Trans. Robot. Automat.* 9(5), 624–637 (1993)
30. Nehaniv, C.L., Dautenhahn, K.: The correspondence problem. In: Nehaniv, C.L., Dautenhahn, K. (eds) *Imitation in Animals and Artifacts*, pp. 41–61, The MIT Press, Cambridge, MA (2002)
31. Song, Y.C., Kautz, H.A.: A testbed for learning by demonstration from natural language and rgb-depth video. In: *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI'12*, 26, 2457–2458. AAAI Press Toronto, Ontario, Canada (2012) <https://doi.org/10.5555/2900929.2901109>
32. Gioioso, G., et al.: An object-based approach to map human hand synergies onto robotic hands with dissimilar kinematics. In: *Robotics: Science and Systems VIII*, pp. 97–104. The MIT Press, Sydney (2012)
33. Rozo, L., Jiménez, P., Torras, C.: A robot learning from demonstration framework to perform force-based manipulation tasks. *Intel Serv. Robot.* 6(1), 33–51 (2013)
34. Chen, X., Jiang, Y., Yang, C.: Stiffness estimation and intention detection for human-robot collaboration. In: *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1802–1807. IEEE, Kristiansand, Norway (2020)
35. Lopes, M., et al.: A computational model of social-learning mechanisms. *Adapt Behav.* 17(6), 467–483 (2009)
36. Osa, T., et al.: An Algorithmic Perspective on Imitation Learning. arXiv preprint arXiv:181106711 (2018)
37. Liu, Y., et al.: Skill transfer learning for autonomous robots and human-robot cooperation: a survey. *Robot. Autonom. Syst.* 103515 128, (2020)
38. Ravichandar, H., et al.: Recent advances in robot learning from demonstration. *Annual Rev. Contr., Robot. Autonom. Syst.* 3, 297–330 (2020)
39. Zhu, Z., Hu, H.: Robot learning from demonstration in robotic assembly: a survey. *Robotics.* 7(2), 17 (2018)
40. Argall, B.D., et al.: A survey of robot learning from demonstration. *Robot. Autonom. Syst.* 57(5), 469–483 (2009)
41. Schaal, S.: Is imitation learning the route to humanoid robots? *Trends Cognit. Sci.* 3(6), 233–242 (1999)
42. Calinon, S.: *Robot Programming by Demonstration*. EPFL Press Lausanne, Switzerland (2009)
43. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Movement imitation with nonlinear dynamical systems in humanoid robots In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 2, pp. 1398–1403. IEEE Washington, DC (2002)
44. Ijspeert, A.J., et al.: Dynamical movement primitives: learning attractor models for motor behaviours. *Neural Comput.* 25(2), 328–373 (2013)
45. Nakanishi, J., et al.: A framework for learning biped locomotion with dynamical movement primitives. In: *4th IEEE/RAS International Conference on Humanoid Robots*, vol. 2, pp. 925–940. IEEE Santa Monica, CA (2004)
46. Zeng, C., Yang, C., Chen, Z.: Bio-inspired robotic impedance adaptation for human-robot collaborative tasks. *Inf. Sci.* 63(170201), 170201 (2020)
47. Wang, N., Chen, C., DiNuovo, A.: A framework of hybrid force/motion skills learning for robots. *IEEE Trans. Cogn. Dev. Syst.* 13, 162–170 (2020)
48. Wang, N., Chen, C., Yang, C.: A robot learning framework based on adaptive admittance control and generalisable motion modelling with neural network controller. *Neurocomputing.* 390, 260–267 (2020)
49. Ijspeert, A.J., Nakanishi, J., Schaal, S.: Learning rhythmic movements by demonstration using nonlinear oscillators. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, pp. 958–963. Lausanne, Switzerland (2002)
50. Pastor, P., et al.: Learning and generalisation of motor skills by learning from demonstration. In: *2009 IEEE International Conference on Robotics and Automation*, pp. 763–768. IEEE, Kobe, Japan (2009)
51. Calinon, S., et al.: Statistical dynamical systems for skills acquisition in humanoid robots. In: *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pp. 323–329. IEEE, Osaka, Japan (2012)
52. Pongas, D., Billard, A., Schaal, S.: Rapid synchronization and accurate phase-locking of rhythmic motor primitives. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 2911–2916. IEEE, Edmonton, Canada (2005)
53. Kober, J., Mohler, B., Peters, J.: Learning perceptual coupling for motor primitives. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 834–839. IEEE, Nice, France (2008)
54. Mülling, K., et al.: Learning to select and generalise striking movements in robot table tennis. *Int. J. Robot. Res.* 32(3), 263–279 (2013)
55. Kober, J., et al.: Reinforcement learning to adjust parametrised motor primitives to new situations. *Aut. Robot.* 33(4), 361–379 (2012)
56. Muelling, K., Kober, J., Peters, J.: Learning table tennis with a mixture of motor primitives. In: *2010 10th IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 411–416. Nashville, USA (2010)
57. Daniel, C., Neumann, G., Peters, J.: Hierarchical relative entropy policy search. In: *Artificial Intelligence and Statistics*, pp. 273–281 (2012)
58. Gams, A., Ude, A.: Generalisation of example movements with dynamic systems. In: *2009 9th IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 28–33. Paris, France (2009)
59. Ude, A., et al.: Task-specific generalisation of discrete and periodic dynamic movement primitives. *IEEE Trans. Robot.* 26(5), 800–815 (2010)
60. Matsubara, T., Hyon, S.H., Morimoto, J.: Learning stylistic dynamic movement primitives from multiple demonstrations. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1277–1283. Taipei, Taiwan (2010)
61. Zhao, Y., et al.: Generating a style-adaptive trajectory from multiple demonstrations. *Int. J. Adv. Rob. Syst.* 11(7), 103 (2014)
62. Rückert, E., d'Avella, A.: Learnt parametrised dynamic movement primitives with shared synergies for controlling robotic and musculoskeletal systems. *Front Comput. Neurosci.* 7, 138 (2013)
63. Ugur, E., Girgin, H.: Compliant parametric dynamic movement primitives. *Robotica.* 38(3), 457–474 (2020)
64. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE.* 77(2), 257–286 (1989)
65. Wang, R., et al.: Dynamic movement primitives plus: for enhanced reproduction quality and efficient trajectory modification using truncated kernels and local biases. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016. pp. 3765–3771. Daejeon, Korea (South) (2016)
66. Wu, Y., et al.: Multi-modal robot apprenticeship: imitation learning using linearly decayed dmp+ in a human-robot dialog system. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1–7. IEEE, Madrid, Spain (2018)
67. Gašpar, T., et al.: Skill learning and action recognition by arc-length dynamic movement primitives. *Robot. Autonom. Syst.* 100, 225–235 (2018)
68. Gams, A., et al.: Coupling movement primitives: interaction with the environment and bimanual tasks. *IEEE Trans. Robot.* 30(4), 816–830 (2014)
69. Khansari-Zadeh, S.M., Billard, A.: Imitation learning of globally stable non-linear point-to-point robot motions using nonlinear programming. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 2676–2683. IEEE, Taipei, Taiwan (2010)
70. Khansari-Zadeh, S.M., Billard, A.: Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Trans. Robot.* 27(5), 943–957 (2011)
71. Khansari-Zadeh, S.M., Billard, A.: Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robot. Autonom. Syst.* 62(6), 752–765 (2014)
72. Calinon, S., Lee, D.: *Learning Control. Humanoid Robotics: A Reference*. Springer, Dordrecht (2017)
73. Reynolds, D.A.: Gaussian mixture models. In: Li S.Z., Jain A.K. (eds) *Encyclopedia of Biometrics*, pp. 827–832. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7488-4_196

74. Lee, S.H., et al.: Learning basis skills by autonomous segmentation of humanoid motion trajectories. In: 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012), pp. 112–119. IEEE, Osaka, Japan (2012)
75. Li, C., Yang, C., Giannetti, C.: Segmentation and generalisation for writing skills transfer from humans to robots. *Cogn. Comput. Syst.* 1(1), 20–25 (2019)
76. Takano, W., Nakamura, Y.: Real-time unsupervised segmentation of human whole-body motion and its application to humanoid robot acquisition of motion symbols. *Robot. Autom. Syst.* 75, 260–272 (2016)
77. Zeebstra, M.J., Calinon, S., Caldwell, D.G.: Variable duration movement encoding with minimal intervention control. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 497–503. IEEE, Stockholm, Sweden (2016)
78. Calinon, S., Pistillo, A., Caldwell, D.G.: Encoding the time and space constraints of a task in explicit-duration hidden Markov model. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3413–3418. San Francisco, CA (2011)
79. Calinon, S., et al.: Learning control. In: Goswami A., Vadakkepat P. (eds) *Humanoid Robotics: A Reference*, pp. 1–52. Springer, Dordrecht (2017)
80. Calinon, S.: A tutorial on task-parameterised movement learning and retrieval. *Intel Serv. Robot.* 9(1), 1–29 (2016)
81. Strang, G., Freund, L.: *Introduction to Applied Mathematics*. American Society of Mechanical Engineers Digital Collection Wellesley, Massachusetts (1986)
82. Forte, D., et al.: On-line motion synthesis and adaptation using a trajectory database. *Robot. Autom. Syst.* 60(10), 1327–1339 (2012)
83. Calinon, S., Alizadeh, T., Caldwell, D.G.: On improving the extrapolation capability of task-parameterised movement models. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 610–616. IEEE, Tokyo, Japan (2013)
84. Alissandrakis, A., Nehaniv, C.L., Dautenhahn, K.: Action, state and effect metrics for robot imitation. In: ROMAN 2006—the 15th IEEE International Symposium on Robot and Human Interactive Communication, pp. 232–237. IEEE, Hatfield, UK (2006)
85. Kruger, V., et al.: Learning actions from observations. *IEEE Robot. Automat. Mag.* 17(2), 30–43 (2010)
86. Paraschos, A., et al.: Probabilistic movement primitives. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, vol. 2, pp. 2616–2624. Lake Tahoe, Nevada (2013). <https://doi.org/10.5555/2999792.2999904>
87. Gomez-Gonzalez, S., et al.: Using probabilistic movement primitives for striking movements. In: 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), pp. 502–508. IEEE, Cancun, Mexico (2016)
88. Atkeson, C.G., Moore, A.W., Schaal, S.: Aha, D.W. (eds) *Locally weighted learning for control*. In: *Lazy Learning*, pp. 75–113. Springer, Dordrecht (1997). https://doi.org/10.1007/978-94-017-2053-3_3
89. Cleveland, W.S.: Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.* 74(368), 829–836 (1979)
90. Vijayakumar, S., D'Souza, A., Schaal, S.: Incremental online learning in high dimensions. *Neural Comput.* 17(12), 2602–2634 (2005)
91. Vlachos, K., Doulgeri, Z.: A control scheme with a novel dmp-robot coupling achieving compliance and tracking accuracy under unknown task dynamics and model uncertainties. *IEEE Robot. Autom. Lett.* 5(2), 2310–2316 (2020)
92. Lauretti, C., Cordella, F., Zollo, L.: A hybrid joint/cartesian dmp-based approach for obstacle avoidance of anthropomorphic assistive robots. *Int. J. Soc. Robot.* 11(5), 783–796 (2019)
93. Chernova, S., Veloso, M.: Confidence-based policy learning from demonstration using Gaussian mixture models. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 1–8. Honolulu, Hawaii (2007)
94. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. B.* 39(1), 1–22 (1977)
95. Hersch, M., et al.: Dynamical system modulation for robot learning via kinesthetic demonstrations. *IEEE Trans. Robot.* 24(6), 1463–1467 (2008)
96. Calinon, S., Kormushev, P., Caldwell, D.G.: Compliant skills acquisition and multi-optima policy search with em-based reinforcement learning. *Robot. Autom. Syst.* 61(4), 369–379 (2013)
97. Yang, C., et al.: Robot learning system based on adaptive neural control and dynamic movement primitives. *IEEE Trans. Neural Netw. Learn. Syst.* 30(3), 777–787 (2018)
98. Stulp, F., Sigaud, O.: Many regression algorithms, one unified model: a review. *Neural Network.* 69, 60–79 (2015)
99. Calinon, S., Guenter, F., Billard, A.: On learning, representing, and generalising a task in a humanoid robot. *IEEE Trans. Syst., Man, Cybern. B.* 37(2), 286–298 (2007)
100. Calinon, S., Billard, A.: Statistical learning by imitation of competing constraints in joint space and task space. *Adv. Robot.* 23(15), 2059–2076 (2009)
101. Zeng, C., et al.: Encoding multiple sensor data for robotic learning skills from multimodal demonstration. *IEEE Access.* 7, 145604–145613 (2019)
102. Kober, J., Peters, J.: Policy search for motor primitives in robotics. *Adv. Neural Inf. Process. Syst.* 21, 849–856 (2008)
103. Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: Proceedings of the Twenty-First International Conference on Machine Learning, pp. 1 (Banff, Alberta, Canada 2004)
104. Li, K., Burdick, J.W.: Human motion analysis in medical robotics via high-dimensional inverse reinforcement learning. *Int. J. Robot. Res.* 39(5), 568–585 (2020)
105. Muratore, F., Gienger, M., Peters, J.: Assessing transferability from simulation to reality for reinforcement learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 1172–1183 (2019)
106. Muratore, F., et al.: Bayesian Domain Randomisation for Sim-to-Real Transfer. arXiv preprint arXiv:200302471, 2020
107. Peng, X.B., et al.: Sim-to-real transfer of robotic control with dynamics randomisation. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–8. IEEE Brisbane, Australia (2018)
108. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* 26(1), 43–49 (1978)
109. VanDenBerg, J., et al.: Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations. In: 2010 IEEE International Conference on Robotics and Automation, pp. 2074–2081. IEEE Anchorage, USA (2010)
110. Zhou, F., De la Torre, F.: Generalized time warping for multimodal alignment of human motion. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1282–1289. IEEE Providence, RI, USA (2012)
111. Maeda, G., et al.: Learning interaction for collaborative tasks with probabilistic movement primitives. In: 2014 IEEE-RAS International Conference on Humanoid Robots, pp. 527–534. IEEE Madrid, Spain (2014)
112. Dang, H., Allen, P.K.: Robot learning of everyday object manipulations via human demonstration. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1284–1289. IEEE Taipei, Taiwan (2010)
113. Meier, F., et al.: Movement segmentation using a primitive library. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3407–3412. IEEE San Francisco, CA (2011)
114. Fox, E., et al.: Sharing features among dynamical systems with beta processes. *Adv. Neural Inf. Process. Syst.* 22, 549–557 (2009)
115. Fox, E.B., et al.: Joint modelling of multiple time series via the beta process with application to motion capture segmentation. *Ann. Appl. Stat.* 8(3), 1281–1313 (2014)
116. Niekum, S., et al.: Learning and generalisation of complex tasks from unstructured demonstrations. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5239–5246. IEEE Vilamoura-Algarve, Portugal (2012)

117. Yang, C., et al.: A learning framework of adaptive manipulative skills from human to robot. *IEEE Trans. Ind. Inf.* 15(2), 1153–1161 (2018)
118. Konidaris, G., et al.: Robot learning from demonstration by constructing skill trees. *Int. J. Robot. Res.* 31(3), 360–375 (2012)
119. Su, Z., et al.: Learning to switch between sensorimotor primitives using multimodal haptic signals. In: Tuci E., Giagkos A., Wilson M., Hallam J. (eds) *International Conference on Simulation of Adaptive Behaviour*, vol. 9825, pp. 170–182. Springer, Cham (2016)
120. Adams, R.P., MacKay, D.J.: Bayesian Online Changepoint Detection. *arXiv preprint arXiv:07103742* (2007)
121. Wettels, N., et al.: Biomimetic tactile sensor array. *Adv. Robot.* 22(8), 829–849 (2008)
122. Yamane, K., Takano, W.: Human motion reconstruction. In: Siciliano B., Khatib O. (eds) *Springer Handbook of Robotics*, pp. 1819–1834. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32552-1_68
123. Aksoy, E.E., et al.: Categorising object-action relations from semantic scene graphs. In: 2010 IEEE International Conference on Robotics and Automation, pp. 398–405. IEEE Anchorage, AK (2010)
124. Wörgötter, F., et al.: A simple ontology of manipulation actions based on hand-object relations. *IEEE Trans. Auton. Mental Dev.* 5(2), 117–134 (2013)
125. Paulius, D., Eales, N., Sun, Y.: A Motion Taxonomy for Manipulation Embedding. *arXiv preprint arXiv:200706695*, 2020
126. Maaten, L.V.D., Hinton, G.: Visualising data using t-sne. *J. Mach. Learning Res.* 9(Nov), 2579–2605 (2008)
127. Bullock, I.M., Ma, R.R., Dollar, A.M.: A hand-centric classification of human and robot dexterous manipulation. *IEEE Trans. Haptics.* 6(2), 129–144 (2012)
128. Feix, T., et al.: The grasp taxonomy of human grasp types. *IEEE Trans. Human-Machine Syst.* 46(1), 66–77 (2015)
129. Abbasi, B., et al.: Grasp taxonomy based on force distribution. In: 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 1098–1103. IEEE New York, NY (2016)
130. Ekvall, S., Kragic, D.: Learning task models from multiple human demonstrations. In: ROMAN 2006-The 15th IEEE International Symposium on Robot and Human Interactive Communication, pp. 358–363. IEEE, Hatfield, UK (2006)
131. Pardowitz, M., et al.: Incremental learning of tasks from user demonstrations, past experiences, and vocal comments. *IEEE Trans. Syst., Man, Cybern. B.* 37(2), 322–332 (2007)
132. Hovland, G.E., Sikka, P., McCarragher, B.J.: Skill acquisition from human demonstration using a hidden markov model In: *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2706–2711. IEEE Minneapolis, MN (1996)
133. Akgun, B., Thomaz, A.: Simultaneously learning actions and goals from demonstration. *Aut. Robot.* 40(2), 211–227 (2016)
134. Patel, M., et al.: Learning object, grasping and manipulation activities using hierarchical hmms. *Aut. Robot.* 37(3), 317–331 (2014)
135. Dindo, H., Schillaci, G.: An adaptive probabilistic approach to goal-level imitation learning. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4452–4457. IEEE Taipei, Taiwan (2010)
136. Nemeč, B., Ude, A.: Action sequencing using dynamic movement primitives. *Robotica.* 30(5), 837 (2012)
137. Kulvicius, T., et al.: Joining movement sequences: modified dynamic movement primitives for robotics applications exemplified on handwriting. *IEEE Trans. Robot.* 28(1), 145–157 (2011)
138. Tikhonoff, V., et al.: Exploring affordances and tool use on the icub. In: 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 130–137. IEEE Atlanta, GA (2013)
139. Lopes, M., Melo, F.S., Montesano, L.: Affordance-based imitation learning in robots. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1015–1021. IEEE San Diego, CA (2007)
140. Do, T.T., Nguyen, A., Reid, I.: An end-to-end deep learning approach for object affordance detection. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 1–5. IEEE, Brisbane, Australia (2018)
141. Heckerman, D.: A tutorial on learning with bayesian networks. In: Holmes D.E., Jain L.C. (eds) *Innovations in Bayesian Networks*, vol. 156, pp. 33–82. Springer, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85066-3_3
142. Melo, F., et al.: A unified framework for imitation-like behaviours. In: *Proceedings on 4th International Symposium Imitation in Animals and Artifacts*, pp. 241–250. Citeseer Newcastle, UK (2007)
143. Ramachandran, D., Amir, E.: Bayesian inverse reinforcement learning. *IJCAI.* 7, 2586–2591 (2007)
144. Kroemer, O., et al.: A kernel-based approach to direct action perception. In: 2012 IEEE international Conference on Robotics and Automation, pp. 2605–2610. IEEE Saint Paul, MN, USA (2012)
145. Zhang, J., et al.: Grasping novel objects with real-time obstacle avoidance. In: Ge S. et al. (eds) *International Conference on Social Robotics*, vol. 11357, pp. 160–169. Springer, Cham (2018)
146. Lin, H., et al.: Adaptive fuzzy Gaussian mixture models for shape approximation in robot grasping. *Int. J. Fuzzy Syst.* 21(4), 1026–1037 (2019)
147. Lee, D., Ott, C., Nakamura, Y.: Mimetic communication model with compliant physical contact in human-humanoid interaction. *Int. J. Robot. Res.* 29(13), 1684–1704 (2010)
148. Maeda, G.J., et al.: Probabilistic movement primitives for coordination of multiple human-robot collaborative tasks. *Aut. Robot.* 41(3), 593–612 (2017)
149. Su, H., et al.: Human activity recognition enhanced robot-assisted minimally invasive surgery. In: Zeghloul S., Laribi M., Sandoval Arevalo J. (eds) *International Conference on Robotics in Alpe-Adria Danube Region*, vol. 84, pp. 121–129. Springer, Cham (2020)
150. Whiten, A., et al.: How do apes ape? *Anim. Learn Behav.* 32(1), 36–52 (2004)
151. Whiten, A., et al.: Emulation, imitation, over-imitation and the scope of culture for child and chimpanzee. *Phil. Trans. R. Soc. B.* 364(1528), 2417–2428 (2009)
152. Ragaglia, M., et al.: Robot learning from demonstrations: emulation learning in environments with moving obstacles. *Robot. Autonom. Syst.* 101, 45–56 (2018)
153. Bohg, J., et al.: Interactive perception: leveraging action in perception and perception in action. *IEEE Trans. Robot.* 33(6), 1273–1291 (2017)
154. Bajcsy, R., Aloimonos, Y., Tsotsos, J.K.: Revisiting active perception. *Auton Robot.* 42(2), 177–196 (2018)
155. Qiao, H., et al.: Biologically inspired visual model with preliminary cognition and active attention adjustment. *IEEE Trans. Cybern.* 45(11), 2612–2624 (2014)
156. Qiao, H., et al.: Introducing memory and association mechanism into a biologically inspired visual model. *IEEE Trans. Cybern.* 44(9), 1485–1496 (2013)
157. Salaris, P., et al.: Online optimal active sensing control. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 672–678. IEEE, Singapore (2017)
158. Salaris, P., et al.: Online optimal perception-aware trajectory generation. *IEEE Trans. Robot.* 35(6), 1307–1322 (2019)
159. Shiarlis, K., Messias, J., Whiteson, S.: Inverse reinforcement learning from failure., In: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems Vol. 9*, pp. 1060–1068. Singapore (2016). <https://doi.org/10.5555/2936924.2937079>
160. Grollman, D.H., Billard, A.: Donut as i do: learning from failed demonstrations. In: 2011 IEEE International Conference on Robotics and Automation, pp. 3804–3809. IEEE, Shanghai, China (2016)
161. Chernova, S., Veloso, M.: Interactive policy learning through confidence-based autonomy. *J. Artif. Intell. Res.* 34, 1–25 (2009)

162. Bobu, A., et al.: Quantifying hypothesis space misspecification in learning from human-robot demonstrations and physical corrections. *IEEE Trans. Robot.* 36(3), 835–854 (2020)
163. Bajcsy, A., et al.: Learning from physical human corrections, one feature at a time. In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 141–149. Chicago, IL (2018)

How to cite this article: Guan, Y., Wang, N., Yang, C.: Review of the techniques used in motor-cognitive human-robot skill transfer. *Cogn. Comput. Syst.* 3(3), 229–252 (2021). <https://doi.org/10.1049/ccs2.12025>