

Review

# Learning from Demonstrations in Human–Robot Collaborative Scenarios: A Survey

Arturo Daniel Sosa-Ceron , Hugo Gustavo Gonzalez-Hernandez \*  and Jorge Antonio Reyes-Avendaño 

School of Engineering and Sciences, Tecnológico de Monterrey, Monterrey 64849, NL, Mexico

\* Correspondence: hgonz@tec.mx

**Abstract:** Human–Robot Collaboration (HRC) is an interdisciplinary research area that has gained attention within the smart manufacturing context. To address changes within manufacturing processes, HRC seeks to combine the impressive physical capabilities of robots with the cognitive abilities of humans to design tasks with high efficiency, repeatability, and adaptability. During the implementation of an HRC cell, a key activity is the robot programming that takes into account not only the robot restrictions and the working space, but also human interactions. One of the most promising techniques is the so-called Learning from Demonstration (LfD), this approach is based on a collection of learning algorithms, inspired by how humans imitate behaviors to learn and acquire new skills. In this way, the programming task could be simplified and provided by the shop floor operator. The aim of this work is to present a survey of this programming technique, with emphasis on collaborative scenarios rather than just an isolated task. The literature was classified and analyzed based on: the main algorithms employed for Skill/Task learning, and the human level of participation during the whole LfD process. Our analysis shows that human intervention has been poorly explored, and its implications have not been carefully considered. Among the different methods of data acquisition, the prevalent method is physical guidance. Regarding data modeling, techniques such as Dynamic Movement Primitives and Semantic Learning were the preferred methods for low-level and high-level task solving, respectively. This paper aims to provide guidance and insights for researchers looking for an introduction to LfD programming methods in collaborative robotics context and identify research opportunities.

**Keywords:** human–robot collaboration; robot learning; learning from demonstrations; skill learning



**Citation:** Sosa-Ceron, A.D.; Gonzalez-Hernandez, H.G.; Reyes-Avendaño, J.A. Learning from Demonstrations in Human–Robot Collaborative Scenarios: A Survey. *Robotics* **2022**, *11*, 126. <https://doi.org/10.3390/robotics11060126>

Academic Editor: Thierry Chaminade

Received: 23 September 2022

Accepted: 10 November 2022

Published: 15 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Manufacturing industry has evolved; traditional manufacturing—focused on mass production—has adopted a new production scheme based on mass customization, capable of satisfying the needs of an ever changing market. The mass customization paradigm has highlighted the importance of possessing enough flexibility and adaptability in manufacturing cells, to resolve this, the concept of Industry 4.0 was born.

Industry 4.0 is a knowledge-based and data-centered manufacturing strategy focused on increasing flexibility and adaptability of a production process while a high level of automation, operational productivity, and efficiency are achieved [1]. Current robotics systems in manufacturing processes allows high precision and repeatability to facilitate different products manufacturing while increasing the efficient use of resources, whether material or temporary, and this provides a considerable reduction in production costs. On the other hand, industrial robots have benefited in their autonomy levels, thanks to the technological advances made in areas of computer science and data acquisition and processing, particularly in the fields of Artificial Intelligence (AI), sensors, and microprocessors to make them become smarter [2]. Although industrial robots offer high levels of precision, reliability, and speed, among other advantages as their key points within the manufacturing processes, they still lack certain necessary characteristics, such as flexibility,

adaptability, or intelligence, to be fully compatible with the idea of smart manufacturing introduced by this shift [3].

One enabling technology for Industry 4.0 is Human–Robot Collaboration (HRC), an interdisciplinary research area involving advanced robotics, neurocognitive modeling, and psychology. HRC refers to application scenarios where a robot, generally a collaborative one (commonly called a Cobot), and a human occupy the same workspace and interact to perform tasks with a common goal [4]. Within manufacturing, HRC seeks to combine the impressive physical capabilities of robots (e.g., high efficiency and repeatability), with the cognitive abilities of humans, (e.g., flexibility and adaptability), necessary to address the uncertainties and variability in massive customization [5].

Still, the field of HRC presents several unsolved industrial and research challenges in terms of intuitiveness, safety, and ergonomic human–robot interaction [6–9], and in robot learning and programming methods [10,11]. This has led to scenarios where collaborative robots are reduced to traditional uncaged manipulators, for which, as a consequence, flexible, adaptive, and truly collaborative tasks are rarely implemented or developed [12]. In an effort to facilitate the programming of collaborative robots, Learning from Demonstration (LfD) appeared as a solution [13]. This is a technique that mimics how humans learn a task by watching a demonstration from another human. The use of LfD as a programming technique simplifies the complex skill of code programming, making the use of the robot accessible, without the need for a programming expert. Thus, the manufacturing preparation is accelerated, as the human collaborator could teach the robot any new desired skill.

The aim of this paper is to investigate the current state of the art on LfD as an advanced programming technique for collaborative robots and to identify how to improve HRC in order to reduce the complexity faced by the human operator in the reconfiguration of manufacturing cells and reprogramming of robot tasks. Based on the most important and consolidated topics in programming techniques for collaborative robots research, this work is intended to serve as a guide for scientists looking for directions on where to focus in the near future to bring collaborative robotics from the laboratory experimentation to implementation on the industrial shop-floor. Thus, this paper seeks to answer the following research questions:

- RQ1: Which are the main programming algorithms applied in the recent scientific literature regarding Learning from Demonstrations for collaborative robots applicable to possible manufacturing scenarios?
- RQ2: Which is the level of collaboration/interaction between human and robot in the tasks that researchers seek to solve with the application of Learning from Demonstration programming algorithms?
- RQ3: How do these solutions align with smart manufacturing/Industry 4.0 paradigm in terms of intuitiveness, safety, and ergonomics during demonstration and/or execution?

Several terms with similar meanings can be found in the literature: Programming by/from Demonstrations, Imitation Learning, Behavioral cloning, Learning from Observation, and Mimicry. In this work, we use the term LfD to encompass all these terms relative to the field.

In recent years, other papers that extensively review the LfD approach for robot programming have been published, as shown in Table 1 including review and survey papers. The work of Hussein et al. [14] emphasizes on the more prominent algorithms applied in research on Imitation Learning, Zhu et al. [15] review how LfD is used to solve manufacturing assembly tasks, the paper by Ravichandar et al. [16] aims to show the advances of LfD in the robotics field, and Xie et al. [17] show a broader perspective on how LfD is applied in path planning operations. In contrast, the uniqueness of this review is the particular focus on how LfD directly impact the HRC in manufacturing operations, and attempts to emphasize the active role of the human during the entire teaching–learning process.

**Table 1.** Published review papers on Robot Learning from Demonstrations.

Year	Author	Title	Focus
2017	Hussein et al. [14]	Imitation Learning: A survey of learning methods	Algorithms
2018	Zhu et al. [15]	Robot Learning from Demonstration in Robotic Assembly: A survey	Assembly operations
2020	Ravichandar et al. [16]	Recent Advances in Robot Learning from Demonstration	General
2020	Xie et al. [17]	Robot learning from demonstration for path planning: A review	Path planning

The rest of the paper is organized as follows: Section 2 describes the research methodology used in this work based on a systematic literature review (SLR) methodology to identify relevant papers for this study. In Section 3, we give a detailed theoretical background in LfD paradigm. In Section 4, the retrieved papers are analyzed and classified in representative categories of each applied algorithm, and the human role during LfD process. Finally, Section 5 is dedicated to discussing the findings from the content analysis and the implications in real scenarios.

## 2. Materials and Methods

We applied a systematic literature review (SLR) in this study because it is based on a systematic, method-driven, and replicable approach. There are several studies on how to conduct a SLR [18–20].

For this work, we defined the following four consecutive steps for our study:

- Step 1: Establish the research objectives of the SLR.
- Step 2: Define the conceptual boundaries of the research.
- Step 3: Organize the data collection by defining the inclusion/exclusion criteria.
- Step 4: Report the validation procedure and efforts.

In the following, we discuss each one of these steps.

### 2.1. Research Objectives of the SLR

As the research questions stated in the introduction, we use the SLR approach to identify the most used algorithms in Learning from Demonstration (LfD) programming for collaborative robots in manufacturing industrial settings. In particular, we want to understand how the results in recent research on the field can be categorized and if the solutions introduced truly focus on solving collaborative tasks, how it connects and relates with the new emerging technologies introduced by Industry 4.0 and Smart Manufacturing concepts, and in which areas we need to focus in the future to successfully implement these approximations and to truly enhance Human–Robot Collaboration (HRC) in industrial settings.

### 2.2. Conceptual Boundaries

This research aims to analyze collaborative robotics. Thus, the setting of the conceptual boundaries was based on the terms “collaborative robotics” and its derivations combined with terms describing its application in an industrial environment (e.g., “industrial”, “production”, “manufacturing”, or “assembly”) while, considering the terms “Learning from Demonstrations” (LfD) or “Programming by Demonstrations” (PbD) and all the possible synonyms (“behavioral cloning”, “imitation learning”, “kinesthetic teaching”). Additionally, to completely understand how collaborative robots in an industrial context relate to the core technologies of Industry 4.0, we added the terms “digital twin” and “cyber-physical systems” to the main search.

### 2.3. Inclusion and Exclusion Criteria

In addition to the conceptual boundaries, several search criteria in terms of database, search terms and publication period need to be defined. Web of Science (WoS) and Scopus were selected as electronic databases for the keyword search, which we identified as two of the most complete and relevant databases for publications in the Engineering and Manufacturing area. Most of the source results obtained using Scopus can be smoothly complemented with the search on Web of Science, therefore, we decided to use both databases to conduct the SLR. For the applied search approach and the inclusion and exclusion criteria in the search query, we decided to divide this task in multiple steps for the evaluation and construction of the search query, as proposed in [21]. In the first step, we identified the literature of the collaborative robotics field using the following search terms while searching for title, abstract and keyword in Scopus and for topic in WoS: “collaborative robotics”, “collaborative robot”, “cobot”, “human-robot collaboration”, “human robot”, “human-robot”; all the terms were linked with the “OR” operand. In this first step, all kinds of subject areas and documents were included, but only works in the English language were selected. In a second step, the focus was on collaborative robots as solutions to the industrial sector. Therefore, in this second step, the following terms were added through a boolean “AND” function: “industry”, “industrial”, “manufacturing”, and “production”; all the terms were linked with the “OR” operand. Finally, in the third step, we completed the search query by constraining the publishing year of the academic literature, thus, we selected works from 2016 to 2022 as time span for this research and also limited the source type to “Journal” and “Article”, “Review” and “Article in Press” as document type to consider only high-quality literature.

### 2.4. Validation of the Search Results

The screening was conducted in two phases by three independent researchers. In the first phase (1° round of screening), only the title and abstract were read. In the second phase (2° round of screening), the whole paper was examined. Each researcher classified each paper in two categories: high relevance and low relevance. Where the three independent researchers came to the same conclusion, i.e., zero differences in the classification, the papers were directly included into the analysis. Papers where differences in the classification occurred were discussed in order to achieve 100% agreement between the three researchers. Additionally, the final decision was supported by a text mining software in order to provide more reliability to the results; the benefits of using these kinds of tools are extensively described in [22,23]. After the screening phases, a total of 43 papers were included in this survey.

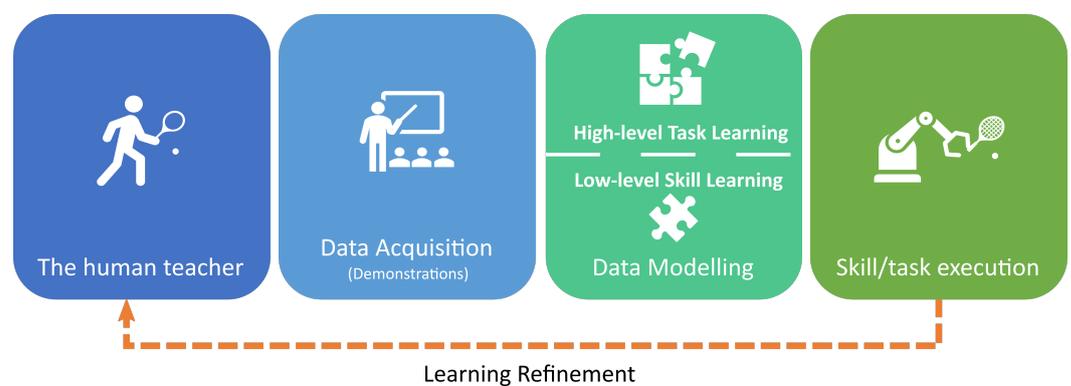
## 3. Theoretical Background

This section is dedicated to the description of common terms used in the research area. As it is a rapidly grown field, several authors use different words to express similar concepts. In this section, we unify and describe the concepts, for a better understanding of this paper. The concepts expressed here will be used later in Section 4 for the analysis.

### 3.1. Learning from Demonstration

Robot Learning from Demonstration (LfD) is a technique for robot programming, based on a collection of learning algorithms, inspired by how humans imitate behaviors to learn and acquire new skills. The motivation behind this technique is to offer the end-user the capability to adapt and customize a robotic system to their needs, without prior programming knowledge or skills. The robot learns the desired behaviors through a natural demonstration of the target skill performed by the end-user. The main idea is that the robot can reproduce the learned skill accurately and efficiently to solve a task. Furthermore, the robot should be able to use the learned skill in situations not taught in the demonstrations. LfD takes aspects of neuroscience to try to imitate the way humans learn after watching a demonstration from another human.

Based on the extensive works of Argall et al. [13], Billard et al. [24], and Chernova et al. [25], in Figure 1, we show the general outline of the LfD process. It consists of five main stages in the design of applications based on LfD programming: Demonstrator Selection, Data Acquisition, Data Modeling, Skill/Task Execution, and Learning Refinement. The first stage refers to the selection of the demonstrator. Ideally, the demonstrator should be a skilled teacher, however, the characteristics of the demonstrator are not always considered. Second is the Data Acquisition, also called the demonstrations stage, where a demonstrator teaches the skill or task, and the cobot records such demonstration using its available sensors. Third, Data Modeling is the creation and application of algorithms used for learning the demonstrated skill or task. Fourth, Skill/Task Execution stage is where the knowledge obtained by the robot is evaluated. Finally, the fifth stage, Refinement of the Learning, is the stage that completes the LfD loop. Whether online or not, its purpose is to gradually improve the learned robot skill towards a better generalization.



**Figure 1.** The five stages of the Learning from Demonstration programming paradigm.

### 3.1.1. Demonstrator Selection Stage

The correct selection of the demonstrator has great repercussions for the LfD system design. Usually, a human is selected to be the demonstrator, but this is not necessarily the only option. Theoretically, another intelligent system could be selected as a demonstrator; however, this has not been attempted in the literature. The characteristics that should be considered are level of expertise, physical and cognitive capabilities, and the number of demonstrators. In most cases, the demonstrator is considered an expert teacher; however, suboptimal demonstrations lead to suboptimal robot (learner) performance. Physical and cognitive capabilities play a significant role in the selection of the ideal interface for demonstrations. An adequate compromise between the correspondence problem and performance should be established.

### 3.1.2. Data Acquisition Stage

Data acquisition refers to the way skills or tasks are captured during the demonstration process. The objective is to extract the critical information from the task and provide the robot with the ability to adapt the learned skill to different situations. This stage responds to the question of what to learn/teach? The interface used to teach is key during this process (also connects the demonstrator with the learning system) because each learning method employed during this stage has advantages and disadvantages associated with the correspondence problem. Careful attention must be given to the sensory and physical systems of the demonstrator, the learner, and the space where the demonstration is taking place, e.g., the perception of the task space from a camera mounted on the robot end-effector will differ greatly from the perception of the demonstrator's eyes. In the same context, even with all the similarities that exist, the movements of a human arm will not have the exact same correspondence in a robot arm configuration space. In short, to solve the correspondence problem, it is necessary to provide a map between the desired trajectories

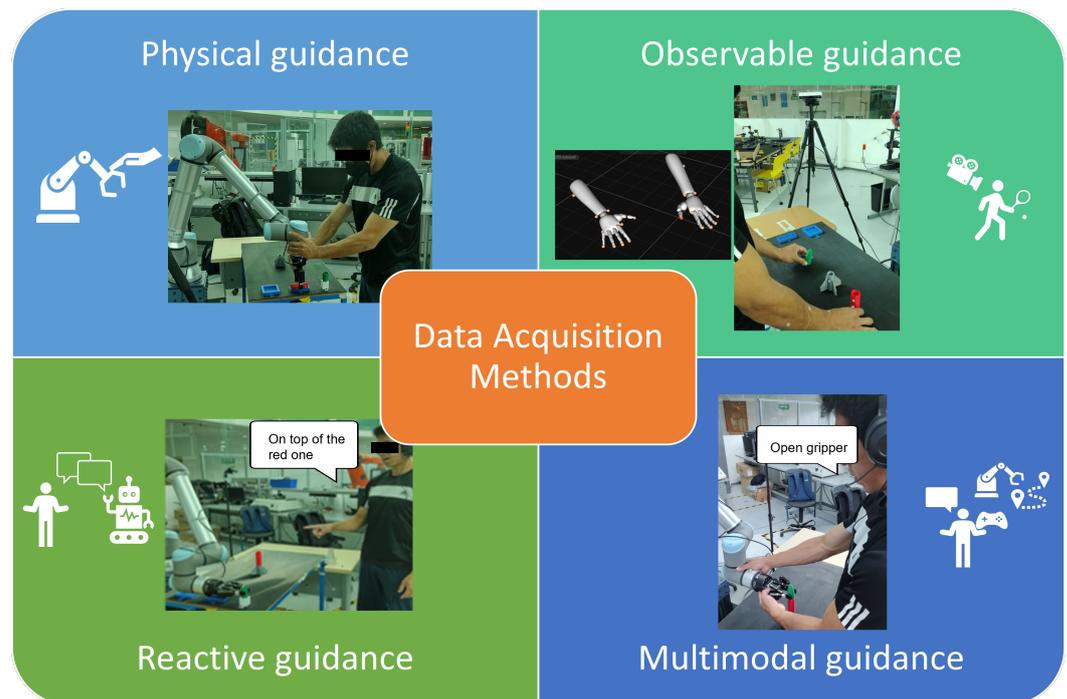
and the joint/configuration space of the learner. There are two types of mapping methods: Direct-mapping-based methods and Indirect-mapping-based methods.

In the Direct-mapping-based methods, the robot is manipulated directly by the demonstrator to record the desired learning trajectories. The advantage of these methods is that the correspondence mapping function can be omitted because all demonstration steps are performed in the robot configuration space. Nevertheless, difficulty in obtaining smooth and effective trajectories is increased due to the complexity faced by the demonstrator in manipulating the body of the robot, especially in robotic systems with high degrees of freedom [24,26].

In the Indirect-mapping-based methods, the demonstrator teaches the desired trajectories in its own configuration space and later converts the information to the robot configuration space [24]. The most important advantages of this method are intuitiveness for the demonstrator and smoother trajectories generation but finding a matching correspondence function could be difficult [25]. For example, techniques such as dimensionality reduction [27,28], or novel data acquisition methods, such as the ones described in the next Section 4, could mitigate this problem.

In the Data Acquisition stage, we can identify four acquisition methods graphically depicted in Figure 2 and fully described below:

- Acquisition by physical guidance: In this kind of acquisition method, the demonstration is performed on the configuration space of the robot by manually positioning along the desired trajectories to learn. Here, interfaces such as Kinaesthetic teaching, teleoperation, and haptics are commonly used. In Figure 2, an example of Kinaesthetic teaching is provided, one disadvantage of this method is the lack of precision in the recording due to the difficulties experienced by the user to reach some positions manually.
- Acquisition by observable guidance: In this context, the demonstrator performs the demonstration in its own configuration space and then uses Indirect-mapping-based techniques to relate the trajectories to the robot configuration space. Interfaces such as inertial sensors, vision sensors, and motion capture systems are commonly used [29]. Figure 2 also shows an example of this method using a motion capture system—one disadvantage is the impossibility of recording contact forces during the trajectories.
- Acquisition by reactive guidance: Different from previous methods, the robot is allowed to perform the task during the demonstration stage by exploring the effects of a predefined set of actions it is allowed to perform in a particular state. Whereas, the demonstrator, from time to time, suggests which actions suit better for a particular state or even directly selects the desired action that the robot should take. Thus, the data from the demonstration are shaped according to the feedback that the demonstrator provides during the acquisition process, avoiding the correspondence problem, by recording the demonstration data directly to the robot configuration space [25]. Methods such as reinforcement learning (RL) and active learning (AL) are widely used [30]. Figure 2 shows how the demonstrator provides suggestion to the robot using a Natural Language interface.
- Acquisition by multimodal guidance: This method uses any combination of the previous methods to better adjust the data recorded through the demonstration, and allows a more natural interaction between robot and demonstrator; this is illustrated in Figure 2. For example, Wang et al. [31] used a combination of observable guidance (a 3D vision system) and physical guidance (force/torque sensory system) to teach an assembly task. Detrimental to this method is the increased complexity for data fusion from multiple input sources.



**Figure 2.** The four Data Acquisition methods available during the LfD process: Physical guidance through Kinaesthetic teaching, Observable guidance using a motion capture system to record the trajectories, Reactive guidance where the hints necessary to solve the task are provided to the robot through Natural Language Processing (NLP), and Multimodal guidance where Kinaesthetic teaching and NLP are used simultaneously for Data Acquisition.

### 3.1.3. Data Modeling Stage

Data Modeling allows the robot to use the information obtained through demonstrations to learn a set of rules (policy) to map the demonstrated actions to the states seen, and ultimately tries to infer which set of actions can be applied to states unobserved during the demonstrations. Depending on the complexity of the demonstrated task, current learning methods can be categorized into low-Level skill learning and high-Level task learning.

#### Low-Level Motions

Low-Level skills are the primitive motions that a robot is capable of performing, e.g., pushing, placing, grasping, etc. The combination of skills allows the robot to complete a more complex task, e.g., pick and place. In the context of the LfD problem, it is crucial to define the appropriate state-space representation for the desired skill. While for the robot the simplest state to record and reproduce is its joints positions of the entire kinematic chain over time, there are cases in which several different state spaces are needed to promote learning the general form of a skill [25].

For instance, in push/pull skill learning, instead of representing the trajectory of the moving target object in the robot joint space, a representation of the same trajectory in the task frame (Cartesian position of the end-effector with respect to the target object) will be more useful in terms of generalization. Moreover, augmenting the state-space representation to include internal and external contact forces in the desired skill learning allows the representations to be generic, to represent any kind of motion.

In low-Level skill learning, there are two dominant learning strategies.

- One-shot or deterministic learning approaches are generally modeled as non-linear deterministic systems where the goal is to reach a particular target by the end of the skill. A commonly used approach in the literature is Dynamic Movement Primitives (DMP) introduced by Ijspeert [32] in order to represent an observer behavior in an

attractor landscape. Learning then consists of instantiating the parameters modulating these motion patterns. Learning a DMP typically requires just a single skill demonstration, this has advantages in the LfD problem in terms of simplicity, but it makes the system more vulnerable to noisy demonstrations as the resulting controller will closely resemble the seed demonstration.

- Multi-shot or probabilistic learning is based on statistical learning methods. Whereas, the demonstration data is modeled with probability density functions, and the learning process is accomplished by exploiting various non-linear regression techniques from machine learning. In this method, multiple demonstrations are needed during the learning process, this property makes them more robust to noisy demonstrations, but more complex to implement due to the number of demonstrations needed. Most of the probabilistic algorithms in LfD are based on the general approach of modeling an action with Hidden Markov Models (HMMs) [33,34]. More recently, works have used HMMs method for skill learning, combined with Gaussian Mixture Models (GMM) [35] and Gaussian Mixture Regression (GMR) [36,37].

### High-Level Tasks

High-level task learning generally is composed of a set of primitive motions (low-level skills) combined to perform a more complex behavior. In this case, the state-space representation consists of a set of features that describe the relevant states of the robot environment and, instead of trajectories, the demonstrations provided are sequences of primitive motions. Learning complex tasks is the goal of LfD, and there are three different ways in which is possible to group the learning process of a task, these are policy learning, reward learning, and semantics learning.

- Policy learning consists of identifying and reproducing the demonstrator policy by generalizing over a set of demonstrations. The demonstration data usually are a sequence of state–action pairs within the task constraints. Here, the learning system can learn a direct mapping that outputs actions given states or learn a task plan mapping the sequence of actions that lead from an initial state to a goal state [26];
- Reward learning is similar to policy learning but, instead of using a set of actions to represent a task, the task is modeled in terms of its goals or objective, often referred to as a reward function. Techniques such as Inverse Reinforcement Learning (IRL) [38] enable the learning system to derive explicit reward functions from demonstration data in this method;
- Semantic learning consists of extracting the correct parameters that compose a task through the interpretation of semantic information present in the demonstrations. These semantics can be classified into task features, object affordances, and task frames [25,26,29].

#### 3.1.4. Skill or Task Execution

In this stage, the evaluation of the learning takes place. Notice that the final goal of LfD is the ability to solve different complex tasks with all the skills encoded (learned) during the demonstrations phase. Ideally, the knowledge acquired should be enough to adapt the functionality of the robot to different dynamic environments with little to none human intervention. Moreover, the learned skills should be able to generalize through similar tasks autonomously.

#### 3.1.5. Refinement Learning

In LfD, normally the acquisition stage and the modeling stage are independent during the learning process, this means that if there is a desire to change some parameters of the learned task, it will be necessary to record new demonstrations in order to learn a new model, which is an offline process. However, in the LfD process, it is equally important the way the demonstrator performs the demonstration of the desired behavior as how the robot (learner) is capable of efficiently demonstrating and communicating the actual

level of knowledge acquired to the end-user during task execution [25,39]. Moreover, with some modeling techniques, many demonstrations are needed to obtain a statistically valid inference during the modeling process, it is unrealistic to expect that a single demonstrator provides a large set of them, one would like the robot to have some initial knowledge to immediately perform the desired task, and ideally, the demonstrations provided would be used only to help the robot to gradually improve its performance [24].

This concept is commonly known as Incremental Learning (IL), which involves techniques that allow the demonstrator to teach new data while preserving previously demonstrated data [40]; in the LfD context, this could be performed in an online manner [41–43]. In the literature, there exist different approaches, the most common ones are based on Active Learning (AL), aimed at domains in which a large quantity of data is available, but labeling is expensive and enables a learner to query the demonstrator and obtain labels for unlabeled training examples [30,38,39]; and Reinforcement Learning (RL), used to refine an existing policy without human involvement by learning directly from environmental reward and exploration [44–46].

#### 4. Analysis

In this section, the selected papers obtained from our methodology are analyzed in order to answer to the research questions stated in Section 1. In the previous section, Learning from Demonstrations (LfD) was introduced as a five-stage process. This was performed in order to better present and analyze the contents of each paper according with two scenarios: the *Human participation during the LfD process* (Data Acquisition, Demonstrator Selection and Refinement Learning stages), and the *Main LfD algorithms for Skill/Task learning* (Data Modeling and Skill/Task execution stages) for collaborative robot applications. Next, the analysis of these scenarios is presented.

##### 4.1. Human Participation in the LfD Process

In the LfD process, the human plays the inherently important role as a demonstrator (teacher), to provide the sequence of actions that a robot should execute during operation. However, sometimes, the fact that the human also acts as an evaluator during the execution of the learned behavior is overlooked. As an evaluator, the human can provide with feedback to the robot (student) during the execution of the learned movements to refine and subsequently improve them over time. Accordingly, in this scenario, the papers which emphasize the human role not only as a demonstrator, but also as evaluator, or both were added.

Classifying the contribution of the demonstrator is important to understand the impact of the teacher role during the learning process and to identify possible weaknesses in the method as well as research gaps that are commonly overlooked during the LfD process. In most papers, little information is provided of the characteristics of the demonstrator during the teaching process. For the purpose of giving better insights through our classification scheme, when the information about the demonstrator is scarce or not explicitly indicated by the authors, the demonstrator is labeled as not specified, otherwise, the label of non-expert was used, when the demonstrations were performed by first-time or novice users for validation purposes, or multiple expertise level, when the demonstrations were performed by several users with different prior knowledge of how to program robots, is assigned according to the validation procedures described in each paper.

One problem with allowing the robot to only learn from demonstrations is the biased idea of considering that the human is always correct. If the process of teaching is deficient, the learning will also be deficient (sub-optimal). For this reason, it is important to allow the robot to interact directly with the environment—known as Reinforcement Learning methods—to gain some insight into how behaviors (movements) not seen during the demonstrations affect the completion of the task and to develop knowledge on how the task could be improved over time automatically.

### Data Acquisition Methods

For the analysis, the Data Acquisition methods were grouped in four different scenarios: Physical, Observable, Reactive, and Multimodal guidance as it is shown in Table 2, to better understand how recent literature deals with the correspondence problem during the demonstration phase.

**Table 2.** Classification of the acquisition methods found in literature.

Classification	Method	Used in
Physical guidance	Kinaesthetic Teaching Teleoperation & Haptics Shadowing	[27,47–65] [66] [67]
Observable guidance	Vision system IMU	[28,68–75] [76]
Reactive guidance	NLP GUI	[46] [77]
Multimodal guidance	Mixture of methods	[31,78–85]

It was found that, to solve the correspondence problem in LfD, the majority of papers use direct mapping methods such as Kinaesthetic teaching as the principal technique for Data Acquisition. However, these kinds of methods are not always a straightforward to use because of the difficulties involved in directly moving a manipulator with multiple degrees of freedom with precision and safety [26]. Some authors have focused their efforts on developing new data acquisition systems for demonstrations, to improve their performance and usability.

In Table 3, we present state-of-the-art acquisition methods developed in recent years focused on simplifying the demonstration process for the teacher based on methods different than those clustered as Physical guidance. Zhang et al. [73] proposed a pen-like device which is tracked with computer vision to simplify the process of teaching a trajectory to a robot, the system uses RGB data to track the position of the teaching device, which is used to generate the path that the robot must follow. While the idea of tracking markers to teach a desired movement is not new, the work focuses on using the minimum required components to work, reducing complexity and costs. The precision of the system was validated on a UR5 collaborative robot from Universal Robots comparing their method with direct manipulation with promising results. Lee et al. [82] present a novel teleoperation control system based on a vision system and Digital Twin (DT) technology. The system captures the desired pose to be taught in a 2D image and then uses a human DT to transfer it to a collaborative robot DT. The relevant aspect of this work is how it solves the correspondence problem using Bezier curve-based similarity to obtain the rotation axis and rotation angle of each robot arm joint based on the arm pose of the human model.

**Table 3.** Novel acquisition methods found in recent literature.

Year	Author	Classification	Method	Robot
2020	Zhang et al. [73]	Observable	Marker tracking	UR5
2021	Soares et al. [81]	Multimodal	Augmented Reality	UR5/ABB IRB2600
2021	Rodriguez et al. [76]	Observable	IMU	Kuka KR 16
2022	Lee et al. [82]	Multimodal	Vision System & Digital Twin	ABB YuMi

Soares et al. [81] implemented a VR system where the user can teach a desired trajectory in its own configuration space through hand movements. The hand positions are recorded from an Inertial Measurement Unit (IMU) embedded in a Microsoft HoloLens 2, while the operator can visualize the desired trajectory through the head-mounted system of the device. This allows an operator without programming experience to program a robot by

demonstration, still more work needs to be completed towards increasing the precision of the trajectories being taught according to the authors. Another work where an IMU-based device was developed to teach trajectories to a robot was presented in [76], where the authors reported that the advantages are the low cost and the portability of the system due to its size, which allows its use in unstructured environments; also, the device is capable of detecting static states in human movements accurately filtering small movements inherent to a human arm through a novel fusion algorithm based on an Adaptive Kalman Filter.

#### 4.2. Main LfD Algorithms for Skill/Task Learning

Data Modeling is the stage where the robot uses the demonstrated data as a base to learn a specific skill or how to solve a certain task. For the sake of clarity, in this work, we consider a skill as a primitive movement that allows the robot to move freely in the environment, while a task is defined as a mixture of these primitive movements used to solve specific industrial problems. Therefore, this learning process is classified into low-level motion (skills) and high-level problem solving (tasks). Hybrid methods are considered to be the combination of skill and tasks learning, which are used to provide a more general solution to certain problems. For this survey, the articles using hybrid methods were identified and emphasized in Tables 4–6 with a star symbol (\*).

**Table 4.** Works focused on developing robot skills using deterministic approaches.

Year	Author	Modeling Method	Encoding of	Skill Learned
2017	* Liang et al. [62]	DMP	Motion	Pick & Place
2018	Peternel et al. [78]	DMP	Motion and Force	Visual servoing
2018	Ghalamzan et al. [57]	DMP and GMM-GMR	Motion	Obstacle avoidance
2019	* Schlette et al. [54]	DMP	Motion and Force	Obstacle avoidance
2021	Iturrate et al. [60]	DMP	Motion and Force	Gluing
2021	Wang et al. [67]	DMP	Motion and Force	Sweeping
2022	Wu et al. [69]	DMP-GP	Motion and Force	Hand over
2022	* Liang et al. [65]	Keyframe-based	Motion	Various*

\* Articles using a Hybrid (Low-level and High-level) learning method.

**Table 5.** Works focused in developing robot skills using probabilistic methods.

Year	Author	Modelling Method	Encoding of	Skill Learned
2016	Koskinopolou et al. [27]	GLPVM and PCA	Motion	Picking and Pushing
2016	Rozo et al. [58]	ADHSMM	Motion	Hand over
2016	Rozo et al. [59]	TP-GMM	Motion and Force	Lifting
2016	Tang et al. [85]	GMM-GMR	Motion and Force	Peg-in-hole
2017	Castelli et al. [72]	GMM-GMR	Motion	Visual servoing
2018	Raiola et al. [56]	GMM-GMR	Motion and Force	Virtual guidance pick and place
2018	* Huang et al. [71]	GMM-GMR	Motion	Sewing
2019	Qu et al. [28]	PCA and GMM-GMR	Motion	Dual arm coordination
2019	* Kyrarini et al. [55]	GMM-GMR	Motion	Object manipulation
2019	Zeng et al. [80]	HSMM-GMR	Motion and Force	Pushing
2020	DeConinck et al. [52]	CNN	Motion	Grasping
2020	* Koert et al. [83]	ProMP	Motion	Hand over
2021	Al-Yacoub et al. [79]	WRF	Motion and Force	Grasping
2021	Wang et al. [61]	GMM-GMR	Motion	Pick & Place
2021	* Hu et al. [50]	GMM-GMR	Motion	Pick and place
2021	Fu et al. [64]	MTProMP/MTiProMP	Motion	Multi-tasking
2021	Zaatari et al. [74]	TPGMM-TPGMR	Motion	Various*
2021	Zaatari et al. [75]	TPGMM-TPGMR	Motion	Various*
2021	Wang et al. [51]	GMM-GMR	Motion	Pick
2022	* Qian et al. [84]	EIProMP	Motion	Hand over
2022	Zhang et al. [49]	GMM-GMR	Motion	Grasping

Table 5. Cont.

Year	Author	Modeling Method	Encoding of	Skill Learned
2022	Hu et al. [48]	TPGMM-TPGMR	Motion and Force	Grasping
2022	Lai et al. [47]	pHRIP	Motion and Force	Target Reaching

\* Articles using a Hybrid (Low-level and High-level) learning method.

Table 6. Works focused on learning task completion.

Year	Author	Modeling	Task
2017	Haage et al. [68]	Semantic Learning	Assembly
2017	* Liang et al. [62]	Policy Learning	Assembly
2018	* Huang et al. [71]	Semantic Learning	Assembly
2018	Schou et al. [66]	Semantic Learning	Various
2019	Winter et al. [46]	Policy Learning	Assembly
2019	Steinmetz et al. [53]	Semantic Learning	Various
2019	Wang et al. [31]	Reward Learning	Assembly
2019	Ramirez-Amaro et al. [63]	Semantic Learning	Sorting
2019	* Schlette et al. [54]	Policy Learning	Assembly
2020	* Koert et al. [83]	Semantic Learning	Assembly
2020	Racca et al. [77]	Semantic Learning	Hand-over
2022	Sun et al. [70]	Semantic Learning	Assembly
2019	* Kyrarini et al. [55]	Policy Learning	Assembly
2022	* Hu et al. [48]	Policy Learning	Assembly
2022	* Liang et al. [65]	Policy Learning	Assembly
2022	* Qian et al. [84]	Semantic Learning	Assembly

\* Articles using a Hybrid (Low-level and High-level) learning method.

#### 4.2.1. Skill Learning

As mentioned in Section 3, the learning of low-level motions consists of teaching the robot individual movements, which include trajectories and interaction forces, from the demonstration. The most common learning methods are classified in deterministic and probabilistic approaches. Dynamic Movement Primitives (DMPs) is a nonlinear method used to generate multidimensional trajectories by the use of non-linear differential equations [86]. In Table 4, the literature based on deterministic methods for skill learning is presented. Here, it is possible to notice that DMPs is the predominant deterministic method used in the current literature. The method is considered to be deterministic because it only requires a single demonstration to encode the initial and goal states of a trajectory.

Simultaneous control of force and position for teaching and execution is physically impossible. To tackle this problem, Iturrate et al. [60] proposed an adaptive controller architecture that can learn continuous kinematic and dynamic task constraint from a single demonstration. The task kinematics were encoded as DMPs, while the dynamics of the task were preserved by encoding the output of the normal estimator during demonstration as Radial Basis Functions (RBFs) synchronized with the DMPs. The demonstration is performed by the user using Kinaesthetic teaching enhanced with a variable-gain admittance control where the damping is continuously adapted based on a velocity–force rule in order to match the user’s intention and deduce the physical effort needed during teaching. The idea is to solve the gluing of electronic components to a PCB; their approach was evaluated on a 3D-printed replica of a PCB with the attached components.

Similarly, Wang et al. [67] proposed a scheme where motion trajectories and contact force profiles are modeled simultaneously using DMPs to provide a comprehensive representation of human motor skills while the system complexity was kept low through the implementation of a momentum-based force observer, which allows the reproduction of the motion and force profiles without involving an additional force sensor. The demonstrations are performed using shadowing, which is a type of kineasthetic teaching in which instead of directly moving the robot, the actions to be taught are performed in an exact replica of the same robot in a master–slave control fashion. This was performed to avoid affecting

the performance of the force observer. The learned skill consists of cleaning a table through a push action of small parts of the table similar to a sweeping movement.

Peternel et al. [78] also combined DMPs with a hybrid force/impedance controller to govern the robot motion and force behavior but in a human–robot co-manipulation environment. This method enables the robot to adapt and modulate the delivered physical assistance as a function of human fatigue. The robot begins as a follower and imitates the human leader to facilitate the cooperative task execution while gradually learning the task parameters from the human. When the robot detects human fatigue, it takes control of the physically demanding aspects of the task while the human keeps control of the cognitively complex aspects. Additionally, the human fatigue is measured with electromyography (EMG) and modeled through a novel approach developed by the authors which was inspired by the dynamical response of an RC circuit. The proposed framework was tested on two collaborative skills, wood sawing, and surface polishing.

In contrast, Ghalamzan et al. [57] propose a combination of deterministic and probabilistic techniques—DMP and GMM/GMR—to dynamically adapt the robot to environments with moving obstacles. In this work, they use the single-shot approach (DMP) to learn the desired motion trajectories in a pick-and-place task, this allows the robot to learn the task model and its constraints, assuming that the initial and the final location of the object to be manipulated are invariant with respect to time. Then, to include the constraints imposed by the environment, they apply multi-shot learning (GMM/GMR) to learn how to avoid static obstacles without deviating from the initial and final goals from the task using multiple demonstrations and obstacles positions. Finally, by generalizing the utility function developed for the task in a static environment, they can approach dynamic environments by using a Model Predictive Control (MPC) strategy. The object and obstacle detection is performed through computer vision, while the demonstrations of the desired task are taught using Kineasthetic teaching.

Liang et al. [62] propose a framework that allows non-expert users to program robots using symbolic planning representations by teaching low-level skills. The framework uses DMPs to teach these atomic actions through demonstrations because DMPs generate trajectories based on the initial and final state of the demonstrated trajectory, this allows to generalize a demonstrated trajectory under the assumption that the learned trajectory is independent of the trajectory performed by the user. The framework also uses a visual perception system to recognize object and certain environment properties to learn high-level conditions to encode symbolic action models. Schlette et al. [54] also exploit the capabilities of the DMPs that allow the reduction of the recorded trajectories to a few parameters that can be modulated to adapt to new situations during execution arising as a result of time independence and encoding of forces and torques in the desired trajectories. Both works were developed in semi-structured environments and the main goal of teaching low-level skills is to use them to solve more complex tasks. In [62], an automated planner generates an action sequence to achieve the task goal, one of its main contributions is a retroactive loop that allows the user to refine the learned skills in case of ambiguities. The usability and simplicity of the framework were tested with the help of multiple users with favorable results. In [54], a development was presented in a robotic assembly cell during the World Robot Challenge 2018 (WRC 2018) where operations such as pick and place, handling and screwing were evaluated.

In DMP formalism, a single model can be used to generate multiple trajectories using its spatial scaling property, but there are several drawbacks to this property in the original formulation [87]. For example, if the scaling term is zero, or near to zero, the non-linear term cannot be learned or the trajectory generated could be very different from the original trajectory. Moreover, the non-linear forcing term usually is written in terms of Gaussian basis functions with different weights and the learning of the DMP model consists in determining each one of these weights. An issue, mentioned in Wu et al. [69], within the parameters that constitute each of these basis functions is the need to be manually adjusted. The researchers propose the formulation of the non-linear forcing term in DMPs as a

Gaussian Process (GP) to automatically optimize the hyper-parameters applying a sparse GP regression, and the spatial scaling is determined by a rotation matrix to solve the problem when the initial and the final positions are identical in some directions. This new formulation is used to teach a handover prediction skill to a robot where the main issue is that the handover location is unknown before the movement starts. Despite this, the formulation presented is capable of making online corrections based on human feedback using a gradient-type update law.

The main advantage of deterministic methods applied to LfD such as DMPs is that a trajectory can be learned from a single demonstration. Thus, the acquisition problem can be simplified, but in face of suboptimal or erroneous demonstrations or in cases where more than one teacher is required, this also means a great disadvantage because the reproduction step will lead to undesired behaviors; additionally, DMPs approaches model variables separately ignoring possible correlation between multiple variables, in terms of robot LfD this represents a drawback in systems where the relationship between motion and force is crucial. Probabilistic methods offer a solution to this kind of scenarios by incorporating variance and other stochastic metrics to the demonstrations in exchange of requiring multiple demonstrations during the learning process. Table 5 enlists the works found in recent literature based probabilistic methods for skill learning.

A solution based on Movement Primitives is the Probabilistic Movement Primitives (ProMP) framework introduced in [88] and extended in [89]. The framework couples the joints of a robot using the covariance between trajectories of several degrees of freedom and regulates the movement by controlling the target velocities and positions in terms of a probability distribution. This method has been the precursor to other techniques such as Interaction ProMP, fully described in [90], which models the correlation between movements of two different agents from joint demonstrated trajectories. Koert et al. in [83,91] proposed a framework to increasingly build a library of robot skills demonstrated by a human teacher called Mixture of Interaction ProMPs, which is an extension to Interaction ProMPs that allows to learn several different interaction patterns by applying Gaussian Mixture Models (GMM) to unlabeled demonstrations; it implements online learning of cooperative tasks by allowing to continuously integrate new training data by the means of new demonstrations. The demonstrations are recorded by Kineasthetic teaching for the robot and motion capture for the human trajectories, the framework was evaluated with the help of ten human subjects.

Fu et al. [64] also combine ProMP and Interactive ProMP with GMM, but the focus is to solve the problem of multi-tasking in robots that solves multiple tasks (MTProMP), and multi-tasking in human–robot collaboration (MTiProMP). The proposed solution was tested on a Baxter robot in passing a set of via-points with MTProMP, but only one joint was considered. The same conditions are applied for testing the collaborative task, but in this case, a change in hand position represents the change in the behavioral intention of the user. Qian et al. [84] extend the formulation behind Interactive ProMP to additionally incorporate immediate environment information, not only the observations of the human actor, but information about obstacles and other environmental changes are considered to proactively assist a human in collaborative tasks. The method was named Environment-adaptive Interactive ProMP (EIProMP) and the key to the model resides in assuming a regression relation between environmental parameters, such as the height and/or width of an obstacle, and a set of weight vectors obtained from the demonstration trajectories. This allows to encode an environment related weight vector to successfully add the environment parameters to the Interactive ProMP formulation. The taught skill was a handover skill for assisting in a push-button assembly task. Another probabilistic method based on DMP is the physical Human–Robot Interaction Primitives (pHRIP) proposed by Lai et al. [47]. The method of pHRIP extends the Interaction Primitives (IP) introduced by Ben Amor et al. [92] by integrating physical interaction forces. In the original IP formulation, a probabilistic approach and modeling of a distribution  $p(\theta)$  over the parameters of the DMPs is used to predict future behaviors of the agent given a partial observed trajectory

and to correlate its movements to those of an observed agent. In pHRIP, the parameter distribution  $p(\theta)$  integrates the observed interaction forces in coupled human–robot dyads and by obtaining the predictive distribution of the DMP parameter set using the partial observations of the phase-aligned interactions, the robot is capable of matching its response to the user’s intent. A planar and Cartesian target reaching task was used to validate the efficacy of the proposed method with different users and the results show that with a small number of observations, the pHRIP can accurately infer the user intent and also adapt the robot to novel situations.

Similarly, in [58], a handover skill was evaluated through an approach for collaborative robots to learn reactive and proactive behaviors from human demonstrations to solve a collaborative task. The approach is based on an Adaptive Duration Hidden Semi-Markov Model (ADHSMM) to allow the robot to react to different human dynamics, improving the collaboration with different users. In this approach, the duration of each state allows the robot to take the lead of a task when it is appropriate—namely, according to the task dynamics previously experienced in the demonstrations—which can be exploited to communicate its intention to the user.

Another work where a Hidden Semi-Markov Model (HSMM) is further used can be found in [80] where a joint force controlled primitive is learned from multiple sensors to solve a pushing motion (over a stapler and soap dispenser). The approach is based on multi-modal information obtained from Kinaesthetic teaching, EMG signals measured from human, and a force sensor mounted between the flange and the end-effector of the robot. HSMM is used to model the demonstration data, to improve the robustness of the system against perturbations, and to model the distribution between position and stiffness. Finally, Gaussian Mixture Regression (GMR), which is a non-parametric regression approach used on functional data to help us in the modeling and prediction of a random variable [36], is applied to generate the control variables for an impedance controller.

One of the most popular probabilistic methods applied to LfD of skill learning is the combination of GMM with GMR which extends the HMMs for action modeling, it was introduced by Calinon et al. [93] and further extended in their later work [94]. The method was applied in multiple works to learn some assembly-related skills [49,51,55].

In literature, other techniques can be found focused on probabilistic methods, as demonstrated by Koskinopolou, et al. [27], where the data acquisition of the demonstrated task was performed through markers on human body tracked with computer vision and an RGB-D camera, mapped the demonstrated data to the robot configuration space using Gaussian Process Latent Variable Model (GPLVM), which is a non-linear generalization of Principal Component Analysis (PCA), and it provided a probabilistic compact transformation of a given high-dimensional dataset to a low-dimensional one. Another approach based on Principal Component Analysis (PCA) can be found in Qu et al. [28] for control of coordination in redundant dual-arm robots. PCA as a dimensionality reduction technique is used for removing uncorrelated data from human-arm demonstrations, then GMM are applied in order to extract human-like coordination characteristics by the Gaussian components, which are later generalized and reproduced through GMR. To finally obtain a human-like coordination motion equation for a dual-arm robot. The system was tested on carrying and a pouring task, the results show that more natural and smooth motions can be achieved in dual-arm robots through this method.

To address non-sequential HRC applications, Al-Yacoub et al. [79] proposed to implement Weighted Random Forests (WRF) as a regression strategy to encode spatial data (such as force) in haptic applications, because this type of regression is robust against overfitting, allowing better generalization in varying conditions (e.g., different users and obstacles). In conventional Random Forest (RF) method, many decisions trees create an uncorrelated forest of trees where the final prediction is more accurate due to the contribution of each individual tree. The idea of weighting each tree based on their performance is to improve the overall performance of the algorithm on unseen test data, the solution proposed is a stochastic weighting approach where the trees are weighted according to

their Root Mean Square Error (RMSE). The algorithm was tested on a co-manipulation skill for moving heavy objects between two points and a co-assembly task that involves a peg-in-hole assembly with demonstrations that involve human–human collaboration and human–robot collaboration scenarios comparing the RF approach to the proposed WRF method. The results showed that WRF improves performance over RF models because of the lower interaction forces and shorter execution times, also, the demonstrations based on human–human collaboration results in more human-like behavior and in less input force required.

De Coninck et al. [52] developed a LfD system using Convolutional Neural Networks (CNNs) for robot grasping. The approach was called GraspNet architecture and it outputs an energy heat map which labels how well a region on the image frame works as the best grasp location and estimates the rotation angle of the desired object, using a single camera mounted on the wrist of the manipulator. Yet, some assumptions made during the experiments should be removed in the future for the sake of a better generalization.

#### 4.2.2. Task Learning

On the other hand, high-level tasks are composed of multiple individual actions (low-level motions) to solve more complex task. We clustered the modeling techniques found in the analyzed papers based on the three classifications presented in Section 3, which are Policy, Reward, and Semantic Learning. The objective of these modeling techniques is to learn a sequence of steps (actions) to follow given a particular state to solve the task (Policy Learning), learn what are the goals of a particular task and solve them based on this information (Reward Learning), or extract the most prominent characteristics of the task to solve it even if the environment or certain conditions change in the future (Semantic Learning).

In Table 6, we present the papers focused only on high-level tasks and the techniques used for solving them. In the context of Semantic Learning, in Haage et al. [68] the semantic representation of an assembly task is achieved through 2D and 3D visual information extraction from the demonstrations, and the semantic analysis of manipulation actions is performed using a graph-based approach. Finally, the task generation and execution is supported by a semantic robotic framework called Knowledge Integration Framework (KIF) to increase robustness and semantic compatibility. Schou et al. [66] developed a software tool called Skill-Based System (SBS) for sequencing and parameterization of skills and tasks. Based on a predefined skill library which can be extended if needed, the operator can program a task selecting the sequence of skills desired and then kinaesthetically teach the parameters of the skill. This approach makes the robotic programming more intuitive and simple for non-expert users in industrial scenarios. Similarly, Steinmetz et al. [53] use a Planning Domain Definition Language (PDDL) to parameterize user-defined skills in order to simplify the robot programming task to the user, while in Ramirez-Amaro et al. [63], to learn the correlation between the robot movements and the environment information, a decision tree classifier is used to interpret the demonstrated activities, the obtained semantic representations are robust and invariant to different demonstration styles of the same activity. In Sun et al. [70], the semantic extraction is performed through a dual-input Convolutional Neural Network (CNN) that incorporates not only the camera context (objects), but also the task context to collaboratively solve an assembly task.

An example of Policy Learning can be found in Winter et al. [46], their proposed method is called Interactive Reinforcement Learning and Potential-Based Reward Shaping (IRL-PBRS), used to solve an assembly task. The method is based on Hierarchical Reinforcement Learning, but the main objective is that the human transfers their knowledge in the form of advice to speed up the RL learning agent. The idea is that the robot should find a valid assembly plan based on the information provided by the human through Natural Language Processing (NLP). The method is capable of learning quickly to keep up with the user-given advice and to adapt to changes in the given demonstrations. On the other hand, Wang et al. [29] solve a customizable assembly with a method called Maximum Entropy

Inverse Reinforcement Learning (MaxEnt-IRL). The idea behind this method is to calculate a reward function (Reward Learning) by learning from the user demonstrations to generate an optimal assembly policy according to the human instructions.

It is worth mentioning that most of the solutions described in this section are based on learning motion trajectories, but the interaction forces of the skill or task are often omitted. This leads to poor or underperforming skill or task reproduction, as demonstrated in [80], where not only the robot forces are modeled, but also the human muscle activation; the results show that the robot is able to better generalize and smoothly reproduce the skill when the interaction forces seen during the demonstrations are modeled. Similar achievements are presented in [67,85] for pushing-like skills such as sweeping and peg-in-hole assemblies, the robot execution was proven to be better compared to only following the motion trajectory. Another example can be found in [60], where the force modeling consideration not only produces better execution, but also helps during the demonstration phase to provide a better user experience during Kineasthetic teaching. Skill/task interaction force modeling is far more important in collaborative interactions between human and robots. This is because it allows a more ergonomic and simple co-manipulation [56,78], as well as safe and human-like interactions between human and robot [47,59,79].

## 5. Discussion and Conclusions

The literature analyzed above shows the improvements made towards LfD in collaborative robotics which can be applied to some industrial tasks, however, there are issues that are worth mentioning. This section summarizes the open problems found in teaching collaborative robots and how to work alongside its human partner to solve collaborative tasks.

Most of the reviewed literature used testing scenarios where only a handful of demonstrators can teach to the robot the desired behaviour; additionally, the majority of the reviewed works assume an expert demonstrator during the demonstration stage to simplify the whole LfD process. While this could be true in terms of task knowledge, this assumption is far from reality and, thus, not directly applicable to real scenarios. In real scenarios, even if an operator has full knowledge of the task to solve, there is always a gap in experience between multiple operators working on the same task. This difference in experience directly affects the quality and time spent during the demonstrations. How the learning algorithms can cope with this kind of variability in user experience is yet to be understood. For example, applying Incremental learning to LfD solutions could allow to teach an Skill/Task, regardless of the operator experience, and refine it later according to the requirements of the implementation [43,46,77].

Moreover, even if the demonstrator is an expert in solving the problem, this does not ensure flawless demonstrations during the LfD process. There are factors related to the intuitiveness, ergonomics, and safety of the Human–Robot interaction that hinder the demonstrator’s capability to teach the desired behavior to the robot during demonstrations. For example, a vast majority of works still rely on using Kinaesthetic teaching for the demonstrations. While this kind of technique reduces drastically the ill-posed correspondence problem, from an ergonomics and intuitiveness perspective, this method also imposes physical and mental constraints to the demonstrator. Carfi et al. [95] demonstrated the difficulties that an operator has while trying to open or close a gripper and position a robotic arm at the same time in a pick-and-place operation. A comparison between human–robot demonstration and human–human demonstration shows that it is easier and less stressful to teach a learner who has an active role during demonstrations.

Other solutions worth exploring are those that integrate new emerging technologies such as Augmented Reality (AR) [96], Virtual Reality (VR) [97], or Digital Twins (DT) [98], to list some. These technologies increase the safety of the testing environments [99,100], and decrease the ergonomic load perceived by the user, to encourage more natural human–robot interactions [101–103]. Particularly, the use of DT in industrial applications show promising results in increased safety and interaction [104–106], but also in reduced im-

plementation time and costs [107]. Nevertheless, more studies focused on human–robot interactions with different demonstrators participation and studies incorporating the aforementioned technologies in the LfD process are needed.

The idea of learning and generalizing trajectories capable of avoiding obstacles is essential. It is undeniable that research should consider the effect of forces acting on these trajectories. The challenge relies on communicating the direction and magnitude of the necessary force during the task/skill execution. Additionally, there are difficulties in sensing the dynamics of external elements, for both robot and human. How is the demonstrator able to teach the necessary force for a task to be completed. How can the robot receive this information correctly? In Kineasthetic teaching, for example, the forces measured by the robot will not be task forces only, but also those applied by the human demonstrator. As correctly pointed out in [60], these two sets of forces will not always be aligned; thus, using them to learn task constraints will result in an alignment error in the constraint frame. Decoupling the human force from the force perceived by the robot is one possibility proposed in [60,67,80] for partial answering of the above questions. For example, in [67], a shadowing guidance technique is used during the demonstration to completely isolate the task interaction forces captured from the sensor on a Baxter robot during demonstrations. Iturrate et al. [60] decouple the forces through their admittance controller, and Zeng et al. [80] measure the human stiffness separately with an EMG sensor. Still, more similar research is needed on ways to learn trajectories and the interaction applied forces to a skill or task for future adoption and implementation of LfD in the manufacturing industry.

Different definitions of cooperative and collaborative tasks can be found in the literature [11,108]. The ultimate goal of using LfD on collaborative robots is the capability of combining the natural abilities of the end user with the power, speed, and repeatability of the robots. In this context, research should focus on defining and solving scenarios where human and robot are capable of interacting for the sake of solving the complexities of the task at hand. How to perform these demonstrations is also a challenge, for example, how many demonstrators should be considered in the design when teaching a joint task? Which interfaces are optimal for teaching a joint task? What is better for performance—learning individually the skills and later using them in the joint task or learning the necessary joint skills from the beginning? There is a need for more experiments to help answer these questions.

Ideally, the robot will learn how to perform the desired task at hand after been taught once; but in real applications, this is extremely complicated, so each LfD process should consider how an user could modify the learned skills or their application to certain tasks depending on how the dynamics of the task at hand change. Take, for example, an assembly task where some specific requirements dictate that the piece should be welded, but in other cases, the use of screws is the principal requirement. How can the end user easily increase the skill library of the robot to expand the tasks that it can solve through time without the need of being taught again? At which point should the human intervene, and should the learning algorithm of the robot be intelligent enough to adapt or develop a new skill necessary for completing its work?

Throughout this work, topics referring to learning from anyone, learning forces, refinement of the learning at design, and truly collaborative tasks were tackled and summarized. The purpose of this work is to provide a review of techniques and possible future research topics.

**Author Contributions:** Conceptualization, A.D.S.-C., H.G.G.-H. and J.A.R.-A.; methodology, A.D.S.-C., H.G.G.-H. and J.A.R.-A.; software, A.D.S.-C.; validation, A.D.S.-C., H.G.G.-H. and J.A.R.-A.; formal analysis, A.D.S.-C., H.G.G.-H. and J.A.R.-A.; investigation, A.D.S.-C., H.G.G.-H. and J.A.R.-A.; resources, A.D.S.-C., H.G.G.-H. and J.A.R.-A.; data curation, A.D.S.-C., H.G.G.-H. and J.A.R.-A.; writing—original draft preparation, A.D.S.-C.; writing—review and editing, A.D.S.-C., H.G.G.-H. and J.A.R.-A.; visualization, A.D.S.-C., H.G.G.-H. and J.A.R.-A.; supervision, H.G.G.-H. and J.A.R.-A.; project administration, H.G.G.-H. and J.A.R.-A.; funding acquisition, H.G.G.-H. and J.A.R.-A. All authors have read and agreed to the published version of the manuscript.

**Funding:** The APC was funded by Tecnológico de Monterrey.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** This work was partially supported by scholarship grant number 646996 of Consejo Nacional de Ciencia y Tecnología (CONACYT), Mexico.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

AL	Active Learning
ADHSMM	Adaptive Duration Hidden Semi-Markov Model
AR	Augmented Reality
CNNs	Convolutional Neural Networks
DT	Digital Twin
DMPs	Dynamic Movement Primitives
EMG	Electromyography
EIProMPs	Environment adaptive Interactive Probabilistic Movement Primitives
GMM	Gaussian Mixture Models
GMR	Gaussian Mixture Regression
GP	Gaussian Process
GPLVM	Gaussian Process Latent Variable Model
HMMs	Hidden Markov Models
HSMM	Hidden Semi-Markov Model
HRC	Human-Robot Collaboration
IL	Incremental Learning
IMU	Inertial Measurement Unit
IP	Interaction Primitives
IRL	Inverse Reinforcement Learning
IRL-PBRS	Interactive Reinforcement Learning and Potential Based Reward Shaping
KIF	Knowledge Integration Framework
LfD	Learning from Demonstration
MaxEnt-IRL	Maximum Entropy Inverse Reinforcement Learning
MPC	Model predictive Control
MTiProMP	Multi-Task interaction Probabilistic Movement Primitive
MTPRoMP	Multi-Task Probabilistic Movement Primitive
NLP	Natural Language Processing
DSL	Domain Specific Language
PbD	Programming by Demonstration
PCA	Principal Component Analysis
pHRIP	physical Human Robot Interaction Primitives
PDDL	Planning Domain Definition Language
ProMPs	Probabilistic Movement Primitives
RBFs	Radial Basis Functions
RF	Random Forest
RL	Reinforcement Learning
RMSE	Root-Mean Square Error
SBS	Skill Based System
SLR	Systematic Literature Review
TPGMM	Task Parametrized Gaussian Mixture Model
VR	Virtual Reality
WoS	Web of Science
WRFs	Weighted Random Forests
WRC 2018	World Robot Challenge

## References

1. Alcácer, V.; Cruz-Machado, V. Scanning the industry 4.0: A literature review on technologies for manufacturing systems. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 899–919. [[CrossRef](#)]
2. Dotoli, M.; Fay, A.; Miśkiewicz, M.; Seatzu, C. An overview of current technologies and emerging trends in factory automation. *Int. J. Prod. Res.* **2019**, *57*, 5047–5067. [[CrossRef](#)]
3. Mittal, S.; Khan, M.A.; Romero, D.; Wuest, T. Smart manufacturing: Characteristics, technologies and enabling factors. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2019**, *233*, 1342–1361. [[CrossRef](#)]
4. Bauer, A.; Wollherr, D.; Buss, M. Human–robot collaboration: A survey. *Int. J. Humanoid Robot.* **2008**, *5*, 47–66. [[CrossRef](#)]
5. Evjemo, L.D.; Gjerstad, T.; Grøtli, E.I.; Sziebig, G. Trends in smart manufacturing: Role of humans and industrial robots in smart factories. *Curr. Robot. Rep.* **2020**, *1*, 35–41. [[CrossRef](#)]
6. Villani, V.; Pini, F.; Leali, F.; Secchi, C. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics* **2018**, *55*, 248–266. [[CrossRef](#)]
7. Bi, Z.M.; Luo, M.; Miao, Z.; Zhang, B.; Zhang, W.J.; Wang, L. Safety assurance mechanisms of collaborative robotic systems in manufacturing. *Robot. Comput.-Integr. Manuf.* **2021**, *67*, 102022. [[CrossRef](#)]
8. Maurice, P.; Padois, V.; Measson, Y.; Bidaud, P. Human-oriented design of collaborative robots. *Int. J. Ind. Ergon.* **2016**, *57*, 88–102. [[CrossRef](#)]
9. Gualtieri, L.; Rauch, E.; Vidoni, R. Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review. *Robot. Comput.-Integr. Manuf.* **2021**, *67*, 101998. [[CrossRef](#)]
10. Hentout, A.; Aouache, M.; Maoudj, A.; Akli, I. Human–robot interaction in industrial collaborative robotics: A literature review of the decade 2008–2017. *Adv. Robot.* **2019**, *33*, 764–799. [[CrossRef](#)]
11. Zaatari, S.E.; Marei, M.; Li, W.; Usman, Z. Cobot programming for collaborative industrial tasks: An overview. *Robot. Auton. Syst.* **2019**, *116*, 162–180. [[CrossRef](#)]
12. Michaelis, J.E.; Siebert-Evenstone, A.; Shaffer, D.W.; Mutlu, B. Collaborative or Simply Uncaged? Understanding Human-Cobot Interactions in Automation. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 25–30 April 2020; pp. 1–12. [[CrossRef](#)]
13. Argall, B.D.; Chernova, S.; Veloso, M.; Browning, B. A survey of robot learning from demonstration. *Robot. Auton. Syst.* **2009**, *57*, 469–483. [[CrossRef](#)]
14. Hussein, A.; Gaber, M.M.; Elyan, E.; Jayne, C. Imitation learning: A survey of learning methods. *ACM Comput. Surv.* **2017**, *50*, 1–35. [[CrossRef](#)]
15. Zhu, Z.; Hu, H. Robot Learning from Demonstration in Robotic Assembly: A Survey. *Robot* **2018**, *7*, 17. [[CrossRef](#)]
16. Ravichandar, H.; Polydoros, A.S.; Chernova, S.; Billard, A. Recent Advances in Robot Learning from Demonstration. *Annu. Rev. Control. Robot. Auton. Syst.* **2020**, *3*, 297–330. [[CrossRef](#)]
17. Xie, Z.W.; Zhang, Q.; Jiang, Z.N.; Liu, H. Robot learning from demonstration for path planning: A review. *Sci. China Technol. Sci.* **2020**, *63*, 1325–1334. [[CrossRef](#)]
18. Kitchenham, B. *Procedures for Performing Systematic Reviews*; Keele University: Keele, UK, 2004.
19. Kitchenham, B.; Brereton, O.P.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. Systematic literature reviews in software engineering—A systematic literature review. *Inf. Softw. Technol.* **2008**, *51*, 7–15. [[CrossRef](#)]
20. Xiao, Y.; Watson, M. Guidance on conducting a systematic literature review. *J. Plan. Educ. Res.* **2019**, *39*, 93–112. [[CrossRef](#)]
21. Scells, H.; Zuccon, G. Generating better queries for systematic reviews. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 475–484. [[CrossRef](#)]
22. Ananiadou, S.; Rea, B.; Okazaki, N.; Procter, R.; Thomas, J. Supporting systematic reviews using text mining. *Soc. Sci. Comput. Rev.* **2009**, *27*, 509–523. [[CrossRef](#)]
23. Tsafnat, G.; Glasziou, P.; Choong, M.K.; Dunn, A.; Galgani, F.; Coiera, E. Systematic review automation technologies. *Syst. Rev.* **2014**, *3*, 74. [[CrossRef](#)]
24. Billard, A.G.; Calinon, S.; Dillmann, R. Learning from Humans. In *Springer Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 1995–2014. [[CrossRef](#)]
25. Chernova, S.; Thomaz, A.L. Robot learning from human teachers. *Synth. Lect. Artif. Intell. Mach. Learn.* **2014**, *28*, 1–121. [[CrossRef](#)]
26. Zhou, Z.; Xiong, R.; Wang, Y.; Zhang, J. Advanced Robot Programming: A Review. *Curr. Robot. Rep.* **2020**, *1*, 251–528. [[CrossRef](#)]
27. Koskinopoulou, M.; Piperakis, S.; Trahanias, P. Learning from demonstration facilitates human-robot collaborative task execution. In Proceedings of the 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Christchurch, New Zealand, 7–10 March 2016; pp. 59–66. [[CrossRef](#)]
28. Qu, J.; Zhang, F.; Wang, Y.; Fu, Y. Human-like coordination motion learning for a redundant dual-arm robot. *Robot. Comput.-Integr. Manuf.* **2019**, *57*, 379–390. [[CrossRef](#)]
29. Wang, W.; Chen, Y.; Li, R.; Jia, Y. Learning and comfort in human-robot interaction: A review. *Appl. Sci.* **2019**, *9*, 5152. [[CrossRef](#)]
30. Lopes, M.; Melo, F.; Montesano, L. *Active Learning for Reward Estimation in Inverse Reinforcement Learning*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5782, pp. 31–46. [[CrossRef](#)]
31. Wang, W.; Li, R.; Chen, Y.; Diekel, Z.M.; Jia, Y. Facilitating Human-Robot Collaborative Tasks by Teaching-Learning-Collaboration from Human Demonstrations. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 640–653. [[CrossRef](#)]

32. Ijspeert, A.J.; Nakanishi, J.; Schaal, S. Learning rhythmic movements by demonstration using nonlinear oscillators. *IEEE Int. Conf. Intell. Robot. Syst.* **2002**, *1*, 958–963. [[CrossRef](#)]
33. Rabiner, L.R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc. IEEE* **1989**, *77*, 257–286. [[CrossRef](#)]
34. Fink, G.A. *Markov Models for Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2014. [[CrossRef](#)]
35. Parsons, O.E. *A Gaussian Mixture Model Approach to Classifying Response Types*; Springer: Cham, Switzerland, 2020; pp. 3–22. [[CrossRef](#)]
36. Ghahramani, Z.; Jordan, M. Supervised learning from incomplete data via an EM approach. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 30 November–3 December 1993; Cowan, J., Tesauro, G., Alspector, J., Eds.; Morgan-Kaufmann: Burlington, MA, USA, 1993; Volume 6.
37. Fabisch, A. gmr: Gaussian Mixture Regression. *J. Open Source Softw.* **2021**, *6*, 3054. [[CrossRef](#)]
38. Odom, P.; Natarajan, S. Active Advice Seeking for Inverse Reinforcement Learning. In Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, 9–13 May 2016; pp. 503–511.
39. Nicolescu, M.N.; Matarić, M.J. Task learning through imitation and human–robot interaction. In *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*; Nehaniv, C.L., Dautenhahn, K., Eds.; Cambridge University Press: Cambridge, UK, 2007; pp. 407–424. [[CrossRef](#)]
40. Luo, Y.; Yin, L.; Bai, W.; Mao, K. An Appraisal of Incremental Learning Methods. *Entropy* **2020**, *22*, 1190. [[CrossRef](#)]
41. Ewerton, M.; Maeda, G.; Kollegger, G.; Wiemeyer, J.; Peters, J. Incremental imitation learning of context-dependent motor skills. In Proceedings of the 2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, Mexico, 15–17 November 2016; pp. 351–358. [[CrossRef](#)]
42. Nozari, S.; Krayani, A.; Marcenaro, L.; Martin, D.; Regazzoni, C. Incremental Learning through Probabilistic Behavior Prediction. In Proceedings of the 2022 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, 29 August–2 September 2022; pp. 1502–1506.
43. Mészáros, A.; Franzese, G.; Kober, J. Learning to Pick at Non-Zero-Velocity From Interactive Demonstrations. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6052–6059. [[CrossRef](#)]
44. Kober, J.; Bagnell, J.A.; Peters, J. Reinforcement learning in robotics: A survey. *Int. J. Robot. Res.* **2013**, *32*, 1238–1274. [[CrossRef](#)]
45. Akkaladevi, S.C.; Plasch, M.; Maddukuri, S.; Eitzinger, C.; Pichler, A.; Rinner, B. Toward an interactive reinforcement based learning framework for human robot collaborative assembly processes. *Front. Robot. AI* **2018**, *5*, 126. [[CrossRef](#)]
46. Winter, J.D.; Beir, A.D.; Makrini, I.E.; de Perre, G.V.; Nowé, A.; Vanderborght, B. Accelerating interactive reinforcement learning by human advice for an assembly task by a cobot. *Robotics* **2019**, *8*, 104. [[CrossRef](#)]
47. Lai, Y.; Paul, G.; Cui, Y.; Matsubara, T. User intent estimation during robot learning using physical human robot interaction primitives. *Auton. Robot.* **2022**, *46*, 421–436. [[CrossRef](#)]
48. Hu, H.; Yang, X.; Lou, Y. A robot learning from demonstration framework for skillful small parts assembly. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 6775–6787. [[CrossRef](#)]
49. Zhang, S.; Huang, H.; Huang, D.; Yao, L.; Wei, J.; Fan, Q. Subtask-learning based for robot self-assembly in flexible collaborative assembly in manufacturing. *Int. J. Adv. Manuf. Technol.* **2022**, *120*, 6807–6819. [[CrossRef](#)]
50. Hu, Y.; Wang, Y.; Hu, K.; Li, W. Adaptive obstacle avoidance in path planning of collaborative robots for dynamic manufacturing. *J. Intell. Manuf.* **2021**, *1*–19. [[CrossRef](#)]
51. Wang, L.; Jia, S.; Wang, G.; Turner, A.; Ratchev, S. Enhancing learning capabilities of movement primitives under distributed probabilistic framework for flexible assembly tasks. *Neural Comput. Appl.* **2021**, *1*–12. [[CrossRef](#)]
52. Coninck, E.D.; Verbelen, T.; Molle, P.V.; Simoens, P.; Dhoedt, B. Learning robots to grasp by demonstration. *Robot. Auton. Syst.* **2020**, *127*, 103474. [[CrossRef](#)]
53. Steinmetz, F.; Nitsch, V.; Stulp, F. Intuitive Task-Level Programming by Demonstration Through Semantic Skill Recognition. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3742–3749. [[CrossRef](#)]
54. Schlette, C.; Buch, A.G.; Hagelskjaer, F.; Iturrate, I.; Kraft, D.; Kramberger, A.; Lindvig, A.P.; Mathiesen, S.; Petersen, H.G.; Rasmussen, M.H.; et al. Advanced Robotics Towards robot cell matrices for agile production-SDU Robotics’ assembly cell at the WRC 2018 Towards robot cell matrices for agile production-SDU Robotics’ assembly cell at the WRC 2018. *Adv. Robot.* **2019**, *2020*, 422–438. [[CrossRef](#)]
55. Kyrarini, M.; Haseeb, M.A.; Ristić-Durrant, D.; Gräser, A. Robot learning of industrial assembly task via human demonstrations. *Auton. Robot.* **2019**, *43*, 239–257. [[CrossRef](#)]
56. Raiola, G.; Restrepo, S.S.; Chevalier, P.; Rodriguez-Ayerbe, P.; Lamy, X.; Tliba, S.; Stulp, F. Co-manipulation with a library of virtual guiding fixtures. *Auton. Robot.* **2018**, *42*, 1037–1051. [[CrossRef](#)]
57. Esfahani, A.M.G.; Ragaglia, M. Robot learning from demonstrations: Emulation learning in environments with moving obstacles. *Robot. Auton. Syst.* **2018**, *101*, 45–56. [[CrossRef](#)]
58. Rozo, L.; Silvério, J.; Calinon, S.; Caldwell, D.G. Learning Controllers for Reactive and Proactive Behaviors in Human–Robot Collaboration. *Front. Robot. AI* **2016**, *3*, 30. [[CrossRef](#)]
59. Rozo, L.; Calinon, S.; Caldwell, D.G.; Jiménez, P.; Torras, C. Learning Physical Collaborative Robot Behaviors From Human Demonstrations. *IEEE Trans. Robot.* **2016**, *32*, 513–527. [[CrossRef](#)]

60. Iturrate, I.; Kramberger, A.; Sloth, C. Quick Setup of Force-Controlled Industrial Gluing Tasks Using Learning From Demonstration. *Front. Robot. AI* **2021**, *8*, 354. [[CrossRef](#)]
61. Wang, Y.Q.; Hu, Y.D.; Zaatari, S.E.; Li, W.D.; Zhou, Y. Optimised Learning from Demonstrations for Collaborative Robots. *Robot.-Comput.-Integr. Manuf.* **2021**, *71*, 102169. [[CrossRef](#)]
62. Liang, Y.S.; Pellier, D.; Fiorino, H.; Pesty, S. Evaluation of a Robot Programming Framework for Non-Experts Using Symbolic Planning Representations. In Proceedings of the 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Lisbon, Portugal, 28 August–1 September 2017; pp. 1121–1126. [[CrossRef](#)]
63. Ramirez-Amaro, K.; Dean-Leon, E.; Bergner, F.; Cheng, G. A Semantic-Based Method for Teaching Industrial Robots New Tasks. *KI-Kunstl. Intell.* **2019**, *33*, 117–122. [[CrossRef](#)]
64. Fu, J.; Du, J.; Teng, X.; Fu, Y.; Wu, L. Adaptive multi-task human-robot interaction based on human behavioral intention. *IEEE Access* **2021**, *9*, 133762–133773. [[CrossRef](#)]
65. Liang, Y.S.; Pellier, D.; Fiorino, H.; Pesty, S. iRoPro: An interactive Robot Programming Framework. *Int. J. Soc. Robot.* **2022**, *14*, 177–191. [[CrossRef](#)]
66. Schou, C.; Andersen, R.S.; Chrysostomou, D.; Bøgh, S.; Madsen, O. Skill-based instruction of collaborative robots in industrial settings. *Robot. Comput.-Integr. Manuf.* **2018**, *53*, 72–80. [[CrossRef](#)]
67. Wang, N.; Chen, C.; Nuovo, A.D. A Framework of Hybrid Force/Motion Skills Learning for Robots. *IEEE Trans. Cogn. Dev. Syst.* **2021**, *13*, 162–170. [[CrossRef](#)]
68. Haage, M.; Piperagkas, G.; Papadopoulos, C.; Mariolis, I.; Malec, J.; Bekiroglu, Y.; Hedelind, M.; Tzovaras, D. Teaching Assembly by Demonstration Using Advanced Human Robot Interaction and a Knowledge Integration Framework. *Procedia Manuf.* **2017**, *11*, 164–173. [[CrossRef](#)]
69. Wu, M.; Taetz, B.; He, Y.; Bleser, G.; Liu, S. An adaptive learning and control framework based on dynamic movement primitives with application to human-robot handovers. *Robot. Auton. Syst.* **2022**, *148*, 103935. [[CrossRef](#)]
70. Sun, Y.; Wang, W.; Chen, Y.; Jia, Y. Learn How to Assist Humans Through Human Teaching and Robot Learning in Human-Robot Collaborative Assembly. *IEEE Trans. Syst. Man, Cybern. Syst.* **2022**, *52*, 728–738. [[CrossRef](#)]
71. Huang, B.; Ye, M.; Hu, Y.; Vandini, A.; Lee, S.L.; Yang, G.Z. A Multirobot Cooperation Framework for Sewing Personalized Stent Grafts. *IEEE Trans. Ind. Inf.* **2018**, *14*, 1776–1785. [[CrossRef](#)]
72. Castelli, F.; Michieletto, S.; Ghidoni, S.; Pagello, E. A machine learning-based visual servoing approach for fast robot control in industrial setting. *Int. J. Adv. Robot. Syst.* **2017**, *14*, 1729881417738884. [[CrossRef](#)]
73. Zhang, H.D.; Liu, S.B.; Lei, Q.J.; He, Y.; Yang, Y.; Bai, Y. Robot programming by demonstration: A novel system for robot trajectory programming based on robot operating system. *Adv. Manuf.* **2020**, *8*, 216–229. [[CrossRef](#)]
74. Zaatari, S.E.; Wang, Y.; Hu, Y.; Li, W. An improved approach of task-parameterized learning from demonstrations for cobots in dynamic manufacturing. *J. Intell. Manuf.* **2021**, *33*, 1503–1519. [[CrossRef](#)]
75. Zaatari, S.E.; Wang, Y.; Li, W.; Peng, Y. iTP-LfD: Improved task parametrised learning from demonstration for adaptive path generation of cobot. *Robot. Comput.-Integr. Manuf.* **2021**, *69*, 102109. [[CrossRef](#)]
76. Rodriguez-Linan, A.; Lopez-Juarez, I.; Maldonado-Ramirez, A.; Zalapa-Elias, A.; Torres-Trevino, L.; Navarro-Gonzalez, J.L.; Chinas-Sanchez, P. An Approach to Acquire Path-Following Skills by Industrial Robots from Human Demonstration. *IEEE Access* **2021**, *9*, 82351–82363. [[CrossRef](#)]
77. Racca, M.; Kyrki, V.; Cakmak, M. Interactive Tuning of Robot Program Parameters via Expected Divergence Maximization. *HRI ACM/IEEE Int. Conf. Hum.-Robot Interact.* **2020**, *10*, 629–638. [[CrossRef](#)]
78. Peternel, L.; Tsagarakis, N.; Caldwell, D.; Ajoudani, A. Robot adaptation to human physical fatigue in human–robot co-manipulation. *Auton. Robot.* **2018**, *42*, 1011–1021. [[CrossRef](#)]
79. Al-Yacoub, A.; Zhao, Y.C.; Eaton, W.; Goh, Y.M.; Lohse, N. Improving human robot collaboration through Force/Torque based learning for object manipulation. *Robot. Comput.-Integr. Manuf.* **2021**, *69*. [[CrossRef](#)]
80. Zeng, C.; Yang, C.; Zhong, J.; Zhang, J. Encoding Multiple Sensor Data for Robotic Learning Skills from Multimodal Demonstration. *IEEE Access* **2019**, *7*, 145604–145613. [[CrossRef](#)]
81. Soares, I.; Petry, M.; Moreira, A.P. Programming Robots by Demonstration Using Augmented Reality. *Sensors* **2021**, *21*, 5976. [[CrossRef](#)]
82. Lee, H.; Kim, D.; Aman, M.; Amin, U.A. Control framework for collaborative robot using imitation learning-based teleoperation from human digital twin to robot digital twin. *Mechatronics* **2022**, *85*, 102833. [[CrossRef](#)]
83. Koert, D.; Trick, S.; Ewerton, M.; Lutter, M.; Peters, J. Incremental Learning of an Open-Ended Collaborative Skill Library. *Int. J. Humanoid Robot.* **2020**, *17*, 2050001. [[CrossRef](#)]
84. Qian, K.; Xu, X.; Liu, H.; Bai, J.; Luo, S. Environment-adaptive learning from demonstration for proactive assistance in human–robot collaborative tasks. *Robot. Auton. Syst.* **2022**, *151*, 104046. [[CrossRef](#)]
85. Tang, T.; Lin, H.C.; Zhao, Y.; Fan, Y.; Chen, W.; Tomizuka, M. Teach industrial robots peg-hole-insertion by human demonstration. In Proceedings of the 2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Banff, AB, Canada, 12–15 July 2016; pp. 488–494. [[CrossRef](#)]
86. Schaal, S.; Peters, J.; Nakanishi, J.; Ijspeert, A. Learning movement primitives. *Springer Tracts Adv. Robot.* **2005**, *15*, 561–572. [[CrossRef](#)]

87. Hoffmann, H.; Pastor, P.; Park, D.H.; Schaal, S. Biologically-inspired dynamical systems for movement generation: Automatic real-time goal adaptation and obstacle avoidance. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 2587–2592. [[CrossRef](#)]
88. Paraschos, A.; Daniel, C.; Peters, J.; Neumann, G. Probabilistic movement primitives. In Proceedings of the 27th Annual Conference on Neural Information Processing Systems (NIPS 2013), Lake Tahoe, NV, USA, 5–10 December 2013.
89. Paraschos, A.; Daniel, C.; Peters, J.; Neumann, G. Using probabilistic movement primitives in robotics. *Auton. Robot.* **2017**, *42*, 529–551. [[CrossRef](#)]
90. Maeda, G.J.; Neumann, G.; Ewerton, M.; Lioutikov, R.; Kroemer, O.; Peters, J. Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks. *Auton. Robot.* **2017**, *41*, 593–612. [[CrossRef](#)]
91. Koert, D.; Trick, S.; Ewerton, M.; Lutter, M.; Peters, J. Online Learning of an Open-Ended Skill Library for Collaborative Tasks. In Proceedings of the 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), Beijing, China, 6–9 November 2018; pp. 1–9. [[CrossRef](#)]
92. Amor, H.B.; Neumann, G.; Kamthe, S.; Kroemer, O.; Peters, J. Interaction primitives for human-robot cooperation tasks. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2014; pp. 2831–2837. [[CrossRef](#)]
93. Calinon, S.; Guenter, F.; Billard, A. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2007**, *37*, 286–298. [[CrossRef](#)] [[PubMed](#)]
94. Calinon, S.; D’Halluin, F.; Sauser, E.L.; Caldwell, D.G.; Billard, A.G. Learning and reproduction of gestures by imitation. *IEEE Robot. Autom. Mag.* **2010**, *17*, 44–54. [[CrossRef](#)]
95. Carfi, A.; Villalobos, J.; Coronado, E.; Bruno, B.; Mastrogiovanni, F. Can Human-Inspired Learning Behaviour Facilitate Human–Robot Interaction? *Int. J. Soc. Robot.* **2020**, *12*, 173–186. [[CrossRef](#)]
96. Carmigniani, J.; Furht, B. Augmented reality: An overview. In *Handbook of Augmented Reality*; Springer: New York, NY, USA, 2011; pp. 3–46.
97. Sherman, W.R.; Craig, A.B. *Understanding Virtual Reality*; Morgan Kaufman: San Francisco, CA, USA, 2003.
98. Grieves, M.; Vickers, J. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. In *Transdisciplinary Perspectives on Complex Systems*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 85–113.
99. Palmarini, R.; Amo, I.F.D.; Bertolino, G.; Dini, G.; Erkoyuncu, J.A.; Roy, R.; Farnsworth, M. Designing an AR interface to improve trust in Human-Robots collaboration. In Proceedings of the 28th CIRP Design Conference, Nantes, France, 23–25 May 2018; Volume 70, pp. 350–355. [[CrossRef](#)]
100. Shu, B.; Sziebig, G.; Pieters, R. Architecture for Safe Human-Robot Collaboration: Multi-Modal Communication in Virtual Reality for Efficient Task Execution. In Proceedings of the 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), Vancouver, BC, Canada, 12–14 June 2019; pp. 2297–2302. [[CrossRef](#)]
101. Materna, Z.; Kapinus, M.; Beran, V.; Smrž, P.; Zemčík, P. Interactive Spatial Augmented Reality in Collaborative Robot Programming: User Experience Evaluation. In Proceedings of the 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Nanjing, China, 27–31 August 2018; pp. 80–87. [[CrossRef](#)]
102. Bambussek, D.; Materna, Z.Z.; Kapinus, M.; Beran, V.V.; Smrz, P.; Bambušek, D.; Materna, Z.Z.; Kapinus, M.; Beran, V.V.; Smrž, P. Combining Interactive Spatial Augmented Reality with Head-Mounted Display for End-User Collaborative Robot Programming. In Proceedings of the 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), New Delhi, India, 14–19 October 2019; pp. 1–8. [[CrossRef](#)]
103. Manou, E.; Vosniakos, G.C.; Matsas, E. Off-line programming of an industrial robot in a virtual reality environment. *Int. J. Interact. Des. Manuf.* **2019**, *13*, 507–519. [[CrossRef](#)]
104. Burghardt, A.; Szybicki, D.; Gierlak, P.; Kurc, K.; Pietruś, P.; Cygan, R. Programming of Industrial Robots Using Virtual Reality and Digital Twins. *Appl. Sci.* **2020**, *10*, 486. [[CrossRef](#)]
105. Malik, A.A.; Brem, A. Digital twins for collaborative robots: A case study in human-robot interaction. *Robot. Comput.-Integr. Manuf.* **2021**, *68*, 102092. [[CrossRef](#)]
106. Pérez, L.; Rodríguez-Jiménez, S.; Rodríguez, N.; Usamentiaga, R.; García, D.F. Digital twin and virtual reality based methodology for multi-robot manufacturing cell commissioning. *Appl. Sci.* **2020**, *10*, 3633. [[CrossRef](#)]
107. Xia, K.; Sacco, C.; Kirkpatrick, M.; Saidy, C.; Nguyen, L.; Kircaliali, A.; Harik, R. A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence. *J. Manuf. Syst.* **2020**, *58*, 210–230. [[CrossRef](#)]
108. Pedersen, M.R.; Nalpantidis, L.; Andersen, R.S.; Schou, C.; Bøgh, S.; Krüger, V.; Madsen, O. Robot skills for manufacturing: From concept to industrial deployment. *Robot. Comput.-Integr. Manuf.* **2016**, *37*, 282–291. [[CrossRef](#)]