9 Machine Learning for Cognitive Robotics

Tetsuya Ogata, Kuniyuki Takahashi, Tatsuro Yamada, Shingo Murata, and Kazuma Sasaki

9.1 Introduction

In recent years, the technology of deep learning has been confirmed to be effective in various fields, such as image recognition, speech recognition, and language processing, and various applied methods have been proposed (LeCun et al. 2015).Deep learning generally refers to hierarchical neural network models of multiple layers with large dimensional inputs. One of the important characteristics of this approach is that the sensory features that human experts should typically design and select based on their knowledge and experience—for example, for a computer vision algorithm—can be self-organized through the learning process. This enables the training of deep-learning models, as long as the teaching labels are given to the target data. Data sets with high-dimensional signals can be used for training. This property enables deep learning to handle various types of data, such as images, sounds, and languages, differently from the way these problems have been treated in other research areas. The performance of deep-learning models is close to that of conventional methods and can in some modalities achieve performance superior to human abilities.

There are several methods of deep learning. One of the representative ones is with the use of autoencoders. An autoencoder is a model to learn so that input and output are the same. Once input data are provided, it is classified as "unsupervised learning" because it is simply learned to reproduce it. For example, an image input of one thousand dimensions is compressed to tens of dimensions in the middle layer, and then it is decompressed to restore the original image. Here, the low-dimensional representation in the middle layer could be used for image recognition by relearning (fine-tuning).

Convolutional neural networks (CNNs) are the current driving force of deep learning. In a multilayer network, the connections between layers are usually connected with full (dense) connection patterns. In CNNs, however, the convolution layer and the pooling layer have sparser connectivity with repeated and shared parameters, and a dense connection layer is typically added at the end. In the case of image recognition, the change of position does not affect the recognition result thanks to the convolution and pooling structure. Rather, it is important to capture a subset of features. Therefore, a CNN has small neural networks (kernels) that take only certain areas of the input image. For example, the values of three-by-three pixels are multiplied by the weights and compressed into a single value. This operation is called convolution. The kernel can be designed in many different sizes

and shapes. Since the kernel reacts to certain features in the image, it slides over the entire image area to produce a compressed representation of the image. And the pooling layer compresses the image size to save memory. With repeated convolution and pooling, the final feature representation is acquired, and finally, the recognition result is output through full connection.

There is also a set of deep networks based on recurrent neural networks (RNNs). An RNN is a neural network that does not directly connect inputs to outputs in a feedforward way, as it also has feedback connections. Even if the inputs are in a similar state, the output can change according to the internal neural condition. In RNNs, not only the connection weights but also the internal states are trained to improve prediction accuracy. RNNs have an advantage for time-series learning. However, it can be difficult to obtain good performance with an RNN because it has the same problem as deep learning, gradient vanishment. Errors are eliminated because it is difficult to propagate output errors to past steps if the learning sequences are long. To solve this problem, a new type of RNN has been developed that has multiple types of neurons. Some neurons retain their internal state in the long term (slow neurons). Some neurons change their internal state in the short term (fast neurons). These are called multitimescale neurons. As a result, fast neurons learn the short time series of input value, and long-term neurons learn the sequence of these short time series. A multitimescale RNN (MTRNN; Yamashita and Tani 2008) uses continuous neurons, with internal states represented by continuous values. By adjusting the time constant of the neuron change, it is possible to create fast and slow neurons. Another commonly used type of deep RNN is the long short-term memory (LSTM; Hochreiter and Schmidhuber 1997). In addition to the weight of the current input, the LSTM neuron learns whether to accept it (Input Gate), whether to output it (Output Gate), whether to keep the current state (Forget Gate), and other various outputs used by the error back-propagation method. LSTM models now perform well, especially in natural language processing.

Various deep-learning models and applications can be used for different modalities, such as vision, audio, and tactile modalities, in cognitive robotics. Furthermore, due to the fact that various modalities can be handled in a similar framework, these can lead to the *multimodal* applications of deep learning. In particular, a robot working in the real world is a typical multimodal system with cameras, microphones, distance sensors, tactile sensors, and actuators.

This chapter provides an overview of the research that focuses mainly on the applications of deep learning for robotics. In subsequent chapters focusing on specific cognitive robotics capabilities, more examples of deep-learning models will be discussed. The first part of this chapter contains three subsections concerning the learning of visual, tactile, and language modalities and skills. The subsequent sections focus on behavior learning related to imitation learning and on reinforcement-learning approaches. The final section discusses the possibilities of deep learning and its future prospects.

9.2 Deep-Learning Model for Modality Application

9.2.1 Robot Vision

The most natural application of deep-learning technology is in the research field of robot vision. For example, Lenz, Lee, and Saxena (2015) proposed a method to output the posi-

tion and direction (four dimensions) of a hand to grasp from a distance an image of an object. Using a CNN, Yang, Li, et al. (2015) identified forty-eight kinds of objects and six types of grasping directly from a YouTube video of a human cooking and applied them to the motion of a robot.

Redmon and Angelova (2015) also used CNNs to predict the grasping position of an object from a three-dimensional RGB-D image consisting of color (RGB) and depth (D) data. Concretely, for an RGB image of 224×224 pixels, a grasping position vector of an object is labeled by human. The grip position vector has six dimensions, including the rectangular shape of the center coordinate, the rotation angle, and the grip position of the vector (figure 9.1).

The success rate is calculated using two conditions: 1) the rotation angle error is within 30°, and 2) the overlapping area ($A \cap B$) with respect to the total area ($A \cup B$) is over 25 percent. However, these criteria do not evaluate the actual motions of the robot. The success rate of grasping using a real robot is not always high.

What is important here is that information regarding the object grasping cannot be obtained from just the image of the object. The learning process should reflect the hardware (body) of the robot and the effects of the possible motion. For example, although the grip position vector shown in figure 9.1 is a feature for a gripper, there is no guarantee that it is a sufficient and optimum feature quantity for general gripper mechanisms. When extracting a region for grasping an object, a robot should consider the physical features of the target object, such as the weight, center-of-mass, surface friction, shape change, and so on. Even if the same hand is used, grasping should be changed in various ways depending on the hand size, the payload, the direction of the approach (trajectory), and more. That is, the learning process should include not only the image of the object but also the motion generated by the robot hardware.

9.2.2 Tactile Learning

Learning the tactile sense is important for robots to allow them to obtain physical information while interacting with environments. This can be useful for operations such as

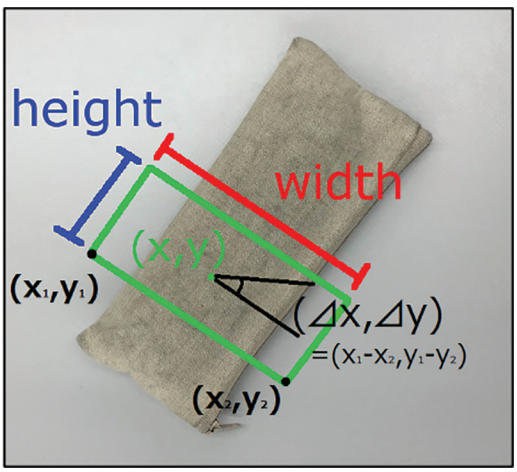

Figure 9.1
The grip position vector.

walking, physical contact with people, and object manipulation. The improved availability of tactile sensors has enabled research in this field to flourish (see chapter 8). Prior to the use of learning-based approaches, tactile sensor data were only used with handcrafted features (Yang, Sun, et al. 2016) or to trigger specific actions (Yamaguchi and Atkeson 2016). However, such methods may not scale well as tactile-sensing technology advances—for example, when a higher resolution and a larger amount of data are necessary, or as task complexity increases. By using learning-based approaches, in particular deep learning, it is now possible to handle tasks such as image recognition and natural language processing, which involve high-dimensional data and were previously difficult to process. Moreover, deep-learning approaches have recently been applied to tactile sensing, such as object recognition (Schmitz et al. 2014; Baishya and Bäuml 2016), tactile properties recognition (Gao et al. 2016; Yuan, Wang, et al. 2017), and grasping (Calandra et al. 2018).

In recent years, within research involving tactile sensors, object manipulation using robotic hands has been gaining attention since manipulation is one of the fundamental functions for a robot to perform various tasks such as tidying up, cooking, and folding clothes. In this chapter, the recent development of tactile learning and the following four categories of the object manipulation process are described: 1) object recognition, 2) grasping, 3) in-hand object pose estimation, and 4) in-hand object manipulation.

Types of tactile sensors

Many different tactile sensors have been developed to improve manipulation in robotic hands (Dahiya et al. 2013; see also chapter 8 for a detailed analysis). The majority of these sensors, however, belong to one of the following three categories:

– Multitouch sensors that can only sense force information along one axis—namely, perpendicular to the surface of the sensor. These types of sensors are known as pressure sensors (Ohmura, Kuniyoshi, and Nagakubo 2006; Iwata and Sugano 2009; Mittendorfer and Cheng 2011; Fishel and Loeb 2012).

– Three-axis sensors that can sense both shear and pressure forces but are only single touch (Paulino et al. 2017).

– Three-axis sensors for both shear and pressure forces that are multitouch (Tomo et al. 2018; Yamaguchi and Atkeson 2016; Yuan, Dong, and Adelson 2017).

At the time of writing, there are only three sensors of the last type: uSkin (Tomo et al. 2018), Finger Vision (Yamaguchi and Atkeson 2016), and GelSight (Johnson and Adelson 2009; Dong, Yuan, and Adelson 2017; Yuan, Dong, and Adelson 2017). The uSkin measures the deformation of silicon during contact by monitoring changes in the magnetic fields of magnets in silicon. The sensor is able to measure both pressure as well as shear force per sensor unit for multiple contact points.

Instead of a magnet, the Finger Vision is a vision-based tactile sensor, meaning that it uses a camera to capture and measure the deformation of its attached marker during contact with a surface. In addition to contact sensing, it can also function as a proximity sensor since the Finger Vision uses transparent silicon.

The GelSight can be manufactured by covering the silicon surface of the Finger Vision with another layer of silicon that contains aluminum powder. The aluminum powder highlights the deformation of the silicon layer more clearly and hence allows for richer informa-

tion during sensing. The GelSight can be duplicated easily and is suitable for deep learning because it also uses a camera, so existing image-processing techniques can be employed to process the data. Therefore, the GelSight has become increasingly popular in research (Calandra et al. 2018; Tian et al. 2019; Zhang et al. 2020; Anzai and Takahashi 2020).

Object recognition

One of the main approaches to recognizing the type of object in a robotic hand (Schmitz et al. 2014), its materials (Baishya and Bäuml 2016; Yuan, Zhu, et al. 2017), and its properties (Gao et al. 2016), using touch and image information, is its classification through supervised learning using manually designed labels. Baishya and Bäuml (2016) and Yuan, Zhu, et al. (2017) estimated the hardness of an object as a continuous value using a tactile sensor through supervised learning. In these approaches, however, the results of class labels and their degrees depend completely on the manner in which these class labels are designed. On the other hand, one of the approaches without manually specified labels represents tactile properties in a continuous space using an unsupervised-learning approach (Takahashi and Tan 2019).

Grasping

A different use case is shown in Calandra et al. (2018), in which they utilized deep reinforcement learning and combined input data acquired from a tactile sensor with images to grasp objects using a parallel gripper, which improved their success rate in grasping experiments compared to only vision. Wu et al. (2019) showed similar results using a multifinger hand. By using a tactile sensor, the stability of a grasp can be evaluated and improved upon regrasping (Calandra et al. 2018; Wu et al. 2019; Hogan et al. 2018).

In-hand object pose estimation

In order to realize the target object pose, it is necessary to be able to estimate the current object posture. Object pose estimation is a well-studied problem in computer vision. Many researchers have been developing methods using depth data (point cloud) or RGB-D data (Choi and Christensen 2012; Aldoma et al. 2012; Choi et al. 2012). Classical approaches with depth data are mainly based on point cloud matching methods, such as iterative closest point (ICP; Rusinkiewicz and Levoy 2001). Since this method requires three-dimensional (3D) models of objects, unknown objects cannot be handled. In the state-of-the-art research in pose estimation, methods that do not require 3D models have been studied using deep learning (Schwarz, Schulz, and Behnke 2015; Hodaň et al. 2018; Hu et al. 2019).

These methods, however, are challenging to apply to in-hand manipulation because of occlusion by the hand in the image or depth data. Since tactile sensors can observe the contact state despite a visual occlusion, they are suitable for overcoming this challenge. Some research has performed object pose estimation with tactile sensors by means of a model-based approach using a 3D model (Bimbo et al. 2016) and without using a 3D model (Anzai and Takahashi 2020).

To overcome challenges such as occlusions or lack of sufficient information, one can use multiple sensors to try to obtain an improved perception of the environment or situation. In this case it is of great importance to know which modals can be trusted in a given situation—in other words, how reliable a given sensor modal is. For example, if a vision sensor is impaired, one should give its data less importance than other sensor modals. It is difficult, however, to determine sensor modal reliability through rule-based methods.

Anzai and Takahashi (2020) proposed a network that can autonomously determine the reliability of each modal.

In-hand object manipulation

To manipulate a grasped object to a target posture is one of the most challenging tasks. Analytical approaches exist, but they come with limitations, such as the known object model and the rigid object (Han et al. 1997; Han and Trinkle 1998). In learning-based approaches, manipulation is performed by predicting the state of the tactile sensor for the motion of a robot's end effector (Tian et al. 2019; Li et al. 2014; Funabashi et al. 2018). Since object manipulation with a multifingered hand is still challenging, most of these studies are simple tasks and take place in experimental settings, with a few exceptions (e.g., Falco et al. 2018).

9.2.3 Learning of Language Grounding in Robot Behavior

Natural language is the most powerful tool for expressing our requests to other agents. Service robots must be able to understand natural language to flexibly respond to human requirements or to effectively work together with humans. However, to arbitrarily design mapping between language, which is a discrete system, and the referents in the real world, which is a continuous and dynamical system, is notoriously difficult, as stated in the symbol grounding problem (Harnad 1990). The meanings of linguistic expressions also greatly depend on the current context that an agent is situated in. For instance, to respond to the instruction "grasp the red ball," a robot is required to generate different trajectories of joint angles in accordance with the position of the red ball. Unlike most situations in industrial factories, our living environment is highly changeable and open ended; new situations almost always differ from the previous ones. It is almost impossible to make explicit rules that can handle all possible situations in a top-down manner.

Many attempts have been made to get robots to learn grounding relationships from their own experiences in a bottom-up manner. Here we review existing studies that consider the learning of grounding relationships between language and behavior in robots. In particular, we discuss the two main approaches to language grounding: probabilistic modeling and neural networks. See also chapter 20 for more details on deep-learning approaches to robot language models.

Probabilistic modeling

One way to model the relationships between language and other modalities is to model them as probabilistic relationships. For example, Inamura et al. (2004) utilized hidden Markov models (HMMs) to recognize and generate human motions. In their framework, protosymbols, which represent a specific motion pattern, emerged in the learning process. Nishihara, Nakamura, and Nagai (2017) utilized a multimodal latent Dirichlet model (MLDA) for a robot to learn object concepts that connected multimodal information consisting of co-occurring word, auditory, visual, and tactile data. Tellex et al. (2011) proposed a framework called generalized grounding graphs, which dynamically instantiated a graphic model depending on the semantic structure of linguistic commands, and they then inferred appropriate plans for navigation and manipulation in the graph.

One advantage of probabilistic models is their high intelligibility. In the case of graphic models, each node in the graph is designed as a meaningful element. Therefore, it is easy

to understand what kind of inference is performed by the model. However, a probabilistic model that has the capability of dealing with long-term dependencies sufficiently has not yet been developed.

Neural networks

On the other hand, methods that model language grounding deterministically also exist. One popular method is neural networks, such as with RNNs. Sugita and Tani (2005) proposed a trainable architecture that consisted of two neural networks—one of which was for language and the other, robot behavior—with a small number of shared nodes called parametric bias (PB). The model learned to embed the relationships between language and behavior in topological organization in the PB space. Ogata et al. (2007) employed a similar architecture to learn the bidirectional mapping between language and robot behavior. Heinrich and Wermter (2014) proposed a model that connected three RNNs. Each RNN was specialized for vision, proprioception, and language, respectively, but they were connected to each other. After learning, the model could generate sentences that described robot motions as a sequence of characters. Stramandinoli, Marocco, and Cangelosi (2017) utilized a Jordan-type RNN (Jordan 1997) to ground abstract words (e.g., use and make) in robots' sensorimotor experiences. The abstract words were learned by recalling the meanings of previously learned basic words and combining them.

An advantage of neural networks is that by introducing recurrent connections and some gating mechanism, such as LSTM (Hochreiter and Schmidhuber 1997), they can achieve a much higher performance in learning temporal structure with long-term dependency without a priori knowledge. One disadvantage of neural networks is that it is difficult to understand their behavior since their representations in hidden layers are in a distributed form. Recently, some studies have proposed methods to visualize the internal behavior of neural networks (Bach et al. 2015; Smilkov et al. 2017) and to make their representations more intelligible (Chen et al. 2016; Xu et al. 2015). The following introduces a recent study that proposed an RNN-based framework to ground language in robot behavior.

Yamada, Matsunaga, and Ogata (2018) attempted to bidirectionally convert language and robot behavior by utilizing two coupled recurrent autoencoders (RAEs; figure 9.2): one RAE coped with language, and the other dealt with behavior.

Each RAE consists of an encoder RNN and a decoder RNN. The encoder RNN compresses a time series (a sentence or a behavioral sequence; $x_1, x_2, \ldots, x_T$) into a fixed-dimensional feature vector z:

$$z = EncoderRNN(x_1, x_2, \ldots, x_T)$$

The decoder RNN produces a sequence by recursively decoding the feature vector:

$$(y_1, y_2, \ldots, y_T) = DecoderRNN(z)$$

The RAE is trained to reconstruct the original sequence through the feature vector—namely, identity function. The loss function is as follows:

$$L = \frac{1}{T}\sum_{t=1}^{T} \psi(x_t, y_t).$$

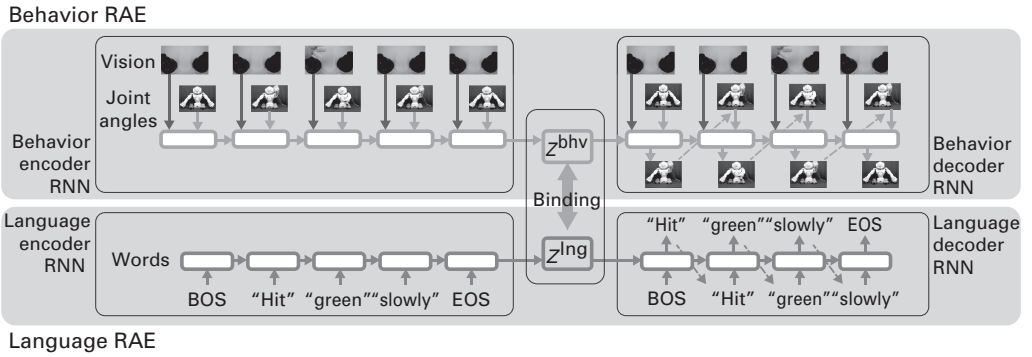

Figure 9.2
Two coupled RAEs to bidirectionally convert language and robot behavior. *Source:* Adapted from Yamada, Matsunaga, and Ogata 2018.

The detail of loss function ψ at each time step depends on the modality. In the learning process, the language RAE and the behavior RAE are optimized to extract the important features of time series data in each modality.

In addition, the whole system is trained in such a way that the feature vectors of co-occurring language and robot behavior get closer to each other, and the feature vectors of unpaired language and behavior grow more distant from each other. With this constraint, this coupled RAE system is able to bidirectionally convert language and behavior through the latent feature space. Producing a behavior sequence in response to a sentence is realized by using the encoder of the language RAE to encode the sentence and the decoder of the behavior RAE to expand the feature vector. In contrast, producing a sentence description of a robot behavior is realized by having the encoder of the behavior RAE encode a behavioral sequence and having the decoder of the language RAE expand the feature vector.

Figure 9.3 shows the latent feature spaces organized by learning in this robot experiment. Each point corresponds to a sentence in the left panel and to a behavioral sequence in the right panel. It can be seen that the behavioral sequences were actually bound with their paired sentences. Here, it is worth noting that because the behavior RAE also receives vision input, the model could respond to the same sentence by producing different joint-angle trajectories depending on the current contexts.

9.3 Imitation Learning (Predictive Learning)

Imitation learning, also referred to as learning from demonstration (LfD) or programming by demonstration (PbD), is a learning-based approach that enables robots to acquire skills (or infer policies) for action generation from a set of expert demonstrations representing the robots' sensorimotor experiences. Imitation learning is mostly performed by a scheme of predictive learning in which robots are required to learn to predict the (sensory-)motor state at the next time step from the sensory(-motor) state at the current time step. This is a more data efficient approach in comparison to the reinforcement learning to be introduced in a forthcoming section. Imitation learning is a particularly useful approach when the use of a reinforcement-learning algorithm is unrealistic due to the difficulty in design-

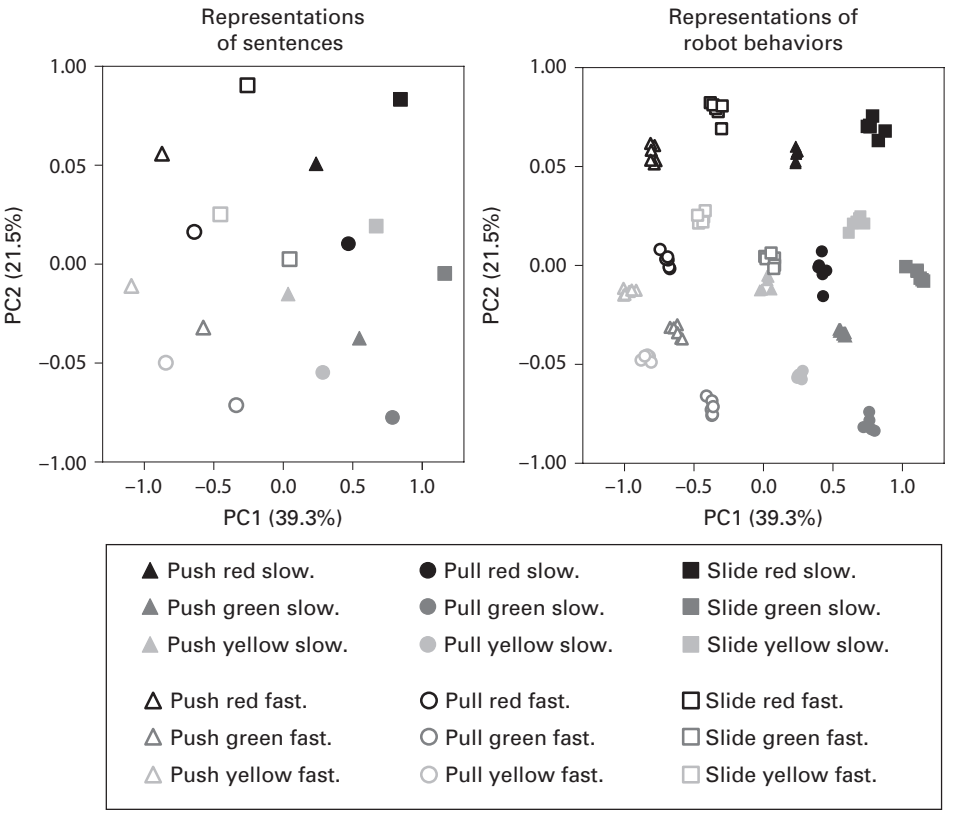

Figure 9.3
Latent representations of language and robot behavior by the coupled RAEs. *Source:* Adapted from Yamada, Matsunaga, and Ogata 2018.

ing a reward function and in performing a massive amount of exploration (with real robots). In the context of (cognitive) robotics and robot learning, imitation learning includes the following two cases: 1) learning from sensorimotor experiences and 2) learning from sensorimotor experiences by observing another agent's demonstrations. In both cases, it is necessary to provide demonstrations about robot performance during the learning process via kinesthetic teaching or teleoperation by a human demonstrator. The difference between them is whether or not the sensory (mainly visual) experiences include demonstrations about the performance of another agent, typically a human. Namely, in the second case robots are required not only to learn to generate their own actions but also to map an observed other's actions to their own by inferring what to perform and how to perform. This is much closer to the original meaning of imitation by humans and animals in the context of cognitive science (Meltzoff and Moore 1977).

There are several machine-learning approaches for performing imitation learning, such as neural networks (e.g., CNNs and RNNs); probabilistic models such as the combination of a Gaussian mixture model and Gaussian mixture regression (e.g., Calinon, Guenter, and Billard 2007); hidden Markov models (e.g., Inamura et al. 2004); and dynamical systems (e.g., dynamic movement primitives in Ijspeert, Nakanishi, and Schaal [2002] and Ijspeert

et al. [2013]). In this section, we focus particularly on neural network–based approaches (refer to review papers for other approaches, such as Argall et al. [2009] and Billard et al. [2008]). In what follows, several studies of the above two cases of imitation learning are examined. In addition, their extensions with deep-learning approaches, such as the use of deep autoencoders for visual feature extraction from raw images and LSTM for learning long-term dependencies, are introduced. Finally, related advanced topics, including one-shot imitation learning and self-supervised learning from play data, are also briefly discussed.

9.3.1 Imitation Learning from Own Sensorimotor Experiences

Ito et al. (2006) studied the learning of primitive actions for object manipulation by using an RNN with parametric bias (RNNPB). In their experiment, the sensorimotor experiences of a small humanoid robot QRIO for ball handling were first collected via kinesthetic teaching. There were two different primitive actions for ball handling, including: 1) rolling a ball from the left to right sides and vice versa (referred to as ball-rolling action hereafter) and 2) lifting the ball and letting it fall to the ground (referred to as ball-lifting action hereafter). Sensorimotor experiences consisted of time-series data items (or trajectories) of visual information represented as ball position and action information represented as joint angles of both arms. The robot with an RNNPB was required to learn to predict the visuomotor state at the next time step given the state at the current time step. Through this learning process, the various primitive actions were represented by the difference in optimized PB vectors. Namely, once a PB vector corresponding to the ball-rolling action is set into the network, the robot generates the ball-rolling action, and once the other vector corresponding to the ball-lifting action is set, the robot generates the ball-lifting action. This means that different primitive actions were acquired as multiple limit cycle attractors in the RNNPB.

One of the important points of this experiment is that the PB vector during action generation after the learning phase was also optimized online in the direction of minimizing prediction errors computed during a time window of immediate past time steps. This iterative optimization of the PB vector enabled the robot to adapt to unexpected situational changes. For example, consider a situation in which the PB vector for the ball-rolling action is set, and the robot is generating the corresponding action. Then, an experimenter suddenly disturbs the ball movement between the left and right sides, and the ball movement stops at the center front of the robot. Before the disturbance, the robot was predicting that the ball would be moving between the left and right sides as a consequence of its own action generation. However, due to the disturbance that stopped the ball movement, the robot feels a discrepancy between the anticipated and actual situations or prediction errors. The only solution to minimize these errors is to switch the originally set PB vector to the other one that generates the ball-lifting action. This switching of the PB vector enables the robot to minimize the generated prediction errors and to perform stable action generation again. The important point of this phenomenon is that the robot had never learned to switch between the different primitive actions. Thanks to the simple computational principle of the so-called prediction error minimization (Nagai 2019), the robot realized adaptive action generation. This is closely related to the active inference scheme based on the free energy principle (Friston et al. 2010).

Chen, Murata, et al. (2016) extended the framework to an interaction between two NAO robots. In their experiment, each robot with an RNNPB first learned a set of primitive actions for ball manipulation with a human experimenter. The learned primitive actions were dependent on the ball movement such that when the ball was heading toward the right side of a robot, the robot was required to hit the ball with its right hand. After the learning phase, the robots faced each other and were required to perform a ball-play interaction. Because the experiment was performed in the real world, with some fluctuations such as the friction between the ball and a table, sometimes the ball dynamics suddenly changed in an unpredictable manner. In such a situation, prediction errors arose in both the robots, and these errors triggered the PB vector of each robot, optimizing it to fit the current situation. This dual optimization of the PB vector of each robot enabled spontaneous action switches without any training.

In the former examples using an RNNPB, the switch between primitive actions was triggered by environmental changes. Next, we consider how such switching can be intentionally generated by learning action sequences consisting of combinations of primitive actions. Yamashita and Tani (2008) and Nishimoto and Tani (2009) tackled this issue by using the MTRNN introduced above. In a manner similar to the RNNPB experiments introduced earlier, they first collected visuomotor experiences of the QRIO robot via kinesthetic teaching. The recorded sequences were more complex than the first study above. For example, in one sequence the robot reached for an object from a home position and then moved the object up and down three times before finally moving it back to the home position. Specifically, each sequence contained multiple primitive actions such as reaching for and moving the object, and the robot was required to switch or repeat such actions. The robot with an MTRNN performed predictive learning of these complex and longer visuomotor experiences by utilizing the sensitivity of the initial conditions of the slow dynamics layer of the MTRNN. After the learning phase, the robot succeeded in generating the learned action sequences. Analysis of the fast and slow dynamics layers revealed that primitive actions were represented in the fast dynamics layer, and the combinations of these primitives (sequence information) were represented in the slow dynamics layer thanks to the self-organized functional hierarchy.

Namikawa, Nishimoto, and Tani (2011) extended this experimental setup and considered how probabilistic transitions among primitive actions could be learned. In the same manner as the former cases, they first recorded visuomotor experiences for an object manipulation in which the QRIO robot moved an object from center to left, from left to center, from center to right, and so on via kinesthetic teaching. These transition patterns were determined probabilistically, and they investigated whether such sequences with probabilistic transitions could be learned by a deterministic MTRNN. The robot after the learning phase reconstructed a demonstrated visuomotor sequence from the beginning by setting an optimized initial state of the slow dynamics layer, but the sequence gradually changed from the learned one. The analysis of the generated action sequences demonstrated that the transition probabilities were still preserved in newly generated sequences. The analysis of each layer of the MTRNN revealed that in the same way as in the former studies (Yamashita and Tani 2008; Nishimoto and Tani 2009), different types of information were stored in each layer. One more interesting phenomenon is that only the slow

dynamics layer exhibited chaotic dynamics with a positive Lyapnov exponent, which led to the reconstruction of the probabilistic transitions by deterministic neural dynamics.

In the experiments conducted before the deep-learning era, such as that just described, the experimental setup was simplified so that, for example, the visual information was just the object position. Here, some scaled-up experiments are introduced that deal with high-dimensional raw visual images by using deep-learning approaches such as a deep (convolutional) autoencoder.

Noda et al. (2014) conducted a study on the integrative learning of multimodal information such as vision, auditory, and motor data using a combination of deep autoencoders for feature extraction and temporal processing. As in the previous studies, they first collected sensorimotor experiences of the NAO robot via kinesthetic teaching. Then, low-dimensional features of high-dimensional raw visual images and auditory information were extracted by using the respective deep autoencoders. The extracted visual and auditory features were concatenated with joint angle information. They used another deep autoencoder called a time-delay neural network (TDNN) that received a time window of the multimodal information and outputs its reconstruction. By using this framework, they realized action generation by prediction and retrieval, such as visual retrieval from auditory and joint angle information using high-dimensional sensorimotor states.

Yang et al. (2017) extended this framework to the human-size industrial robot Nextage and performed a towel-folding task. It is known that towel handling is a challenging task in robotics because modeling a deformable object is difficult. They recorded visuomotor experiences via teleoperation using a 3D mouse. In their experiment, the normal autoencoder for visual feature extraction was replaced with a deep convolutional autoencoder (ConvAE). They realized repeatable towel folding with a high success rate after the learning phase. Kase and colleagues replaced the TDNN used in the above two experiments with RNN-based architectures, an MTRNN (Kase et al. 2018) and an LSTM (Kase et al. 2019). These replacements realized much longer and complex task executions such as put-in-the-box and skewering thanks to their characteristics of functional hierarchy and long short-term memories.

9.3.2 Imitation Learning from Observing Another Agent's Demonstrations

When learning from observing another agent's action generation, robots need to infer what to perform and how to perform. Arie et al. (2012) considered this issue by using an MTRNN. In their experiment, a small humanoid robot, HOAP-3, learned a set of visuomotor sequences consisting of multiple primitive actions. For example, in one sequence the robot first reached for an object from a home position, then moved the object right, then knocked the object over, and finally moved the object back to the home position. Note that the robot learned not only its own action generation but also how to map an observed action of human performance to its performance. There were four primitive actions, including R (moving the object to the right), L (moving the object to the left), K (knocking over the object), and U (moving the object upward). The robot first learned three different types of visuomotor sequences (RK, UK, and UL) produced by itself and the experimenter. After these sequences, the robot was subjected to the demonstration of only the human's performance for the RL sequence. The robot was evaluated on whether it could generate its

own action for the RL sequence, which had not been learned, by mapping the observed demonstration of human performance to its own performance.

The slow dynamics layer of the MTRNN had two special neural units whose initial conditions were optimized to be the same values when the demonstrations were the same patterns, regardless of the generation of robot performance and the observation of human performance. The other two units in the slow dynamics layer served as a PB vector that discriminated the self-mode (generation of robot performance) and the other-mode (observation of human performance) by assigning a particular value for each (one for the self-mode and minus one for the other-mode). In the evaluation after the additional learning phase, an action-specific initial state for the demonstration of human performance for the RL sequence was set, and the demonstrator-specific PB vector was switched to the self-mode. This enabled the robot to generate the unlearned combinatory actions for the RL sequence.

Nakajo et al. (2015) considered another important topic concerning the acquisition of viewpoint representation. Humans can understand what action is demonstrated by another regardless of a difference in viewpoint. Acquiring such an ability is useful for robots because the demonstration of human performance can be provided from any direction. However, this is not straightforward for robots because the visual information from the demonstration of human performance from different viewpoints is distinct. They used an MTRNN for learning the demonstrations of object manipulation for both the robot and human performances. In their experiment, a human demonstrator performed actions from multiple viewpoints. They provided constraints on the initial state optimization by introducing a subnetwork for representing viewpoints. Their analysis of the initial state space of the subnetwork revealed that the positional relationship of the viewpoints was self-organized in the space. In their experiment, although the structured representation of viewpoints was self-organized, how to map the demonstration of human performance provided from multiple viewpoints to the same robot performance remained an issue.

To tackle this issue, Nakajo et al. (2018) extended the experiment by introducing a sequence-to-sequence (seq2seq) deep-learning approach that has been widely used, especially in machine translation (Sutskever, Vinyals, and Le 2014). The seq2seq framework consists of an RNN-based encoder-decoder architecture. In the machine translation, the encoder RNN receives source sentence information, such as an English sentence, sequentially and transforms it into a fixed-dimensional vector. The decoder receives this vector and transforms it to target sentence information, such as a Japanese sentence. By referring to this information processing of the seq2seq framework, they first encoded visual features of video information about the demonstration of human performance extracted by a convolutional encoder with an MTRNN. Then an achieved fixed-dimensional vector was transformed to the robot's action generation. After a learning phase, the robot was able to map the demonstration of human performance provided from an unlearned viewpoint to its own action generation. The analysis of each layer of the MTRNN shows the representation of actions, objects, and viewpoints. More specifically, after the demonstration of human performance, the fast dynamics layer represented viewpoint information, and the slow dynamics layer represented action and object information without any viewpoint information. The key point for the success of mapping from unlearned human demonstration

to robot performance is that the slow dynamics layer acquired the viewpoint-invariant representation about the actions and objects by squishing the viewpoint information, which is unnecessary for a robot's own action generation after the observation.

9.3.3 One-Shot Imitation Learning and Self-Supervised Learning

One of the new directions in imitation learning is one-shot imitation learning (Finn et al. 2017; Yu et al. 2018; Duan et al. 2017). One-shot imitation learning means that robots are required to learn a new task from only a single demonstration of the robot's or human's performance for the given task. As an example, Finn et al. (2017) combined a metalearning algorithm called model-agnostic meta-learning (MAML; Finn, Abbeel, and Levine 2017) and imitation learning. The MAML enables neural networks to learn a new task from only a few training data. More specifically, the MAML assumes various tasks, and it samples some tasks from which it also samples training and validation data items (at least one item for each). During a meta-learning phase, first the training loss for each task is computed by using initial model parameters and the sampled training data item. By using the computed training loss for each task, the initial model parameters are (tentatively) adapted for each task by gradient descent. Then the validation loss for each task is computed by using the corresponding adapted parameters and the sampled validation data item. Finally, the initial model parameters are optimized to minimize the sum of the validation losses by gradient descent. This means the metalearning algorithm tries to discover generalized initial parameters that can be easily adapted for any task. During a subsequent meta-testing phase, only a single training data item from a new task kept separate from tasks for the meta-learning phase is given, and the generalized initial parameters can be quickly adapted to the task.

In their experiment using a robot PR2, they first collected demonstrations of robot performance for various tasks of object placing via teleoperation. The collected demonstrations consisted of raw visual images from a camera mounted on the robot and action information. The meta-learning was conducted by using these demonstrations to learn how to infer a policy for a new task from only a single demonstration of robot performance. Then, in the meta-testing phase, the robot learned a new task from a single demonstration provided via teleoperation by a human. This is effective for learning a new task quickly; however, the problem is that the framework needs a demonstration of robot performance, and providing a single demonstration of human performance is more straightforward. To tackle this issue, Yu et al. (2018) extended the framework by introducing domain-adaptive meta-learning (DAML). This enables robots to learn how to infer a policy for a new task from only a single demonstration of human performance. They evaluated this extended framework with both the PR2 and Sawyer robots. As expected, these robots could learn a new task from a single demonstration of human performance and could also learn a new task even when the demonstration was performed in different viewpoints and background environmental situations.

Another new direction is self-supervised learning (Nair et al. 2017; Pathak et al. 2018; Lynch et al. 2019). In all the experiments explained above, the demonstrations by human experts were provided for performing specific tasks. As an alternative approach, Lynch et al. (2019) proposed a new paradigm of learning from play (LfP), in which robots acquire

various skills for object manipulation only from play data given by teleoperators and realize goal-directed tasks after a learning phase. In their experiment, human operators first teleoperated a robot in a simulation environment. In the environment, multiple objects for manipulation sat on a desk equipped with a drawer and a shelf with buttons that turned on lights. The operators were asked to freely explore the environment by operating the robot, and visuomotor experiences during this free exploration were collected. The important point is that the curiosity and intrinsic motivation of the operators enabled the acquisition of various types of complex and interactive actions with both manipulative and nonmanipulative objects available in the environment. The collected visuomotor experiences were learned by the play-supervised latent motor plans (Play-LMP) framework that consists of a plan proposal encoder, a plan recognition encoder, and an action decoder. During a learning phase, the first part of the visuomotor experiences was randomly sampled as a sequence. Then only the initial and final states of the sampled sequence were encoded by the plan proposal encoder, and the entire sequence was encoded by the plan recognition encoder. Both encoders generated a latent plan representation and that from the recognition encoder was provided for the action decoder. The encoders and decoder were jointly optimized to maximize action likelihood on the decoder and minimize the KL divergence between the distributions of the latent plan representations from the encoders. After the learning process, providing the current and goal states to the plan proposal encoder and sending the generated latent plan representation from this encoder to the action decoder can generate an action sequence that interpolates the current and goal states. The experimental results showed that the robots that learned from play data were more robust to perturbations in comparison to robots that learned from demonstrations for specific tasks. They also exhibited retrying-until-success behavior thanks to the diversity of the play data.

9.4 Reinforcement-Learning Robot Applications

In the previous part of this chapter, we reviewed neural network–based methods to control robots using predefined data sets of a robot's behavior. In contrast to this "off-line" method, online learning techniques collect samples of the training data set while optimizing models. We now take a look at online learning methods with the deep-learning method called "deep reinforcement learning." This approach provides a way to explore solutions that enable a robot to learn visuomotor tasks instead of a carefully designed training data set. However, it is known that reinforcement-learning methods tend to require large amounts of episode sampling because of noises of rewards or the stochastic property of interaction. In the case of robot tasks, performing many episodes with real robots is costly (e.g., time, computational costs, robot hardware reliability). In this section, we first give an overview of the reinforcement-learning problem setting. Next, we review research on robot tasks using deep reinforcement learning from the viewpoint of how to reduce the cost of episode sampling.

9.4.1 Reinforcement-Learning Problem Setting

The reinforcement-learning (RL) problem setting assumes the interaction between a controllable agent (e.g., a robot controller) and an environment (Sutton and Barto 2018; figure 9.4). For example, a controller of a picking robot can be regarded as an agent, and the environment

corresponds to the space surrounding the robot with some target objects. The agent interacts with the environment by performing an action a. Then the environment's states are altered by the action, and this returns new states and a reward signal r. The reward signal represents how well the current state transition is going, such as the achievement of the task—for example, it may be +1 when the robot successfully picks an object, 0 when the robot moves its arm toward the object, and −1 for failures. The interaction between the agent and the environment will produce a sequential tuple of state, action, and new state with reward (s, a, r, s'). Usually, the RL problem assumes this tuple is sampled from a finite Markov decision process (MDP). To infer an action from the current state is represented as a function called policy $\pi(a|s)$, and the state transition dynamics is formulated as a stochastic probability function $p(s'|s)$. The goal of RL is to find a policy that can maximize the expected sum of reward (called return) in each state of interactions. The expected return is often called "value" $v(s) = \mathbb{E}(\sum r|s)$.

Finding the best policy or explicitly computing the accurate value is intractable due to the stochastic property of MDP; thus, we need to approximate value function. RL approaches can be categorized into several types of this approximation method. One of them is to approximate value conditioned by actions, called "action value." If we can compute an accurate action value, the agent will be able to obtain the best return by selecting an action whose action value is the highest at each time step. The action value is also difficult to compute as well as the state value, so it should be approximated by Monte Carlo methods on episode data sampled by the interactions between agent and environment. The RL Q-learning method adopts a bootstrapping method of the action value by predicting the sum of discounted future rewards. The approximation ability of the action value estimator is the key to the performance of Q-learning. Using deep-learning models as action value approximators has led to significant improvement in RL agents' abilities in video game environments, whose states are usually large-dimensional image data (Mnih et al. 2015; Vinyals et al. 2019). The other RL approach is to optimize a parameterized policy function directly. In the context of deep RL, the policy function is implemented using deep-learning models and optimized via gradient ascent toward the higher state value, called the policy gradient method. This optimization method allows actions to be in continuous space, whereas Q-learning usually allows only discrete action space. Policy gradient methods

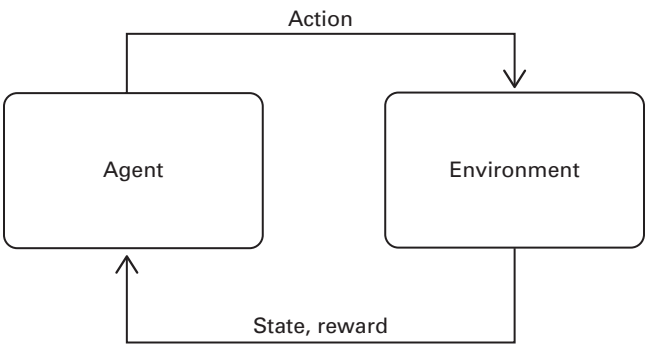

Figure 9.4
Interaction between agent and environment.

have several variants with respect to the type of policy functions and optimization techniques used to stabilize value estimation. Another way to categorize the RL approach is to distinguish whether a learning method is explicitly modeling state transition probability $p(s'|s)$. Methods that model state transition probability are called "model based," whereas "model-free" do not model it. The model-based approach promises lower sample complexity compared to model-free methods because we could substitute predicted future states for states given by running real interactions. When the action space is discrete and the state transition can be accurately simulated on a long time-step horizon, the heuristics of action searches, such as the Monte Carlo tree search, can be used for collecting good sample data for value estimation (Silver et al. 2018). In cases of robotic experimental settings, the state is required to have a large amount of sensory data, including camera images or poses of the robot, so other value estimation or policy optimization methods are required.

By harnessing the power of the deep-learning model's function approximator ability, RL methods have recently been applied to large-dimensional state data and complex tasks, such as games (Mnih et al. 2015; Vinyals et al. 2019) and generative tasks (Ganin et al. 2018; Huang, Heng, and Zhou 2019), including in robotics. However, RL still requires us to collect a good deal of sample data by having the agent explore the environment, in contrast to imitating expert behavior by supervised learning. Running a lot of real robot interactions requires a huge cost in terms of the experiment and the risks of damaging the robots as they explore. Therefore, deep RL researchers have tried to make optimization methods more efficient and stable. One major research direction is to make data collection efficient, and the other is to leverage sample complexity using model-based approaches.

9.4.2 Making Data Collection Efficient

One of the ways to reduce data collection using real robots is to utilize physics simulation software. Although a simulator drastically reduces the cost of experiments, there are huge *reality gaps* due to the limited abilities of simulated environments and robots to reproduce physical world dynamics. One of the approaches to overcome the problem of the reality gap is to augment collected sample data by adding noise to simulation processes, also known as "domain randomization" (Tobin et al. 2017). For example, experiments by Andrychowicz et al. (2020) randomized the property of the robot, the physical parameters such as mass or gravity, and the visual appearance. Domain randomization is expected to improve generalization ability with regard to noise in real environment states or state transition dynamics. Instead of randomizing the state given by a simulator's renderer, replacing state images with more realistic images faked by a generative model has also been investigated. Bousmalis et al. (2018) reported that they drastically reduce the amount of episode sampling in the real robot environment by enhancing the quality of the simulated state image using a generative adversarial network. Adding constraints to force an RL agent trained in simulated environments to behave like an agent in the real environment has also been attempted. Fang et al. (2018) incorporated the adversarial loss of classifying the source of episode data in order to transfer knowledge from an agent in simulation to one in the real environment.

RL experiments on simulators often require multiple software environments running in parallel for sampling efficiency. Conducting real robot exploration tasks in parallel could also reduce data collection time. Levine et al. (2018) built multirobot arm-picking environments

and trained an action value estimator for large collected data samples of images of cluttered objects. The action that controlled the robot arm was obtained by an evolutionary strategy whose candidates were evaluated by the action values estimated as success rates by a deep neural network.

Providing expert episode sequences helps exploration. It is also expected to reduce data collection cost. Peng and colleagues (Peng, Abbeel, et al. 2018; Peng, Kanazawa, Malik, et al. 2018) showed that human motion capture data assisted with a robot control agent's exploration in a simulator. They added a reward that encouraged simulated robots to take poses similar to a human's target poses in the original task, such as walking or performing acrobat motions. Also, the initial state at exploration was sampled from target poses to observe states that are difficult to achieve by taking random actions from the same initial state.

Incorporating reward for imitating expert sequences is related to inverse reinforcement learning, which is an RL approach for estimating reward function from expert data (Ng and Russell 2000). Finn et al. (2016) and Peng, Kanazawa, Toyer, et al. (2018) proposed the use of a generative adversarial protocol to determine the similarities between episodes by the RL agent and the expert data. In this case, the reward was given by a discriminator network trained to distinguish between the sequences from the agent's exploration and the expert. A training reward function approximator network was also expected to relieve the sparseness of the reward. Basic RL requires us to design reward functions for representing task achievements carefully. Very sparse reward distribution, such as a nonzero signal only at the end of an episode, makes exploration challenging since value estimation becomes unstable. A reward estimator by a trained machine-learning model is expected to give nonzero rewards even during episodes. Ganin et al. (2018) proposed a painting RL agent that can be trained by reward signals given by a discriminator network able to distinguish whether a picture image is drawn by the agent or by a human.

9.4.3 Reducing Data Collection by Modeling Environment Dynamics

Model-based RL methods allow policy optimization to acquire sequential data predicted from environment models, and thus they promise to reduce sample complexity in contrast to model-free algorithms. The recent success of generative deep-learning models has led to their utilization in modeling high-dimensional and complex state transitions—for example, image frame sequences. Ebert et al. (2018) proposed image sequence modeling conditioned by a robot's actions for object manipulation tasks. They collected image sequences by moving the robot's arm with random actions and training a deep convolutional network to predict future image frames. After training an image frame predictor, actions were directly optimized by a cross-entropy method, which is a derivative-free optimization method. They produced multiple predicted image sequences from their existing image obtained by a robot with action candidates. Each action candidate was then evaluated with the predicted image at the end of the time-step horizon for differences between the given goal image and the predicted image, or pixel annotation by an experimenter. A combination of future image predictions and a derivative-free algorithm were also proposed by Ha and Schmidhuber (2018). In this study, a state transition function was modeled by a stochastic neural model based on a mixture density network. They argued that the states predicted by deterministic dynamics make the policy optimization adver-

sarial. Nevertheless, nondeterministic modeling will easily lead to inaccurate state prediction due to the uncertainty of the future. Hafner et al. (2018) proposed a combination of both RL modeling methods using a recurrent state-space model (Karl et al. 2019). Instead of directly optimizing the action sequence, model-free RL methods can be used jointly with model-based RL methods. An issue when combining model-based RL with model-free optimization methods is inaccurate dynamics modeling. Kurutach et al. (2018) indicated that policy optimization tends to exploit the region of state space insufficient for achieving good performance. Buckman et al. (2018) proposed the use of an ensemble of several versions of the learned dynamics to stabilize value estimation.

9.5 Conclusion

This chapter introduced several research examples of robot applications using machine learning, especially deep learning, for tasks such as robot vision, the learning of tactile sense and motion, imitation learning, prediction learning, reinforcement learning, and language learning.

It is important to realize that robotics research showing the robot's performance only in simulation and/or in specific environments cannot lead to practical applications. One of the most critical conditions to consider is the evaluation of the robustness of the various noisy situations in the real environment.

In Japan, various manufacturers of industrial robots have already developed multiple prototypes of robot applications of imitation learning and prediction learning. The modularization of robotic systems at the hardware and software levels is progressing quickly, and big developments are expected to be realized with deep-learning technology. In general, the robotics approaches using AI deep-learning methods have the potential to significantly advance cognitive capabilities in robots.

Additional Reading and Resources

- A comprehensive book on deep-learning methods: Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. Cambridge, MA: MIT Press (free online copy: https://www.deeplearningbook.org).
- Position paper discussing the challenges and opportunities connecting robotics with deep learning: Sünderhauf, Niko, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, et al. 2018. "The Limits and Potentials of Deep Learning for Robotics." *International Journal of Robotics Research* 37 (4–5): 405–420.
- Recent volume with extensive coverage of reinforcement-learning methods: Sutton, Richard S., and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.
- OpenAI Gym tool kit for developing reinforcement-learning simulation, including with simulated robots: https://gym.openai.com.

References

Aldoma, Aitor, Zoltan Csaba Marton, Federico Tombari, Walter Wohlkinger, Christian Potthast, Bernhard Zeisl, Radu Rusu, Suat Gedikli, and Markus Vincze. 2012. "Tutorial: Point Cloud Library: Three-Dimensional Object Recognition and 6 DOF Pose Estimation." *IEEE Robotics and Automation Magazine* 19 (3): 80–91. https://doi.org/10.1109/mra.2012.2206675.

Andrychowicz, Open AI: Marcin, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, et al. 2020. "Learning Dexterous In-Hand Manipulation." *International Journal of Robotics Research* 39 (1): 3–20. https://doi.org/10.1177/0278364919887447.

Anzai, Tomoki, and Kuniyuki Takahashi. 2020. "Deep Gated Multi-modal Learning: In-Hand Object Pose Changes Estimation Using Tactile and Image Data." In *IEEE International Conference on Intelligent Robots and Systems*. New York: IEEE.

Argall, Brenna D., Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. "A Survey of Robot Learning from Demonstration." *Robotics and Autonomous Systems* 57 (5): 469–483. https://doi.org/10.1016/j.robot.2008.10.024.

Arie, Hiroaki, Takafumi Arakaki, Shigeki Sugano, and Jun Tani. 2012. "Imitating Others by Composition of Primitive Actions: A Neuro-Dynamic Model." *Robotics and Autonomous Systems* 60 (5): 729–741. https://doi.org/10.1016/j.robot.2011.11.005.

Bach, Sebastian, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus Robert Müller, and Wojciech Samek. 2015. "On Pixel-Wise Explanations for Non-linear Classifier Decisions by Layer-Wise Relevance Propagation." Edited by Oscar Deniz Suarez. *PLoS One* 10 (7): e0130140. https://doi.org/10.1371/journal.pone.0130140.

Baishya, Shiv S., and Berthold Bäuml. 2016. "Robust Material Classification with a Tactile Skin Using Deep Learning." In *IEEE International Conference on Intelligent Robots and Systems*, 8–15. New York: IEEE. https://doi.org/10.1109/iros.2016.7758088.

Billard, Aude, Sylvain Calinon, Rüdiger Dillmann, and Stefan Schaal. 2008. "Robot Programming by Demonstration." In *Springer Handbook of Robotics*, edited by Bruno Siciliano and Oussama Khatib, 1371–1394. Berlin: Springer. https://doi.org/10.1007/978-3-540-30301-5_60.

Bimbo, Joao, Shan Luo, Kaspar Althoefer, and Hongbin Liu. 2016. "In-Hand Object Pose Estimation Using Covariance-Based Tactile to Geometry Matching." *IEEE Robotics and Automation Letters* 1 (1): 570–577. https://doi.org/10.1109/lra.2016.2517244.

Bousmalis, Konstantinos, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, et al. 2018. "Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping." In *Proceedings—IEEE International Conference on Robotics and Automation*, 4243–4250. New York: IEEE. https://doi.org/10.1109/icra.2018.8460875.

Buckman, Jacob, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. 2018. "Sample-Efficient Reinforcement Learning with Stochastic Ensemble Value Expansion." In *Advances in Neural Information Processing Systems*, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, 8224–8234. Red Hook, NY: Curran. http://papers.nips.cc/paper/8044-sample-efficient-reinforcement-learning-with-stochastic-ensemble-value-expansion.pdf.

Calandra, Roberto, Andrew Owens, Dinesh Jayaraman, Justin Lin, Wenzhen Yuan, Jitendra Malik, Edward H. Adelson, and Sergey Levine. 2018. "More than a Feeling: Learning to Grasp and Regrasp Using Vision and Touch." *IEEE Robotics and Automation Letters* 3 (4): 3300–3307. https://doi.org/10.1109/lra.2018.2852779.

Calinon, Sylvain, Florent Guenter, and Aude Billard. 2007. "On Learning, Representing, and Generalizing a Task in a Humanoid Robot." *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 37 (2): 286–298. https://doi.org/10.1109/tsmcb.2006.886952.

Chen, Xi, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets." In *Advances in Neural Information Processing Systems*, edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, 2180–2188. Red Hook, NY: Curran. http://papers.nips.cc/paper/6399-infogan-interpretable-representation-learning-by-information-maximizing-generative-adversarial-nets.pdf.

Chen, Yiwen, Shingo Murata, Hiroaki Arie, Tetsuya Ogata, Jun Tani, and Shigeki Sugano. 2016. "Emergence of Interactive Behaviors between Two Robots by Prediction Error Minimization Mechanism." In *2016 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics*, 302–307. New York: IEEE. https://doi.org/10.1109/devlrn.2016.7846838.

Choi, Changhyun, and Henrik I. Christensen. 2012. "3D Pose Estimation of Daily Objects Using an RGB-D Camera." In *IEEE International Conference on Intelligent Robots and Systems*, 3342–3349. New York: IEEE. https://doi.org/10.1109/iros.2012.6386067.

Choi, Changhyun, Yuichi Taguchi, Oncel Tuzel, Ming Yu Liu, and Srikumar Ramalingam. 2012. "Voting-Based Pose Estimation for Robotic Assembly Using a 3D Sensor." In *Proceedings—IEEE International Conference on Robotics and Automation*, 1724–1731. New York: IEEE. https://doi.org/10.1109/icra.2012.6225371.

Dahiya, Ravinder S., Philipp Mittendorfer, Maurizio Valle, Gordon Cheng, and Vladimir J. Lumelsky. 2013. "Directions toward Effective Utilization of Tactile Skin: A Review." *IEEE Sensors Journal* 13 (11): 4121–4138. https://doi.org/10.1109/jsen.2013.2279056.

Dong, Siyuan, Wenzhen Yuan, and Edward H. Adelson. 2017. "Improved GelSight Tactile Sensor for Measuring Geometry and Slip." In *IEEE International Conference on Intelligent Robots and Systems*, 137–144. New York: IEEE. https://doi.org/10.1109/iros.2017.8202149.

Duan, Yan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. 2017. "One-Shot Imitation Learning." In *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 1087–1098. Red Hook, NY: Curran. http://papers.nips.cc/paper/6709-one-shot-imitation-learning.pdf.

Ebert, Frederik, Chelsea Finn, Sudeep Dasari, Annie Xie, Alex Lee, and Sergey Levine. 2018. "Visual Foresight: Model-Based Deep Reinforcement Learning for Vision-Based Robotic Control." ArXiv preprint: http://arxiv.org/abs/1812.00568.

Falco, Pietro, Abdallah Attawia, Matteo Saveriano, and Dongheui Lee. 2018. "On Policy Learning Robust to Irreversible Events: An Application to Robotic In-Hand Manipulation." *IEEE Robotics and Automation Letters* 3 (3): 1482–1489. https://doi.org/10.1109/lra.2018.2800110.

Fang, Kuan, Yunfei Bai, Stefan Hinterstoisser, Silvio Savarese, and Mrinal Kalakrishnan. 2018. "Multi-task Domain Adaptation for Deep Learning of Instance Grasping from Simulation." In *Proceedings—IEEE International Conference on Robotics and Automation*, 3516–3523. New York: IEEE. https://doi.org/10.1109/icra.2018.8461041.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine. 2017. "Model-Agnostic Meta-learning for Fast Adaptation of Deep Networks." In *34th International Conference on Machine Learning, ICML 2017* 3:1856–1868. JMLR.org.

Finn, Chelsea, Paul Christiano, Pieter Abbeel, and Sergey Levine. 2016. "A Connection between Generative Adversarial Networks, Inverse Reinforcement Learning, and Energy-Based Models." ArXiv preprint: http://arxiv.org/abs/1611.03852.

Finn, Chelsea, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. 2017. "One-Shot Visual Imitation Learning via Meta-learning." *Proceedings of the 1st Conference on Robot Learning (CoRL 2017)*, 1–12. ArXiv preprint: http://arxiv.org/abs/1709.04905.

Fishel, Jeremy A., and Gerald E. Loeb. 2012. "Sensing Tactile Microvibrations with the BioTac Comparison with Human Sensitivity." In *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, 1122–1127. New York: IEEE. https://doi.org/10.1109/biorob.2012.6290741.

Friston, Karl J., Jean Daunizeau, James Kilner, and Stefan J. Kiebel. 2010. "Action and Behavior: A Free-Energy Formulation." *Biological Cybernetics* 102 (3): 227–260. https://doi.org/10.1007/s00422-010-0364-z.

Funabashi, Satoshi, Alexander Schmitz, Takashi Sato, Sophon Somlor, and Shigeki Sugano. 2018. "Versatile In-Hand Manipulation of Objects with Different Sizes and Shapes Using Neural Networks." In *IEEE-RAS International Conference on Humanoid Robots*, 768–775. New York: IEEE. https://doi.org/10.1109/humanoids.2018.8624961.

Ganin, Yaroslav, Tejas Kulkarni, Igor Babuschkin, S. M. Ali Eslami, and Oriol Vinyals. 2018. "Synthesizing Programs for Images Using Reinforced Adversarial Learning." ArXiv preprint: http://arxiv.org/abs/1804.01118.

Gao, Yang, Lisa Anne Hendricks, Katherine J. Kuchenbecker, and Trevor Darrell. 2016. "Deep Learning for Tactile Understanding from Visual and Haptic Data." In *Proceedings—IEEE International Conference on Robotics and Automation*, 536–543. New York: IEEE. https://doi.org/10.1109/icra.2016.7487176.

Ha, David, and Jürgen Schmidhuber. 2018. "Recurrent World Models Facilitate Policy Evolution." *Advances in Neural Information Processing Systems* C:2450–2462. ArXiv preprint: http://arxiv.org/abs/1809.01999.

Hafner, Danijar, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. 2018. "Learning Latent Dynamics for Planning from Pixels." ArXiv preprint: http://arxiv.org/abs/1811.04551.

Han, L., Y. S. Guan, Z. X. Li, Q. Shi, and J. C. Trinkle. 1997. "Dextrous Manipulation with Rolling Contacts." In *Proceedings of International Conference on Robotics and Automation* 2:992–997. New York: IEEE. https://doi.org/10.1109/robot.1997.614264.

Han, L., and J. C. Trinkle. 1998. "Dextrous Manipulation by Rolling and Finger Gaiting." In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation* 1:730–735. Cat. No.98CH36146. New York: IEEE. https://doi.org/10.1109/robot.1998.677060.

Harnad, Stevan. 1990. "The Symbol Grounding Problem." *Physica D: Nonlinear Phenomena* 42 (1–3): 335–346. https://doi.org/10.1016/0167-2789(90)90087-6.

Heinrich, Stefan, and Stefan Wermter. 2014. "Interactive Language Understanding with Multiple Timescale Recurrent Neural Networks." In *Lecture Notes in Computer Science*, edited by Stefan Wermter, Cornelius Weber, Włodzisław Duch, Timo Honkela, Petia Koprinkova-Hristova, Sven Magg, Günther Palm, and Alessandro E. P. Villa, 8681 LNCS:193–200. Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-319-11179-7_25.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Hodaň, Tomáš, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, et al. 2018. "BOP: Benchmark for 6D Object Pose Estimation." In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11214 LNCS:19–35. https://doi.org/10.1007/978-3-030-01249-6_2.

Hogan, Francois R., Maria Bauza, Oleguer Canal, Elliott Donlon, and Alberto Rodriguez. 2018. "Tactile Regrasp: Grasp Adjustments via Simulated Tactile Transformations." In *IEEE International Conference on Intelligent Robots and Systems*, 2963–2970. New York: IEEE. https://doi.org/10.1109/iros.2018.8593528.

Hu, Yinlin, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. 2019. "Segmentation-Driven 6D Object Pose Estimation." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 3380–3389. New York: IEEE. https://doi.org/10.1109/cvpr.2019.00350.

Huang, Zhewei, Wen Heng, and Shuchang Zhou. 2019. "Learning to Paint with Model-Based Deep Reinforcement Learning," ArXiv preprint: http://arxiv.org/abs/1903.04411.

Ijspeert, Auke Jan, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. 2013. "Dynamical Movement Primitives: Learning Attractor Models Formotor Behaviors." *Neural Computation* 25 (2): 328–373. https://doi.org/10.1162/neco_a_00393.

Ijspeert, Auke Jan, Jun Nakanishi, and Stefan Schaal. 2002. "Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots." In *Proceedings—IEEE International Conference on Robotics and Automation* 2:1398–1403. New York: IEEE. https://doi.org/10.1109/robot.2002.1014739.

Inamura, Tetsunari, Iwaki Toshima, Hiroaki Tanie, and Yoshihiko Nakamura. 2004. "Embodied Symbol Emergence Based on Mimesis Theory." *International Journal of Robotics Research* 23 (4–5): 363–377. https://doi.org/10.1177/0278364904042199.

Ito, Masato, Kuniaki Noda, Yukiko Hoshino, and Jun Tani. 2006. "Dynamic and Interactive Generation of Object Handling Behaviors by a Small Humanoid Robot Using a Dynamic Neural Network Model." *Neural Networks* 19 (3): 323–337. https://doi.org/10.1016/j.neunet.2006.02.007.

Iwata, Hiroyasu, and Shigeki Sugano. 2009. "Design of Human Symbiotic Robot TWENDY-ONE." In *Proceedings—IEEE International Conference on Robotics and Automation*, 580–586. New York: IEEE. https://doi.org/10.1109/robot.2009.5152702.

Johnson, Micah K., and Edward H. Adelson. 2009. "Retrographic Sensing for the Measurement of Surface Texture and Shape." In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 1070–1077. New York: IEEE. https://doi.org/10.1109/cvpr.2009.5206534.

Jordan, Michael I. 1997. "Serial Order: A Parallel Distributed Processing Approach." In *Advances in Psychology*, edited by John W. Donahoe and Vivian Packard Dorsel, 121:471–495. Advances in Psychology. Amsterdam: North-Holland. https://doi.org/10.1016/s0166-4115(97)80111-2.

Karl, Maximilian, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. 2019. "Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data." *5th International Conference on Learning Representations, ICLR 2017—Conference Track Proceedings*. ArXiv preprint: https://arxiv.org/abs/1605.06432.

Kase, Kei, Ryoichi Nakajo, Hiroki Mori, and Tetsuya Ogata. 2019. "Learning Multiple Sensorimotor Units to Complete Compound Tasks Using an RNN with Multiple Attractors." In *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4244–4249. https://doi.org/10.1109/iros40897.2019.8967780.

Kase, Kei, Kanata Suzuki, Pin Chu Yang, Hiroki Mori, and Tetsuya Ogata. 2018. "Put-in-Box Task Generated from Multiple Discrete Tasks by a Humanoid Robot Using Deep Learning." In *Proceedings—IEEE International Conference on Robotics and Automation*, 6447–6452. New York: IEEE. https://doi.org/10.1109/icra.2018.8460623.

Kurutach, Thanard, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. 2018. "Model-Ensemble Trust-Region Policy Optimization." In *6th International Conference on Learning Representations, ICLR 2018—Conference Track Proceedings*. Available at https://iclr.cc/Conferences/2018/Schedule.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." *Nature* 521 (7553): 436–444.

Lenz, Ian, Honglak Lee, and Ashutosh Saxena. 2015. "Deep Learning for Detecting Robotic Grasps." *International Journal of Robotics Research* 34 (4–5): 705–724. https://doi.org/10.1177/0278364914549607.

Levine, Sergey, Peter Pastor, Alex Krizhevsky, Julian Ibarz, and Deirdre Quillen. 2018. "Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection." *International Journal of Robotics Research* 37 (4–5): 421–436. https://doi.org/10.1177/0278364917710318.

Li, Miao, Hang Yin, Kenji Tahara, and Aude Billard. 2014. "Learning Object-Level Impedance Control for Robust Grasping and Dexterous Manipulation." In *Proceedings—IEEE International Conference on Robotics and Automation*, 6784–6791. New York: IEEE. https://doi.org/10.1109/icra.2014.6907861.

Lynch, Corey, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. 2019. "Learning Latent Plans from Play." *Proceedings of the 3rd Conference on Robot Learning (CoRL 2019)*. ArXiv preprint: http://arxiv.org/abs/1903.01973.

Meltzoff, Andrew N., and M. Keith Moore. 1977. "Imitation of Facial and Manual Gestures by Human Neonates." *Science* 198 (4312): 75–78. https://doi.org/10.1126/science.198.4312.75.

Mittendorfer, Philipp, and Gordon Cheng. 2011. "Humanoid Multimodal Tactile-Sensing Modules." *IEEE Transactions on Robotics* 27 (3): 401–410. https://doi.org/10.1109/tro.2011.2106330.

Mnih, Volodymyr, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, et al. 2015. "Human-Level Control through Deep Reinforcement Learning." *Nature* 518 (7540): 529–533. https://doi.org/10.1038/nature14236.

Nagai, Yukie. 2019. "Predictive Learning: Its Key Role in Early Cognitive Development." *Philosophical Transactions of the Royal Society B: Biological Sciences* 374 (1771): 20180030. https://doi.org/10.1098/rstb.2018.0030.

Nair, Ashvin, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. 2017. "Combining Self-Supervised Learning and Imitation for Vision-Based Rope Manipulation." In *Proceedings—IEEE International Conference on Robotics and Automation*, 2146–2153. New York: IEEE. https://doi.org/10.1109/icra.2017.7989247.

Nakajo, Ryoichi, Shingo Murata, Hiroaki Arie, and Tetsuya Ogata. 2015. "Acquisition of Viewpoint Representation in Imitative Learning from Own Sensory-Motor Experiences." In *5th Joint International Conference on Development and Learning and Epigenetic Robotics, ICDL-EpiRob 2015*, 326–331. New York: IEEE. https://doi.org/10.1109/devlrn.2015.7346166.

Nakajo, Ryoichi, Shingo Murata, Hiroaki Arie, and Tetsuya Ogata. 2018. "Acquisition of Viewpoint Transformation and Action Mappings via Sequence to Sequence Imitative Learning by Deep Neural Networks." *Frontiers in Neurorobotics* 12:46. https://doi.org/10.3389/fnbot.2018.00046.

Namikawa, Jun, Ryunosuke Nishimoto, and Jun Tani. 2011. "A Neurodynamic Account of Spontaneous Behavior." *PLoS Computational Biology* 7 (10): e1002221–e1002221. https://doi.org/10.1371/journal.pcbi.1002221.

Ng, Andrew, and Stuart Russell. 2000. "Algorithms for Inverse Reinforcement Learning." In *Proceedings of the Seventeenth International Conference on Machine Learning* 0:663–670. San Francisco: Morgan Kaufmann. https://doi.org/10.2460/ajvr.67.2.323.

Nishihara, Joe, Tomoaki Nakamura, and Takayuki Nagai. 2017. "Online Algorithm for Robots to Learn Object Concepts and Language Model." *IEEE Transactions on Cognitive and Developmental Systems* 9 (3): 255–268. https://doi.org/10.1109/tcds.2016.2552579.

Nishimoto, Ryunosuke, and Jun Tani. 2009. "Development of Hierarchical Structures for Actions and Motor Imagery: A Constructivist View from Synthetic Neuro-Robotics Study." *Psychological Research* 73 (4): 545–558. https://doi.org/10.1007/s00426-009-0236-0.

Noda, Kuniaki, Hiroaki Arie, Yuki Suga, and Tetsuya Ogata. 2014. "Multimodal Integration Learning of Robot Behavior Using Deep Neural Networks." *Robotics and Autonomous Systems* 62 (6): 721–736. https://doi.org/10.1016/j.robot.2014.03.003.

Ogata, Tetsuya, Masamitsu Murase, Jim Tani, Kazunori Komatani, and Hiroshi G. Okuno. 2007. "Two-Way Translation of Compound Sentences and Arm Motions by Recurrent Neural Networks." In *IEEE International Conference on Intelligent Robots and Systems*, 1858–1863. New York: IEEE. https://doi.org/10.1109/iros.2007.4399265.

Ohmura, Yoshiyuki, Yasuo Kuniyoshi, and Akihiko Nagakubo. 2006. "Conformable and Scalable Tactile Sensor Skin for a Curved Surfaces." In *Proceedings—IEEE International Conference on Robotics and Automation*, 1348–1353. New York: IEEE. https://doi.org/10.1109/robot.2006.1641896.

Pathak, Deepak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A. Efros, and Trevor Darrell. 2018. "Zero-Shot Visual Imitation." In *6th International Conference on Learning Representations, ICLR 2018—Conference Track Proceedings*, 2050–2053. Available at https://iclr.cc/Conferences/2018/Schedule.

Paulino, Tiago, Pedro Ribeiro, Miguel Neto, Susana Cardoso, Alexander Schmitz, Jose Santos-Victor, Alexandre Bernardino, and Lorenzo Jamone. 2017. "Low-Cost 3-Axis Soft Tactile Sensors for the Human-Friendly Robot

Vizzy." In *Proceedings—IEEE International Conference on Robotics and Automation*, 966–971. New York: IEEE. https://doi.org/10.1109/icra.2017.7989118.

Peng, Xue Bin, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. 2018. "DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills." *ACM Transactions on Graphics* 37 (4): 1–14. https://doi.org/10.1145/3197517.3201311.

Peng, Xue Bin, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. 2018. "SFV: Reinforcement Learning of Physical Skills from Videos" 37 (6). https://doi.org/10.1145/3272127.3275014.

Peng, Xue Bin, Angjoo Kanazawa, Sam Toyer, Pieter Abbeel, and Sergey Levine. 2018. "Variational Discriminator Bottleneck: Improving Imitation Learning, Inverse RL, and GANs by Constraining Information Flow." ArXiv preprint: http://arxiv.org/abs/1810.00821.

Redmon, Joseph, and Anelia Angelova. 2015. "Real-Time Grasp Detection Using Convolutional Neural Networks." In *Proceedings—IEEE International Conference on Robotics and Automation*, 1316–1322. New York: IEEE. https://doi.org/10.1109/icra.2015.7139361.

Rusinkiewicz, Szymon, and Marc Levoy. 2001. "Efficient Variants of the ICP Algorithm." In *Proceedings of International Conference on 3-D Digital Imaging and Modeling, 3DIM*, 145–152. New York: IEEE. https://doi.org/10.1109/im.2001.924423.

Schmitz, Alexander, Yusuke Bansho, Kuniaki Noda, Hiroyasu Iwata, Tetsuya Ogata, and Shigeki Sugano. 2014. "Tactile Object Recognition Using Deep Learning and Dropout." In *IEEE-RAS International Conference on Humanoid Robots*, 1044–1050. New York: IEEE. https://doi.org/10.1109/humanoids.2014.7041493.

Schwarz, Max, Hannes Schulz, and Sven Behnke. 2015. "RGB-D Object Recognition and Pose Estimation Based on Pre-trained Convolutional Neural Network Features." In *Proceedings—IEEE International Conference on Robotics and Automation*, 1329–1335. New York: IEEE. https://doi.org/10.1109/icra.2015.7139363.

Silver, David, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, et al. 2018. "A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go through Self-Play." *Science* 362 (6419): 1140–1144. https://doi.org/10.1126/science.aar6404.

Smilkov, Daniel, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. "SmoothGrad: Removing Noise by Adding Noise." ArXiv preprint: 1706.03825. http://arxiv.org/abs/1706.03825.

Stramandinoli, Francesca, Davide Marocco, and Angelo Cangelosi. 2017. "Making Sense of Words: A Robotic Model for Language Abstraction." *Autonomous Robots* 41 (2): 367–383. https://doi.org/10.1007/s10514-016-9587-8.

Sugita, Yuuya, and Jun Tani. 2005. "Learning Semantic Combinatoriality from the Interaction between Linguistic and Behavioral Processes." *Adaptive Behavior* 13 (1): 33–52. https://doi.org/10.1177/105971230501300102.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. "Sequence to Sequence Learning with Neural Networks." In *Advances in Neural Information Processing Systems* 4:3104–3112.

Sutton, Richard S., and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Takahashi, Kuniyuki, and Jethro Tan. 2019. "Deep Visuo-tactile Learning: Estimation of Tactile Properties from Images." In *Proceedings—IEEE International Conference on Robotics and Automation*, 8951–8957. New York: IEEE. https://doi.org/10.1109/icra.2019.8794285.

Tellex, Stefanie, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. 2011. "Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation." In *Proceedings of the National Conference on Artificial Intelligence* 2:1507–1514.

Tian, Stephen, Frederik Ebert, Dinesh Jayaraman, Mayur Mudigonda, Chelsea Finn, Roberto Calandra, and Sergey Levine. 2019. "Manipulation by Feel: Touch-Based Control with Deep Predictive Models." In *Proceedings—IEEE International Conference on Robotics and Automation*, 818–824. New York: IEEE. https://doi.org/10.1109/icra.2019.8794219.

Tobin, Josh, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. 2017. "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World." In *IEEE International Conference on Intelligent Robots and Systems*. New York: IEEE. https://doi.org/10.1109/iros.2017.8202133.

Tomo, Tito Pradhono, Alexander Schmitz, Wai Keat Wong, Harris Kristanto, Sophon Somlor, Jinsun Hwang, Lorenzo Jamone, and Shigeki Sugano. 2018. "Covering a Robot Fingertip with USkin: A Soft Electronic Skin with Distributed 3-Axis Force Sensitive Elements for Robot Hands." *IEEE Robotics and Automation Letters* 3 (1): 124–131. https://doi.org/10.1109/lra.2017.2734965.

Vinyals, Oriol, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, et al. 2019. "Grandmaster Level in StarCraft II Using Multi-agent Reinforcement Learning." *Nature* 575 (7782): 350–354. https://doi.org/10.1038/s41586-019-1724-z.

Wu, Bohan, Iretiayo Akinola, Jacob Varley, and Peter Allen. 2019. "MAT: Multi-fingered Adaptive Tactile Grasping via Deep Reinforcement Learning." *3rd Conference on Robot Learning*. ArXiv preprint: https://arxiv.org/abs/1909.04787.

Xu, Kelvin, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." In *32nd International Conference on Machine Learning 2015* 3:2048–2057.

Yamada, Tatsuro, Hiroyuki Matsunaga, and Tetsuya Ogata. 2018. "Paired Recurrent Autoencoders for Bidirectional Translation between Robot Actions and Linguistic Descriptions." *IEEE Robotics and Automation Letters* 3 (4): 3441–3448. https://doi.org/10.1109/lra.2018.2852838.

Yamaguchi, Akihiko, and Christopher G. Atkeson. 2016. "Combining Finger Vision and Optical Tactile Sensing: Reducing and Handling Errors While Cutting Vegetables." In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 1045–1051. New York: IEEE. https://doi.org/10.1109/humanoids.2016.7803400.

Yamashita, Yuichi, and Jun Tani. 2008. "Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: A Humanoid Robot Experiment." *PLoS Computational Biology* 4 (11). https://doi.org/10.1371/journal.pcbi.1000220.

Yang, Haolin, Fuchun Sun, Wenbing Huang, Lele Cao, and Bin Fang. 2016. "Tactile Sequence Based Object Categorization: A Bag of Features Modeled by Linear Dynamic System with Symmetric Transition Matrix." In *Proceedings of the International Joint Conference on Neural Networks*, 5218–5225. New York: IEEE. https://doi.org/10.1109/ijcnn.2016.7727889.

Yang, Pin Chu, Kazuma Sasaki, Kanata Suzuki, Kei Kase, Shigeki Sugano, and Tetsuya Ogata. 2017. "Repeatable Folding Task by Humanoid Robot Worker Using Deep Learning." *IEEE Robotics and Automation Letters* 2 (2): 397–403. https://doi.org/10.1109/lra.2016.2633383.

Yang, Yezhou, Yi Li, Cornelia Fermüller, and Yiannis Aloimonos. 2015. "Robot Learning Manipulation Action Plans by 'Watching' Unconstrained Videos from the World Wide Web." In *Proceedings of the National Conference on Artificial Intelligence* 5:3686–3692.

Yu, Tianhe, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. 2018. "One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-learning." *In Proceedings of the Robotics: Science and Systems XIV (RSS 2018)*, 1–12. https://doi.org/10.15607/rss.2018.xiv.002.

Yuan, Wenzhen, Siyuan Dong, and Edward H. Adelson. 2017 "GelSight: High-Resolution Robot Tactile Sensors for Estimating Geometry and Force." *Sensors* 17 (12): 2762. https://doi.org/10.3390/s17122762.

Yuan, Wenzhen, Shaoxiong Wang, Siyuan Dong, and Edward Adelson. 2017. "Connecting Look and Feel: Associating the Visual and Tactile Properties of Physical Materials." In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 4494–4502. New York: IEEE. https://doi.org/10.1109/cvpr.2017.478.

Yuan, Wenzhen, Chenzhuo Zhu, Andrew Owens, Mandayam A. Srinivasan, and Edward H. Adelson. 2017. "Shape-Independent Hardness Estimation Using Deep Learning and a GelSight Tactile Sensor." In *Proceedings—IEEE International Conference on Robotics and Automation*, 951–958. New York: IEEE. https://doi.org/10.1109/icra.2017.7989116.

Zhang, Yazhan, Weihao Yuan, Zicheng Kan, and Michael Yu Wang. 2020. "Towards Learning to Detect and Predict Contact Events on Vision-Based Tactile Sensors." *3rd Conference on Robot Learning*, 1395–1404. ArXiv preprint: https://arxiv.org/abs/1910.03973.