Programming By Demonstration using A Learning Based Approach: A Mini Review

Atul Acharya
Department of Mechanical Engineering
University of South Florida
Tampa, USA
atulacharya@usf.edu

Dr. Rajiv Dubey
Department of Mechanical Engineering
University of South Florida
Tampa. USA
dubey@usf.edu

Dr. Redwan Alqasemi
Department of Mechanical Engineering
University of South Florida
Tampa, USA
alqasemi@usf.edu

Abstract — ***Wheelchair-mounted robotic arms are used in rehabilitation robotics to help persons with physical impairment perform activity of daily living (ADL) tasks. However, the dexterity of manipulation tasks makes the teleoperation of the robotic arm challenging for the user, as it is difficult to control all degrees of freedom with a handheld joystick or a touch screen device. PbD (Programming by demonstration) allows the user to demonstrate the desired behavior and enables the system to learn from the demonstrations and adapt to a new environment. This learned model can perform a new set of action in a new environment. Learning from demonstration includes object identification and recognition, trajectory planning, obstacle avoidance, and adapting to a new environment, wherever necessary. PbD using a learning-based approach learns the task through a model that captures the underlying structures of the task. The model can be a probabilistic graphical model, a neural network, or a combination of both. PbD with learning can be generalized and applied to new situations as this method enables the robot to learn the model rather than just memorizing and imitating the demonstration. In addition to this, it also helps in efficient learning with a reduced number of demonstrations. This survey focuses on an overview of the recent machine learning (ML) techniques used with PbD to perform dexterous manipulation tasks that enable the robot to learn and apply what is learned to a new set of tasks and a new environment.***

Keywords—ADL, teleoperation, PbD, object recognition, trajectory planning, obstacle avoidance, manipulation, machine-learning techniques

I. INTRODUCTION

Humans can determine how to approach an object and pick it up when introduced to a new one. They can see the object, identify the location and manipulate their arms to reach an object of interest, avoiding the obstacles in the environment. It has always been an area of interest to create robots that can perform dexterous manipulation and grasping comparable to humans. However, introducing new objects and tasks, the constantly changing environment, and the complexity involved in manipulation, make it challenging to solve the problem [1]. Nearly one million American adults need assistance to perform ADL tasks, i.e., pouring a glass of milk or water, combing hair, or feeding themselves [2].

Wheelchair-mounted robotic arms are equipped with dexterous tools to help users perform the desired action without relying on the human caregiver. They can be equipped with a vision system for object detection and recognition and can be programmed to perform complex manipulation tasks with autonomy. However, this requires high programming skills and constant updates with changing environments, which is evident in the real world [2]. Motion planning can help eliminate specific lower-level actions like trajectories. However, it requires specifying goal location and via points, which could be more reliable in a constantly changing environment [3]. Moreover, the users should easily be able to control wheelchair-mounted assistive robots for effective operation. Most Wheelchair-mounted robotic arms are equipped with control interfaces, such as joysticks, that provide low-dimensional control, owing to the user's physical limitations, although a robotic arm has more degrees of freedom (DoFs).

PbD is a method that allows a robot to learn intuitively from the user demonstration and requires minimal programming skills. This process enables the robot to learn and adapt to a demonstration by humans, along with the constraints in the environment and the model of the task to be performed. The procedure of demonstration groups the human as a teacher and the robot as a learner, which facilitates the robot to learn from the demonstration and autonomously perform novel tasks in a new environment [4]. There are several ways to demonstrate a task to the robot. Most widely, the demonstration approaches can be categorized into three broad categories – kinesthetic teaching, teleoperation, and imitation learning, or passive observation. The mode of demonstration can be chosen based on the complexity of the application. The ease of demonstration and mapping the demonstration to the working space of the robot are the critical factors in choosing the method.

PbD using a learning-based approach includes using ML algorithms to generalize the learned behavior from human demonstration to a new environment. The choice of the learning algorithm depends on the specific task, the type and quality of the available data, and the desired performance metrics. PbD or LfD is often combined with ML techniques to learn

manipulation tasks in real environments [5]. It is essential to develop a model that a robot can easily be taught to achieve complex dexterity and adapt to the manipulation skills during a demonstration. This model is often called skill learning framework and is desired if flexible, allowing the robot to learn and adapt to the human tutor [6]. PbD using learning involves two distinct phases. The first phase is a task learning phase that creates representation from the demonstration, followed by the task refining phase. This iterative process helps converge the learned task representation to a desired task [7].

The paper's organization is as follows: We categorize the PbD approaches in section II, followed by identifying the training methods in section III. This section also includes a detailed survey of different learning methods used in each category. Section IV includes a discussion of the appropriate methods for PbD. Finally, concluding remarks are made in Section V.

II. CATEGORIES OF DEMONSTRATION

As discussed in Section I, the choice of the mode of demonstration depends on the complexity of the manipulation, the ease of demonstration, and mapping the demonstration to the work space. The demonstration approaches are broadly categorized as – kinesthetic teaching, teleoperation, and imitation learning, or passive observation, which is shown in Fig 1. We will discuss each method briefly and compare them against the key factors discussed above.

A. Kinesthetic teaching

This is the mode of demonstration where the user physically guides the robot through the task with the robot's own body. The user moves the robot's joints with their hands to demonstrate the desired motion. The inbuilt sensors record the kinematic parameters like joint angles and torques, which is then fed to the learning model. It relies on robot hardware and requires no additional interfaces or user training. This mode of demonstration holds following the ease of demonstration and mapping the demonstration to the robot's workspace, as it is directly on the robot utilizing its inbuilt sensors to record. This process cannot control all the DoF of the robot by the user [3].

B. Teleoperation

This is a mode of demonstration where the user utilizes an interface to control the robot's DoF. The interfaces can be a joystick [6, 8, 9], a GUI such as a touch screen or a monitor [7, 8], and haptic devices [6, 10, 11, 12] in some cases. This approach opens the door to remote controlling as the user need not be near the robot to demonstrate the action. Also, the user can demonstrate a large-scale multi-robot system using this technique.

However, this approach demands rigorous user training in using the interface to control the DoFs of the robot. Teleoperation can handle higher degrees of freedom, and it is easy to map the demonstration to the robot's workspace, as the robot system is also involved in task execution in lieu with the demonstration provided.

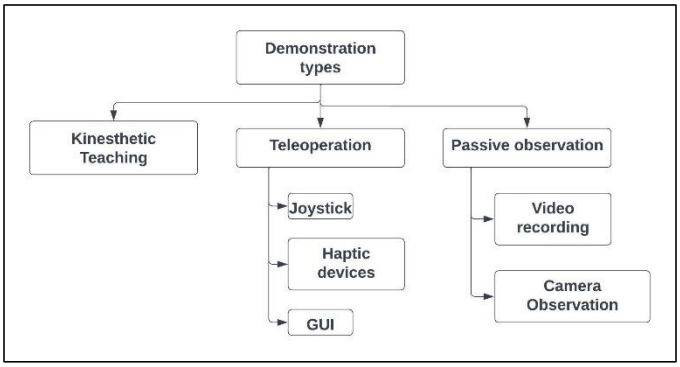

Fig 1. Categories of demonstration

C. Passive Observation or Imitation learning

Passive observation, also known as imitation learning, is an approach where the user demonstrates the action using his/her own body, which the robot observes passively, without execution. Additional sensors sometimes facilitate this demonstration for tracking. This approach is very suitable for robots with Higher DoF, where kinesthetic teaching or teleoperation can be cumbersome. However, mapping the demonstration to the robot's workspace can be complicated. The demonstration techniques can be recording a video of a demonstration [5, 8, 13] or camera observation [14, 15, 16]. The summary of the demonstration categories discussed above has been illustrated in Fig. 1.

III. TRAINING METHODS IN MACHINE LEARNING

From the perspective of learning algorithm, learning from demonstration is categorized into two broad sections: direct approach and indirect approach. The direct approach is referred to as the process of learning the policy from the demonstration provided, which maps the state to the action, commonly known as Behavior cloning. The indirect approach focuses on recovering the reward function from the demonstration and learning the policy that maximizes the recovered reward functions [6]. This paper will classify the training methods as unsupervised, supervised, and reinforcement learning, and will discuss each in detail. This categorization will help us to understand the training methods in a broader way where all the associated techniques can be discussed.

A. Unsupervised Learning

Unsupervised learning (UL), as suggested by its name, is a method of training a ML model that utilizes unlabeled data. Here, the unlabeled data refers to a data set that has not been identified in terms of characteristics, properties, or classifications. Therefore, the model is trained to predict the patterns in the input data and establish a relationship within it [3, 17].

Consider a system with a sequence of input x_1, x_2, x_3, …, x_t where x_t is a sensory input at any time t. Based on the application, this input can be different such as a video or a pixel in an image or something else. UL is defined by the system's

ability to learn independently without predetermining the output. Hence, the system receives a set of inputs x_1, x_2, x_3, …, x_t without the target output or the rewards from the environment. In this case, the framework for learning is based on finding patterns in the input data and building representations, which can help predict future inputs and decision-making. This is a method of learning a probabilistic model of data by estimating a model to represent the distribution for an input x_t, based on previous inputs x_1, x_2, x_3, …, x_{t-1}. This model can detect an abrupt change in the data and classification of the data by evaluating the probabilities [18]. Clustering techniques like the Gaussian mixture model, Expectation maximization, and K-means and dimensionality reduction are some methods used in UL.

Dynamic Movement Primitive (DMP) framework is widely used in PbD because of its smoothness and continuity of the generalized trajectories. However, it only uses one demonstration to learn the model. As learning the task from multiple demonstrations is preferable, Song et al. [15] proposed a Probability-based movement primitive (PbMP) that includes multiple demonstrations into one model using a probabilistic approach. It utilizes the concept of key points to detect motion units in the kinematic data from the demonstration. Input signals with respective regression models represent each trajectory segment. The model has different linear equations with different values for independent variables. The difference in the slopes of the variable point regression model represents the signal inconsistency between consecutive segments. Based on these difference scores, the non-Maximum suppression (NMS) algorithm is used to select representative candidate points. It uses an unsupervised segmentation algorithm that does not require the user to provide prior knowledge. It can extract the common feature from the demonstrated data using a hidden semi-Markov-model (HSMM). Estimation of HSMM parameters is done by using an expectation-maximization algorithm for all the sequences.

Gaussian mixture model (GMM), one of the most recurrent type of UL model, presets multiple Gaussian distributions into a fitted space of training dataset, called Gaussian mixture regression (GMR). This ML model is trained using expectation maximization (EM) algorithm. The new data point is classified according to the distribution where it most likely belongs. The task parameterized GMM (TP-GMM) is useful in robotic manipulators for adaptive trajectories. It is a variation to GMM that allows to perform GMR considering different frames for observation recording [17].

B. Supervised Learning

Supervised Learning (SL) algorithm is a ML technique that utilizes labeled data for training. The labeled data are identified in terms of characteristics, properties, or classifications and consist of input and corresponding output data. The pattern is learned from the labeled data that train the model to predict the output for new input data [3, 17].

If we consider human learning, SL is similar to learning from reading books [21]. SL includes regression, classification, hierarchical task networks, and neural networks [3]. Yongqiang et al. [13] proposed a self-supervised learning approach, generalization by self-supervised practicing (GSSP), that learns pouring skills from unsupervised demonstrations. The learned skill could achieve accuracy and speed compared to a human with a reduction in mean volume error lower than the state-of-the-art works. They utilize the recurrent neural network (RNN) based pouring skill model designed to process its inputs in order. The pouring skill model is generalized to different container shapes, liquid types, and granular materials. The input features to the RNN are the angular velocity, volume at time t, volume to be poured, the initial volume of a liquid in a source container, and the height H and body diameter D of the source container when modeled as a cylinder.

For robotic grasping, discriminative approaches are used that sample grasp candidates and rank them via neural network. The use of a grasp quality convolution neural network trained by using the dataset formed from the outcome of the physics simulation to grasp objects in randomized poses on a plane and the aligned crop of a depth image where the grasp is located helps predict the grasp success for given grasp candidates and depth images [1]. Using multiple layers of unit collection that interact with the input (pixel values when images are considered), Convolution neural networks (CNN) has its utility in image recognition and analysis, recommender systems, video processing, natural language processing (NLP), object recognition, and face recognition [19, 20]. Imitation learning is one of the methods that utilize SL. Expert demonstrations train the policy, and based on the dataset, the SL algorithms are launched to improve policy [21]. The ability of CNN to extract powerful image features and that of long short-term memory (LSTM) at predicting time series data has exhibited better performance when combined [20]. Simge et al. [20] proposed the CNN-LSTM version to achieve fact training with the best estimation accuracy. A comparison was made between CNN and CN-LStM models, which later presented better tracking accuracy and smoother estimation results. Inigo et al. [22] used DMP with CNN to find insertion tasks in demonstrations to find the suitable feature to extract that distinguishes insertion from other movements. The CNN combined with gripper activations help divide the tasks into phases Encoded and DMPs. This approach has proven to be more robust than regular DMPs.

Hsien-l et al. [23] utilized GMM to train the skin model that classifies the skin color and non-skin color from an image input, followed by hand position and orientation to translate and rotate the hand image to a neutral pose. CNN approach is adopted to classify seven types of human hand gestures. The GMM for modeling the skin color is validated. Multiple images were taken from a subject for each gesture type to train and test the CNN. As the convolution and subsampling of images help CNN learn the gestures, the proposed system is proven useful for various types of gestures. Also, using GMM and CNN together was indifferent to changes in the lighting conditions and was able to provide detailed features of the gestures that helped achieve best results.

Zen et al. [24] introduced semantic robot programming (SRP) for declarative robot programming over demonstrated scenes. The scenes can be perceived from RGBD observations via discriminatively-informed generative estimation of scenes and transfers (DIGEST). SRP uses R-CNN as a discriminative object detector to obtain a set of bounding boxes. Each bounding box outputs the confidence measure via a deep convolution neural network. R-CNN object detector is trained on the dataset that has 15 grocery objects.

C. Reinforcement learning

Reinforcement learning (RL) is a machine learning technique where the model learns by trial and error. Based on the feedback received from the environment, the model is trained to make decisions. The ultimate goal is maximizing the reward signal with state agent combination and feedback from the environment. Some widely used RL methods are Q-learning and deep reinforcement learning. An agent is a robotic system that performs an action a ϵ A and gets an observation s ϵ S, of the state of the environment from sensors. As the robot interacts with the environment, it changes. As a result of this, a reward is obtained by the agent r ϵ R from the environment, which rates the action based on its performance. The goal of the agent is to explore and discover an optimum policy that maximizes the reward [25].

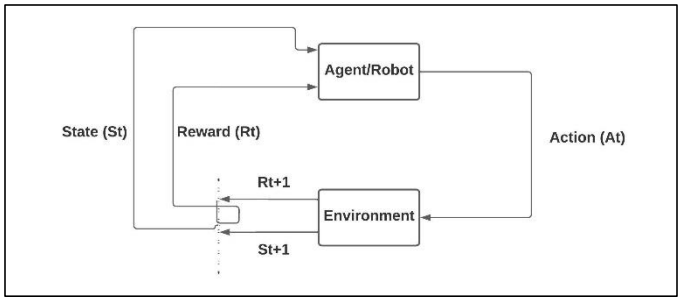

Fig. 2 Q-Learning Framework

RL helps the agent maximize the outcome (reward) based on trial and error (exploration), where the agent can perceive or infer information and adapt it adaptively towards changing environment, similar to the learning process in humans. Learning is inspired by punishment and reward resulting from the change of state in the environment [26]. In continuous action space, the DRL method can map from image inputs to the control policy, which is helpful in robot control. In discrete action spaces, DRL methods exhibit strong search capability within the high-dimensional decision space, which is helpful for the exploration and navigation of intelligent agents in a new environment [27]. Reinforcement learning differs from supervised learning as it does not need labeled input/output pairs. It focuses on finding the balance between exploration and exploitation. Reinforcement learning is modeled as a Markov decision process (MDP), which has a set of environment and agent states S, a set of actions of the agent A, a probability of transition from state s to s' at any time t, and an immediate reward R_a, after a transition from s to s' with an action a. One of the widely used RL method is Q learning.

$$q_{new}(s, a) = (1-\alpha)\, q(s, a) + \alpha\,(R(t) + 1 + \gamma \max q(s', a'))_{a'} \quad (1)$$

where q (s, a) in the old value, $(Rt + 1 + \gamma maxq(s', a')$ is the learned value, α is the learning rate ($0 < \alpha < 1$), and R(t) is the reward ant any time t. The framework of Q learning is demonstrated in fig 2.

Q-learning is a model-free RL algorithm where the value function Q is arbitrarily initialized. Depending on the action and the possible future states, its value is updated [17].

Sean et al. [10] used user feedback on the system's performance online and RL to train the interface. The approach scales with regular use; increased use of an interface to perform regular activities results in increased competence and personalization of the interface. The main challenge occurs due to the requirement of a large amount of training data from the sparsity of rewards. The authors propose a hierarchical solution to the problem: offline training is utilized to perform tasks, then the online user feedback is used to learn the mapping of user input to robot behavior. This is an example of human-in-the-loop learning which is quite common in rehabilitation robotics. First, the task-conditioned policy is pre-trained to perform various tasks without human interaction. Secondly, a user is in the loop in the online learning phase that utilizes RL with sparse, user-provided rewards to interpret the user's input. The optimal policy for the task is computed using the pre-trained task-conditioned policy by observing information from the task that the user completes. The algorithm is called assistive teleoperation via human-in-the-loop reinforcement learning (ASHA), which is evaluated with 12 participants using a webcam and eye gaze to perform three simulated manipulations, viz., flipping switches, opening a shelf, and rotating a valve. The algorithm learns to map 128-dimensional gaze feature to 7-dimensional joint torques in less than 10 minutes of online training and adapting to changing environments.

Robot learning by mapping between human inputs and their intended action using RL has also been used with work related to shared autonomy. Dylan et al. [2] designed a teleoperation algorithm for assistive robots to learn latent actions from task demonstrations. Unlike the teleoperation strategy, latent actions improved objective and subjective performance. Navigating an unknown environment by automatic exploration is also a key area for performing ADL tasks. A deep reinforcement learning-based decision algorithm, auxiliary task fully convolutional Q-Learning (AFCQN), is proposed by Haoran et al. [27] that utilizes a deep neural network to learn exploration strategy. Exploration is taken as a sequence decision-making task by the authors. They utilize the Markov decision process as a framework for decision-making. PbD requires extensive training for initial task learning and generalization of the learned model to a different environment. In robotic task learning problems, the deep model fusion (DMF) RL algorithm can efficiently generalize the learned task by model fusion that

helps to solve the problem of adaptation to a new environment. A multi-objective guided reward system converts sparse rewards to dense rewards to speed up the training process. The robot is pre-trained in various environments to obtain different policy models. When the environment changes, the DMF-RL method is used to improve the performance by the fusion of pre-trained policy models, as all the models have helpful information that is useful in boosting performance [28].

RL methods have also been used in humanoid motion planning of robotic arms that mimics human arm's motion. This process broadly includes two steps: (1) Extraction of human motion rules (HMR) and (2) RL training. Aolie et al. [14] used VICON optical motion capture system for HMR extraction to obtain the trajectory of human arm. For RL training, deep deterministic policy gradient (DDPG) and hindsight experience replay (HER) are adopted to train the humanoid motion of the robotic arm that combines former motion rules and designs corresponding reward functions. States of the robotic arm were analyzed and the action features on the robotic arm platform were compared with the human arm action.

PbD requires an accurate demonstration for the learning algorithm to be efficient. However, with human teachers, the demonstrations are not perfect all the time and can often provide incorrect information. Interactive RL learning allows the agent to learn quicker than non-interactive RL as the agent learns from two sources: Observation of the environment and feedback from a secondary critic source, like a human teacher or sensor feedback. However, the information provided by the critic is only sometimes perfect. Taylor et al. [29] introduced revision estimation from a partially Incorrect resources (REPaIR) framework that can estimate correction to imperfect feedback. Corruption function is defined between the correct and received reward, which is updated using a reward function R. To learn the reward functions from human demonstrations and preferences, Malayandi et al. [9] utilized a new framework of reward learning, DemPref that uses demonstrations in addition to preference queries to learn the reward function.

PbD has been an essential paradigm in learning ADL tasks with shared assistive control. However, more reactive assistive behavior can be generated by combining the motion from the demonstration with a real-time goal prediction method. This can also help reduce the number of joystick control inputs, a key factor in rehabilitation robotics. Calvin et al. [6] proposed a method that blends demonstration-generated assistive motion with user input based on goal predictions to achieve the task. The authors used the DMP-based assistive control method to predict the user goal by comparing the user input with DMP-generated assistive motion during the control process. They also compared the method with Partially observable Markov's Decision Process (POMDP) and direct control using a joystick. The time taken to complete the task and the number of user inputs required is the least in the method proposed by the authors.

Tymoteusz et al. [25] evaluated positioning accuracy, motion trajectory, and the number of steps required to position a robotic arm task using various RL algorithms. DDPG, twin delayed deep deterministic policy gradient (TD3), soft actor-critic (SAC), and HER were evaluated in six different combinations: DDPG, TD3, SAC, DDPG+HER, TD3+HER, and SAC+HER. Sparse, dense, and dense trajectory reward functions were tested for each of the six combinations. The advantage of combining DDPG, TD3, and SAC with HER was seen for sparse reward. DDPG and DDPG+HER were found to be best for dense rewards. Finally, for dense trajectory reward, the smallest positioning error was obtained for TD3, and the least standard deviation was obtained for the DDPG+HER algorithm. Yoan et al. [7] used tree boosted relational imitation learning (TBRIL) to learn the policy close to the one demonstrated. The Authors propose a task refining process based on the GUI that shows the user the elements of the task learned by the system and allows them to correct it.

IV. Discussion

PbD is a method widely used in robot learning as it reduces complex programming to achieve the manipulation task and relies on demonstration provided by a human to replicate the task. However, the crucial factor for the PbD to be efficient is the quality of the demonstration and the availability of labeled data for predicting the outcome based on the input signal. It is helpful in rehabilitation robotics, considering the ability of the user to tele operate to perform dexterous manipulation tasks. A complete task can be broken down into actions, and the sequence of actions for ADL tasks can be demonstrated and stored as a data set. Also, a set of actions can be common to multiple tasks, and learning these actions would help complete various tasks autonomously. PbD depends on the quality of the demonstration and availability of labeled data, which is not possible in all cases.

Moreover, the human-in-the-loop constantly needs to provide feedback if the task is not performed as desired. Hence, this leads to a requirement for a robust learning method that is capable of learning the policy from the demonstration and predicting desirable output from given input based on either the probabilistic approach or maximizing the reward function. The three broad methods, viz., unsupervised, supervised, and reinforcement learning, have been discussed in the previous section with detailed descriptions of each method and recent techniques developed and used for robot learning. The choice of a particular method will depend on the cost and duration of the training, availability of data as discrete/continuous or labeled/unlabeled, the approach and quality of demonstration, the complexity of the manipulation task involved, and adaptability to a constantly changing environment. For example, UL is feasible in terms of cost and duration of the training and can use unlabeled data to predict the possible output. However, it is not feasible when the complexity of the manipulation task and adaptability to a changing environment is considered, as it relies on a probabilistic model and predicts

the desired output based on unlabeled input data. Conversely, SL is based on neural networks that can train a model to perform complex manipulation tasks and adapt to a changing environment. However, it requires labeled data to predict the output based on previous input/output combinations.

RL techniques have proven to be promising in PbD, which learns the model based on exploration and exploitation and maximizes the reward function to produce a desired output. The recently proposed RL techniques have been discussed in the above section that solves problems related to the quality of demonstration and incorrect feedback, navigation in a novel environment, predicting user goal in shared assistive control, integrating human demonstrations with preferences, humanoid motion planning of a robotic arm, controlling high DoF robots with low DoF latent actions, and assistive teleoperation with human-in-the-loop. Most of the methods mentioned above are useful when efficient learning with shared autonomy is considered, which is the case in rehabilitation robotics. While it has many advantages, some limitations include the Exploration-Exploitation trade-off, i.e., a balance between the two, designing appropriate reward functions, Computational limitation with increased state space, and requirement of a large amount of data for efficient learning. Blending the RL techniques has proven to help eliminate some of the problems. Although RL is the most efficient method that helps efficient training of the model, it may have difficulty generalizing to an unseen environment. This limits its usefulness in real-world applications.

Future work includes exploring the methods and combining them to eliminate the limitations of individual methods. Intense research is required to eliminate the evident problems in robot learning and minimize the difference between human demonstration and learned action by utilizing the models for complete and efficient learning.

V. Conclusion

This review provides an overview of the significance of Programming by demonstration (PbD) in robot learning, different approaches of demonstration used in PbD, the associated machine learning methods and techniques for efficient robot learning, the strength and limitations of the learning methods, and a future direction of the research associated with the field of PbD. Several aspects of PbD using a learning-based approach was studied. We identified different approaches of demonstration followed by a detailed discussion of machine learning methods and techniques that are available under each method, the state-of-the-art methods proposed by different researchers and their use in robot learning, and a brief discussion on the strengths and limitations of the learning methods and scope of future work. PbD has potential in the field of robot learning because of the ease of programming without any expert skills and uses a demonstration approach which is the most efficient way to teach. Blending PbD with machine learning techniques helps in efficient learning of the task and adapting to a change in the environment. This is helpful in shared autonomy, where the robot learns from demonstration and a training model to achieve a desired output. Also, the feedback provided by the user can be utilized to learn any deviation from the previously learned task. This is an essential field in robotics that can enable robots to efficiently learn from a human teacher and mimic the action to provide a desired output.

References

[1] Kilian Kleeberger, Richard Bormann, Werner Kraus, and Marco F. Huber, "A Survey on Learning-Based Robotic Grasping," Current Robotics Reports (2020) 1:239-249, https://doi.org/10.1007/s43154-020-00021-6

[2] Dylan P. Losey, Krishnan Srinivasan, Ajay Mandlekar, Animesh garg, and Dorsa Sadigh," Controlling Assistive Robots with Learned Latent Actions," 2020 IEEE International Conference on Robotics and Automation (ICRA), 31 May - 31 August, 2020. Paris, France

[3] Harish Ravichandar, Athanasios S. Polydoros, Sonia Chernova, and Aude Billard, "Robot Learning from Demonstration: A Review of Recent Advances," Annual review of Control, Robotics, and Autonomous Systems YYYY. AA:1-33, https://doi.org/10.1146

[4] Gbenga Abiodun Odesanmi, Qining Wang, and JIngeng Mai, "Skill Learning framework for human-robot interaction and manipulation tasks," Robotics and Computer-Integrated Manufacturing, https://doi.org/10.1016/j.rcim.2022.102444

[5] Yanqiang Mo, Hikaru Sasaki, Takamitsu Matsubara, and Kimitoshi Yamazaki," Multi-step motion learning by combining learning-from demonstration and policy-search," https://doi.org/10.1080/01691864.2022.2163187

[6] Calvin Z. Qiao, Maram Sakr, Katharina Muelling, and Henny Admoni, "Learning from Demonstration for Real-Time User Goal Prediction and Shared Assistive Control," 2021 IEEE International Conference on Robotics and Automation (ICRA 2021), doi: 10.1109/ICRA48506.2021.9560758

[7] Yoan Mollard, Thibaut Munzer, Andrea Baisero, Marc Toussaint, and Manuel Lopes, "Robot Programming from Demonstration, Feedback and Transfer," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

[8] Thomas Eiband, Christoph Willibald, Isabel Tannert, Bernhard Weber, and Dongheui Lee, "Collaborative programming of robotic task decisions and recovery behaviors," Autonomous Robots (2023) 47:229–247, https://doi.org/10.1007/s10514-022-10062-9

[9] Malayandi Palan, Nicholas C. Landolfi, Gleb Shevchuk, and Dorsa Sadigh, "Learning Reward Funtions by Integrating Human Demonstrations and Preferences," arXiv:1906.08928

[10] Sean Chen, Jensen Gao, Siddharth Reddy, Glen Berseth, Anca D. Dragan, and Sergey Levine," ASHA: Assistive Teleoperation via Human-in-the-Loop Reinforcement Learning," arXiv:2022.02465

[11] Suhas Kadalagere Sampath, Ning Wang, Hao Wu, and Chenguang Yang, "Review on human-like robot manipulation using dexterous hands," Cognitive computation and systems, DOI: 10.1049/ccs2.12073

[12] Lei Huang, Zihan Zhu, and Zhengbo Zou, "To imitate or not to imitate: Boosting reinforcement learning-based construction robotic control for long-horizon tasks using virtual demonstrations," Automation in construction, https://doi.org/10.1016/j.autcon.2022.104691

[13] Yongqiang Huang, Juan Wilches, and Yu Sun, "Robot gaining accurate pouring skills through self-supervised learning and generalization," Robotics and autonomous systems, https://doi.org/10.1016/j.robot.2020.103692

[14] Aolei Yang, Yanling Chen, Wasif Naeem, Minrui Fee, and Ling Chen, "Humanoid motion planning of robotic arm based on human arm action

feature and reinforcement learning," https://doi.org/10.1016/j.mechatronics.2021.102630

[15] Caiwei Song, Gangfeng Liu, Xuehe Zhang, XIzhe Zang, Congcong Xu, and Jie Xhao, "Robot complex motion learning based on Unsupervised trajectory segmentation and movement primitives," ISA Transactions 97 (2020) 325-335, https://doi.org/10.1016/j.isatra.2019.08.007

[16] Michael J. McDonald, and Dylan Hadfield-Menell, "Guided Imitation task and Motion planning," 5th Conference on Robot Learning (CoRL 2021), London, UK

[17] Francesco Semeraro, Alexander Griffiths, and Angelo Cangelosi, "Human-robot collaboration and machine learning: A systematic review of recent research," Robotics and Computer-Integrated Manufacturing 79 (2023) 102432, https://doi.org/10.1016/j.rcim.2022.102432

[18] O. Bousquet et al.: Machine Learning 2003, LNAI 3176, pp. 72–112, 2004

[19] Ajay Shrestha, and Ausif Mamood, "Review of deep learning algorithms and architectures," IEEE Access doi: 10.1109/ACCESS.2019.2912200

[20] Simge Nur Aslan, Recep Ozalp, Aysegul Ucar, and Cuneyt Guzelis, "New CNN and Hybrid CNN-LSTM models for learning object manipulation of humanoid robots from demonstration," Cluster Computing (2022) 25:1575–1590, https://doi.org/10.1007/s10586-021-03348-7

[21] Zhihao Liu, Quan Liu, Wenjun Xu, Lihui Wang, and Zude Zhou, "Robot learning towards smart robotic manufacturing: A review," Robotics and Computer-Integrated Manufacturing 77 (2022) 102360

[22] Inigo Ituratte, Etienne Roberge, Esben Hallundbaek Ostergaard, Vincent Duchaine, and Thiusius Rajeeth Savarimuthu, "Improving the generalizability of Robot Assembly tasks learned from demonstration via CNN-based segmentation," 2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)

[23] Hsien-l Lin, and Y. P. Chiang, "Understanding human hand gestures for learning robot pick-and-place tasks," International journal of advanced robotic systems, doi: 10.5772/60093

[24] Zhen Zeng, Zheming Zhou, Zhiqiang Sui, and Odest Chadwicke Jenkins, "Semantic Robot programming for Goal-directed manipulation in cluttered scenes," 2018 IEEE International Conference on Robotics and Automation (ICRA), May 21-25, 2018, Brisbane, Australia

[25] Tymoteusz Lindner, Andrzej Milecki, and Daniel Wyrwal, "Positioning of the Robotics arm using different Reinforcement Learning Algorithms," International Journal of Control, Automation and Systems, http://dx.doi.org/10.1007/s12555-020-0069-6

[26] Rongrong Liu, Florent Nageotte, Philippe Zanne, Michel de Mathelin, and Birgitta Dresp-Langley, "Deep reinforcement learning for the control of robotic manipulation: A focussed Mini-review," Robotics 2021, 10, 22. https://doi.org/10.3390/robotics10010022

[27] Haoran Li, Qichao Zhang, and DOngbin Zhao," Deep Reinforcement Learning-Based Automatic Exploration for Navigation in Unknown Environment," IEEE transactions on neural networks and learning systems, vol. 31, no. 6, June 2020

[28] Tianying Wang et. al, "Efficient Robotic Task Generalization Using Deep Model Fusion Reinforcement Learning," IEEE International Conference on Robotics and Biomimetics Dali, China, December 2019

[29] Taylor A. Kessler Faulkner, Elaine Schaertl Short, and Andrea L. Thomaz," Interactive Reinforcement Learning with Inaccurate Feedback," 2020 IEEE International Conference on Robotics and Automation (ICRA), 31 May - 31 August, 2020. Paris, France

[30] Liu, R, Nageotte, F, Zanne, P, de Mathelin, M, and Dresp-Langley, B, "Deep Reinforcement Learning for the Control of Robotic Manipulation: A Focussed Mini-Review," Robotics 2021, 10, 22. https://doi.org/10.3390/ robotics10010022

[31] Marcos Maroto-Gomez, Sara Marques-Villaroya, Jose Carlos Castillo, Alvaro Castro-Gonzalez, and Maria Malfaz, "Active Learning based on Computer vision and human-robot interaction for the user profiling and behavior personalization of an autonomous social robot," Engineering Applications of Artificial Intelligence 117 (2023) 105631

[32] Alessandro Pieropan, Giampiero Salvi, Karl Pauwels, and Hedvig Kjellstorm," Audio-Visual Classification and Detection of Human Manipulation Actions," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), September 14-18, 2014, Chicago, IL, USA

[33] Jerry Zhi-Yang He, Aditi Raghunathan, Daniel S. Brown, Zackory Erickson, and Anca D. Dragan, "Learning Representations that Enable Generalization in Assistive Tasks," 6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand

[34] Marina Kollmitz, Torsten Koller, Joschka Boedecker, and Wolfram Burgard," Learning Human-aware Robot Navigation from Physical Interaction via Inverse Reinforcement Learning," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 25-29, 2020, Las Vegas, NV, USA

[35] Daniele Meli, Hirenkumar Nakawala, and Paolo Fiorini," Logic Programming for deliberative robotic task planning," https://doi.org/10.1007/s10462-022-10389-w

[36] Lingfeng Tao, Jiucai Zhang, and Xiaoli Zhang," Multi-Phase Multi-Objective Dexterous Manipulation with Adaptive Hierarchical Curriculum."

[37] Lukas Brunke et al, "Safe learning in Robotics: From Learning-based Control to safe reinforcement Learning," https://doi.org/10.1146/annurev-control-042920- 020211, Annual review of Control, Robotics, and Autonomous systems

[38] Kirill Kronhardt, Max Pascher, and Jens Gerken, "Understanding Shared control for Assistive Robotic arms," arXiv:2303.01993

[39] Sergio Spano et al, "An Efficient Hardware Implementation of Reinforcement Learning: The Q-Learning Algorithm." doi: 10.1109/ACCESS.2019.2961174

[40] Mingshan Chi, Yaxin Liu, Yufeng Yao, Yan Liu, Shouqiang Li, Chao Zeng, and Ming Zhong, "Development and evaluation of demonstration information recording approach for Wheelchair mounted robotic arm." Complex & Intelligent Systems, https://doi.org/10.1007/s40747-021-00350-9