Recent Advances of Deep Robotic Affordance Learning: A Reinforcement Learning Perspective

Xintong Yang[1], Ze Ji[1], Jing Wu[2], and Yu-Kun Lai[2]

***Abstract*—As a popular concept proposed in the field of psychology, affordance has been regarded as one of the important abilities that enable humans to understand and interact with the environment. Briefly, it captures the possibilities and effects of the actions of an agent applied to a specific object or, more generally, a part of the environment. This paper provides a short review of the recent developments of deep robotic affordance learning (DRAL), which aims to develop data-driven methods that use the concept of affordance to aid in robotic tasks. We first classify these papers from a reinforcement learning (RL) perspective and draw connections between RL and affordances. The technical details of each category are discussed and their limitations are identified. We further summarise them and identify future challenges from the aspects of observations, actions, affordance representation, data-collection and real-world deployment. A final remark is given at the end to propose a promising future direction of the RL-based affordance definition to include the predictions of arbitrary action consequences.**

I. Introduction

Humans interact with various objects in the environment in a purposeful and meaningful way, because we have the ability to understand affordances – the functionalities of objects, the possibilities and effects of our actions and the relationship between the two. As originally defined by Gibson [1], the affordances of an object or a place in an environment provide knowledge about what actions are possible and what the consequences of these actions are with respect to a certain agent (a human, an animal or a robot). In short, it indicates **possibilities** and **effects** of the agent's actions given an object or a part (an image observation) of the environment. In the field of robotics, affordances could serve with great potential to bridge robot perception and action [2]. This has been actively integrated and explored with machine learning techniques in recent years [3]–[6]. Jamone *et al.* proposed a thorough review and drew connections among the studies of affordances in psychology, neuroscience and robotics [3]. Yamanobe *et al.* summarised the use of affordances specifically in robotic manipulation tasks [4]. Ardón *et al.* summarised and provided guidance on design choices and how affordance relations can be used to boost policy learning [5].

Manuscript received: Month, Day, Year; Revised Month, Day, Year; Accepted Month, Day, Year.

This paper was recommended for publication by Editor Editor A. Name upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by (organizations/grants which supported the work.)

(Corresponding author: Ze Ji)

[1]School of Engineering, Cardiff University, Cardiff, UK {yangx66, jiz1}@cardiff.ac.uk

[2]School of Computer Science and Informatics, Cardiff University, Cardiff, UK {wuj11, laiy4}@cardiff.ac.uk

However, as pointed out in [6], there is still a lack of consensus for a formal definition of affordances, and many previous works are limited to affordances in the form of object functionalities [6]–[8]. A number of existing mathematical formulations focus on statistical relationships between the agent, its actions and its environment, but are not general enough to be integrated into ANY main-stream robot control frameworks to support both action inference and affordance learning [3], [4], [9]. The main reason is the lack of a rigorous mathematical connection between the concept of affordances and robot control theory without assuming any prior knowledge of high-level observation construction [7], [8], be it learning-based or model-based. Recently, Khetarpal *et al.* [10] proposed to define, learn and compute affordances based on the reinforcement learning (RL) paradigm with Markov decision processes (MDPs) of any kind, which is a classic and increasingly important robot control paradigm [11], [12]. We propose in this paper to summarise and classify recent publications (since 2015) in deep robotic affordance learning (DRAL) following the RL-based definition in [10] for the following motivations:

- The RL-based definition helps to unify and classify DRAL works from a behavioural learning perspective, providing new insights to understand and clarify the different usages of affordances in the literature;
- The definition in [10] is the most general in the literature as all concepts are defined over a generic MDP without any assumption of the environmental or agent aspect. It suits any kind of environmental affordances and agents as long as they can be described by MDPs, which is commonly achievable.
- As the primary aim of DRAL is to enable robots to infer afforded actions, the RL community provides a rich body of methods ready to be integrated with affordances;
- Understanding and analysing the concept based on a mathematical framework helps to provide computationally and practically valuable insights.

In practice, knowing the affordances means knowing the desired effects of some actions and whether these effects can be realised in some situations. With this in mind, Khetarpal *et al.* introduced the notion of *intents* that captures the desired outcome of an action based on the reinforcement learning (RL) framework [10], [11]. For example, the intent of a moving right action in a gridworld task is the agent being moved to the cell on the right. The intent is not always satisfied, e.g., when the cell on the right is a wall. Thus, the definition of affordance is a subset of the state-action space in which the intent is indeed satisfied [10]. In order words, the moving right action is afforded at every state where the moving right intent is satisfied.

Notice that there are two levels of the topic: 1) the learning and discovery of affordances and 2) the use of affordances. Researchers have only recently started to study the first level, e.g., option/subgoal discovery [13]. Most research focuses on the use of the knowledge of affordances, meaning how to estimate the action possibilities and/or infer the afforded actions. These works are classified into three categories as follows.

- For the majority of the DRAL works, the focus is to estimate the action possibilities given an observation and then infer afforded actions from it (Section III). These works can be further classified into methods that model the action possibilities as binary variables (subsection III-A) [14]–[21] and continuous variables (subsection III-B) [22]–[27];
- The second line of works proposes to generate afforded actions from a set of object keypoints (Section IV) [28]–[33]. The keypoints are used to geometrically constrain the search space of action inference methods within the set of afforded actions.
- The last part of the reviewed papers suggests learning a partial dynamic model for only afforded actions, resulting in faster model learning and motion planning (Section V) [10], [34], [35].

The rest of this review is organised as follows. Section II briefly recalls the definition of affordances in reinforcement learning proposed by [10], classifies the reviewed works and draws connections between RL and affordances. Sections III, V and IV provide the main technical ideas and discuss the pros and cons of the reviewed papers. Section VI summarises these works and poses future challenges from the perspectives of observations, actions, affordance representations, data collection, and real-world deployment. Section VII concludes this review.

II. Affordance definition in MDPs

For the sake of clarity, we recall in this section the reinforcement learning (RL) problem and the definition of affordance based on the Markov Decision Processes (MDPs) [10].

An MDP is a tuple $M = \langle \mathcal{S}, \mathcal{A}, r, P, \gamma \rangle$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, r is the reward function, $P(s'|s,a)$ is the system transition dynamics and $\gamma \in [0,1]$ is the discount factor [11]. The RL problem is in general to find an optimal policy, $\pi : \mathcal{S} \rightarrow \mathcal{A}$, which produces actions that maximise the expected discounted future return $\mathbb{E}_\pi[G_t] = \mathbb{E}_\pi\left[\sum_t^\infty \gamma^t r_t\right]$. The typical process of learning such a policy loops over the procedures of data collection, policy evaluation and policy improvement [11].

Given an action $a \in \mathcal{A}$, an intent $I_a(s)$ maps a state $s \in \mathcal{S}$ to a state distribution that the action is intended to achieve. The intent model can thus be seen as a partial dynamic model: $P_I(s'|s,a)$, which only captures the dynamics for a subset of states where the action has a desired effect. Given the full system dynamic model $P(s'|s,a)$, an intent is satisfied (i.e., an action is affordable) at a state, to a degree ϵ, if and only if:

$$d(P_I(s'|s,a), P(s'|s,a)) \leq \epsilon \tag{1}$$

where d is a function that measures the difference between two distributions and $\epsilon \in [0,1]$ is a precision parameter. Given a set of intents $\mathcal{I} = \cup_{a\in\mathcal{A}} I_a$, the affordance is then defined as a relation $\mathcal{AF}_\mathcal{I} \subseteq \mathcal{S} \times \mathcal{A}$, such that $\forall(s,a) \in \mathcal{AF}_\mathcal{I}$, Eq. 1 is satisfied. Accordingly, an affordance prediction model, or an action possibility model, gives the probability of whether a pair of state and action belongs to the set of affordance:

$$p^{\mathcal{AF}}(s,a) = p((s,a) \in \mathcal{AF}_\mathcal{I}) \tag{2}$$

Remark 1: Practically speaking, knowing the affordance set means knowing the desired effects of a subset of actions (intents, action effects) and the subset of states where these effects can be achieved (states where the intents are satisfied, action possibilities). Before inferring the afforded actions or computing the action possibilities, one must know what actions, or what effects, are concerned or to be used. This logic implies that a robot must have learnt or been given some prior knowledge of the concerned actions beforehand. At the current stage of DRAL research, this knowledge was given by researchers, who then focused on the estimation of action possibilities and the inference of afforded actions. We categorise and discuss these methods in three classes:

- works that tried to infer the afforded actions from the estimated action possibilities $\hat{p}^{\mathcal{AF}}$ (section III);
- works that tried to infer the afforded actions of objects in terms of keypoints (section IV);
- works that tried to infer afforded actions by planning with $\hat{p}^{\mathcal{AF}}$ and a learnt partial dynamic model associated with intents, $\hat{P}_I(s'|s,a)$ (section V).

In the following sections, especially Sections III and IV, the readers shall see that most recent works using affordances in robotics do not reside their methods in the RL framework, although these methods can be explained from the RL perspective.

Remark 2: From the RL perspective, or a behavioural learning perspective, the knowledge of affordances can help to accelerate and improve almost every aspect of the RL process by constraining the action space. These include 1) the learning of a value function, a policy, or a world model, 2) the design of an exploration strategy, and 3) the action inference process (stochastic sampling, planning or value-based greedy actions). For example, if an action possibility model is available, one can integrate it into the exploration process of any RL algorithm, such that it only collects experiences, where actions do cause changes to the environment. Alternatively, one may constrain the updates of a policy within the set of affordable actions. Also, as demonstrated by [10], focusing on the set of afforded actions simplifies the learning of a world model and accelerates planning.

Either for data collection, policy learning, world-model learning or action planning, the use of affordances in RL may have its best potential in the hierarchical reinforcement learning (HRL) framework, where an agent learns to use a set of motion primitives (sub-policies, skills, temporally-extended actions) to achieve different tasks [36]. Knowing the possibilities and effects of the skills can accelerate learning by 1) constraining and guiding the choices of exploring skills, and 2) filtering out experiences with irrelevant or non-effective actions. These will lead to shortened exploration time and increased sample efficiency, as it is effectively shrinking the size of the solution space (or the number of valid actions).

Remark 3: A further step to take in this regard is the learning and discovery of affordances. Knowing the set of affordances is promising and valuable in terms of accelerating learning; however, enabling an agent to learn and discover affordances makes the agent robust to potential changes in the environment and the agent itself. This is closely related to the popular topic of option/subgoal discovery in HRL [13]. Future research topics in this regard include learning new skills, adapting old skills, skill composition, action space design, etc. One can envision a robot acquiring new skills in a new environment or modifying old skills as its hardware wear and tear.

III. Modelling Action Possibilities

This section discusses recent papers on modelling and learning action possibilities. This section examines two lines of works that represents $\hat{p}^{\mathcal{AF}}$ in Eq.2 (the probabilities of an action or a set of actions being affordable given an observation) as binary segmentation masks (III-A) and continuous action success scores (III-B). We summarise these works and discuss their limitations in Section III-C.

Based on the definition given in Section II, these methods compute $\hat{p}^{\mathcal{AF}}$ for a set of actions given a state. The estimated $\hat{p}^{\mathcal{AF}}$ can be used to infer desirable actions in various ways based on its representations, such as taking the action with the maximum possibility, i.e., computing $\arg\max_{a\in\mathcal{A}}\ \hat{p}^{\mathcal{AF}}$. In practice, computing $\hat{p}^{\mathcal{AF}}$ is commonly based on sensory observations, such as point clouds or images, instead of the true system states. The observation representations, training methods, deployment tasks and motion generation methods adopted by these works are summarised in Table I.

A. Image/point cloud segmentation

Many works propose to model what actions are afforded on which part of an object as an image or point cloud segmentation problem [14]–[21]. In these works, a segmented part of an object image or point cloud is labelled with one or more affordable actions, i.e., a binary mask that indicates whether an action can be applied to that part of the object. The action possibilities are simplified into binary variables and represented as pixel-level masks. For example, as shown in Fig. 1, the pixels or points of the handle of a cup are labelled as being graspable, while those of the hollow part of the cup are labelled as containable. It is common for different parts of an object to have different affordances. It is also common for the same part of an object to have multiple affordances [18].

As a natural extension, these pixel-level or point-level affordance predictions were used to provide the downstream manipulation policy with extra task information. The most straightforward way in grasping tasks is to designate the centre of the detected affordance masks as a grasping location [18]. A more recent method treated the predicted segmentation masks as an extra channel of the image observations. A manipulation policy then processed this extended image to determine what actions to take [19]. A self-supervised learning method was proposed to learn to predict the pixel masks for gripper-object interaction centres from human teleoperation demonstrations of a table tidy-up task [21]. These pixel masks were then used in the real world for a model-based policy to move the gripper closer to the interaction point of an object and a reinforcement learning policy to pick up the object. There was also an attempt to learn a latent representation of object affordances with Variational Auto-Encoders [20], [37]. It was successfully trained using simulation data and transferred to a real-world robotic system, aided by a domain randomisation technique. They used the latent representation to generate robot trajectories that move the gripper to a point above a cup [20].

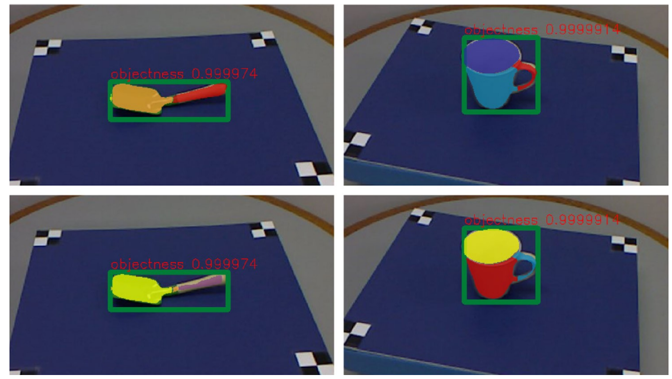


Fig. 1: Segmented images from [18]. Red parts afford grasping, orange afford supporting, deep blue afford containing, blue afford wrap-grasping, and purple afford pounding.

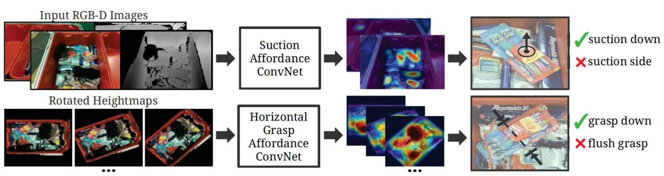
B. Action scores

Several works proposed to represent the action possibility as a continuous variable that indicates how confident it is that an action can be successfully executed (is affordable) [22]–[27]. In contrast, the segmentation masks discussed in the last subsection are binary variables.

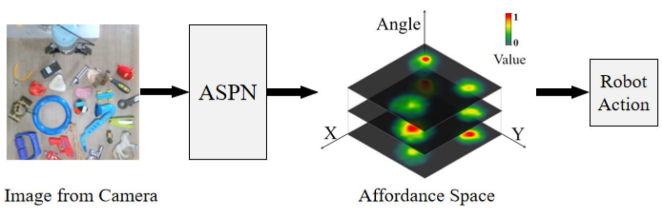
Zeng *et al.* proposed to model the success probabilities of four kinds of primitive grasping and suction actions given the RGB-D observation of a cluttered scene [22]. The probability distributions are defined as matrices whose entries represent the success rates of executing actions at the pixel locations (see Fig. 2). Similarly, Cai *et al.* proposed to predict graspability, ungraspability and background affordances over image pixels, achieving a grasping success rate of 93% on a set of household items, 91% on a set of adversarial items and 87% in clutter scenarios [23]. The network was trained with synthetic data generated by an antipodal grasp heuristic in simulation in a self-supervised fashion. Wu *et al.* extended such a 2D affordance map defined in the pixel space into a 3D space,

TABLE I: Summary of papers focused on learning action possibilities. **Cat.:** category; **IPS:** image/point cloud segmentation; **AS:** action scores; **PCD:** point cloud data; **SL:** supervised learning; **SSL:** self-supervised learning; **Sim:** simulation **Real:** real-world; **DoF:** degree of freedom; **PJG:** parallel-jaw grasp; **BOSM:** binary object segmentation mask; **RL:** reinforcement learning; **Sim-to-Real:** simulation to real-world transfer.

Paper	Cat.	Affordance (afforded actions)	Input	Method	Deployment Task	Motion
[14]	IPS	Created UMD dataset	RGB-D	SL	None	-
[15]	IPS	Grasp; Cut; Poke; Pound; Pour; Support	PCD	SL	3-Finger Dexterous grasp (Sim)	Planning
[16]	IPS	Created IIT-AFF dataset	RGB	SL	Dexterous grasp (Real)	Planning
[17]	IPS	from IIT-AFF[16]; UMD[14] datasets	RGB	SL	Dexterous grasp (Real)	Planning
[18]	IPS	from UMD[14] dataset	RGB-D	SL	4DoF PJG; bean-scoop (Real)	Planning
[19]	IPS	Dexterous grasp	RGB-D	SL	Dexterous grasp (Sim)	RL
[20]	IPS	from UMD[14] dataset	RGB/RGB-D	SL	Cup-locate (Sim & Real)	Planning
[21]	IPS	Grasp	RGB-D	SSL	4DoF PJG (Sim & Real)	Primitive & RL
[22]	AS	Grasp; Suction	RGB-D	SL	3DoF PJG & suction (Real)	Primitive
[23]	AS	Grasp	RGB	SSL	4DoF PJG (Sim & Real)	Primitive
[24]	AS	Grasp; Push	RGB	SSL	4DoF PJG & push (Sim & Real)	Primitive
[25]	AS	Grasp	BOSM	SSL	4DoF PJG (Sim-to-Real)	Primitive & RL
[26]	AS	Push; Pull	RGB-D & PCD	SSL	Push & pull (Sim)	Primitive
[27]	AS	Pick; Move; Place; Go-to; Open/close drawer	RGB	SL/RL	Kitchen tasks (Sim & Real)	Primitive & SL/RL



(a) Action score prediction for four kinds of primitives [22].



(b) Grasping success score prediction [24].

Fig. 2: Examples of action score prediction.

estimating the graspability not only in different x-y positions, but also in different grasping angles [24]. Another work proposed to first train a neural network to predict object classes and segmentation masks of a cluttered scene, and then train a Deep Q-Network (DQN) to predict the grasping success scores based only on the segmentation masks [25]. This work successfully transferred the learnt grasping score prediction system to the real world with domain randomisation. Recently, Mo *et al.* proposed to predict action scores for a set of six motion primitives based on RGB-D images or point clouds. They designed a three-branch network architecture to 1) predict the actionability of a pixel or a point, 2) propose gripper orientations and 3) estimate the success score of the primitive action given the action pixel and orientation [26]. In another interesting recent work [27], the authors proposed to represent the action possibilities of a large number of pretrained motion skills by the action value function in the RL framework based on RGB observations. These papers are closely related to the works in vision-based robotic grasping (VBRG), where many works were not linked to the concept of affordance. For a thorough review of VBRG, please refer to [38], [39].

C. Summary and limitations

To summarise, though some recent works tried to estimate action possibilities for a variety of actions, most of them focused on grasping tasks when deploying the learning system. These works leveraged motions that are generated by a motion planner or hand-crafted by humans. In terms of affordance learning, they sought to estimate whether a planned motion or primitive can be successfully performed at an image pixel location or a point in the point cloud. The learnt affordance model was used to infer a desired action by extracting a pixel location or a point that is centred at the affordable region or with the highest action possibility. There are several limitations regarding the papers discussed in this section.

1) At the current research stage, the community lacks an image segmentation dataset for object affordances at large scale [5], when compared to datasets like COCO [40] or ImageNet [41]. It is promising to build larger datasets, as demonstrated by the ImageNet dataset for image classification, though a vast amount of human labour is required. To reduce such human labour, self-supervised learning techniques could be employed, such as automatic labelling [21], [42] and interactive labelling [43].

2) Though multi-affordance detection has drawn researchers' attention [17], [18], real-world manipulation experiments using affordances are restricted to only one or two categories (mostly grasping) [15]–[23], [25]. Not much attention was given to other actions such as push and pull [24], [26]. In addition, they are subject to fully or partially hand-crafted motion primitives (e.g., top-down parallel-jaw grasping), thus are limited to a very small set of object-action relationships. For example, they cannot represent affordances for 6DoF (Degree of Freedom) grasping actions or non-primitive interactions. A recent work in coupling language instructions and mobile robot motion skills made a pioneering example on more complex action affordance learning and real-world grounding [27].

3) These methods only predict action possibilities, ignoring the knowledge about the effects of these actions. From a human perspective, we tend to use affordance knowledge for

planning, which requires us to be aware of not only what the possible actions are, but also what the results of these actions are. The next section elaborates on recent attempts to incorporate both action possibilities and effects.

4) These works exclude the dependencies between the executions of multiple actions and the influences of different manipulation objectives. For example, the possibilities of grasping a cup at its handle would differ when the robot is tasked to hang it up, place it on a table or hand it out to another agent. This involves a planning process for different final task objectives. We discuss more on this point in the next section.

IV. Keypoint Affordance

In the last section, we discuss papers that sought to first compute the action possibilities, $\hat{p}^{\mathcal{AF}}$, and then infer the afforded actions from the action possibilities, for example, computing a binary or continuous matrix that indicates whether a gripper can pick up an object at each pixel location of an RGB-D image. In these cases, a pixel in an image or a point in a point cloud is associated with an action as a parameter of a motion planner or a primitive.

In this section, we review works that proposed to generate the afforded actions by predicting object keypoints, skipping the computation process of the action possibility [28]–[33]. The keypoints were defined as the functional points of an object. They were associated with affordance because they could be used by some action inference methods (e.g., a motion planner) to generate afforded actions. Keypoints provide the action inference method with a smaller search space and easier-to-define task-relevant geometric constraints. From the RL perspective, the keypoints can be seen as an abstract observation that indicates the action space for a policy or value function, or as a constrained action space that corresponds to a set of affordable motion primitives. The latter one is adopted by many previous works. Previously, keypoint methods with non-deep learning techniques were limited to specific objects of a particular shape and size [3]. In this review, we focus on deep learning-based methods that are able to generalise to unseen and novel objects [28]–[33]. A summary of the observations, object types, training methods, deployment tasks and motion generation methods of these works are given in Table II.

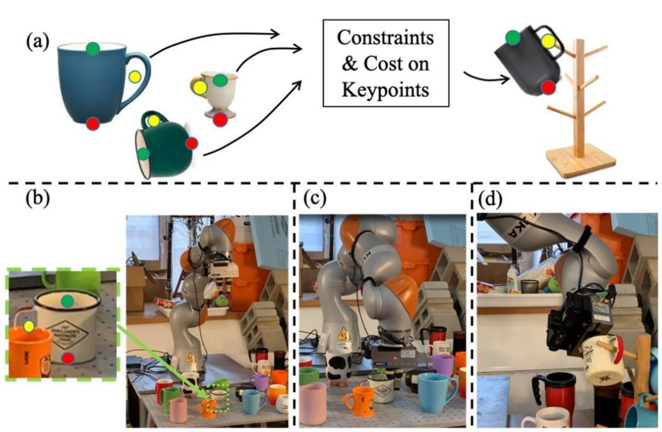


Fig. 3: Category-level keypoint detection from [28]. (a) Detected keypoints for different cups in planning; (b) keypoint detection; (c) grasping; (d) hanging.

Manuelli *et al.* proposed kPAM, which defined keypoints for objects that belong to the same category (Fig. 3) and supported grasping, placing and hanging actions to be inferred from the keypoints, for example, three keypoints at the handle, top and bottom for mugs. These keypoints were predicted given a segmented RGB-D image and then used by a motion planner to generate motions for picking and placing tasks. The authors later formulated a feedback control framework with the keypoint-based object and action representations, which accomplished a peg-in-hole insertion task with a variety of objects [29]. They also extended the method to include a shape completion technique, named kPAM-SC, so that the generated motions can handle object collision [30]. Another work, KETO, used a three-keypoint pattern, including a grasp point, a function point and an effect point, to represent hammer-like tools and infer hammering motions [31]. A generative network was trained to produce keypoint candidates given an object point cloud. An evaluation network was trained to predict the manipulation success scores for these keypoints. The training process was conducted in a self-supervised manner using task completion signals. These keypoints, along with a set of task keypoints within a simulation environment, were used to generate motions by solving a Quadratic Programming problem [31]. Turpin *et al.*, proposed GIFT [32], which predicted a set of representational keypoints for an object and then selected from them a grasping point and an interaction point. This procedure allowed the functional keypoint pattern to be discovered instead of being specified by users. They represented the functional keypoint proposal model as a Graph Neural Network (GNN) over the representational keypoints. They then computed a robot motion using model predictive control and evaluated the task-specific return for the motion. The functional keypoint proposal model was trained by optimising a REINFORCE loss with the task-specific return.

Instead of predicting keypoints for a category of objects as done in [28]–[32], Xu *et al.* proposed to define keypoints for afforded actions on images [33]. They modified the affordance image segmentation dataset UMD [14] by assigning a set of five 2D keypoints to each affordance region. These keypoints defined the position and direction information about the afforded actions. They proposed a two-branch deep neural network, AffKp, to learn affordance image segmentation and keypoint detection in parallel via supervised learning. The predicted keypoints were projected from the image plane to the real-world frame and used to infer the corresponding afforded actions.

Summary: To sum up, these works proposed to infer afforded actions that manipulate an object from a set of keypoints defined on the object. According to the affordance definition introduced in Section II, they are classified as methods that compute the afforded actions, rather than compute the action possibilities, for example, inferring various grasping configurations from a predicted grasping point on a tool handle [31] instead of a set of action possibilities [22]. Most of

TABLE II: Summary of papers focused on affordance keypoint prediction. **PCD:** point cloud data; **SL:** supervised learning; **SSL:** self-supervised learning; **Sim:** simulation **Real:** real-world; **DoF:** degree of freedom; **PJG:** parallel-jaw grasp; **MPC:** model predictive control.

Paper	Object classes	Affordance (afforded actions)	Input	Method	Deployment Task	Motion
[28]	Shoes; Mugs	6DoF PJG, place & hang	RGB-D	SL	Shoe-placing, mug-placing & mug-hanging (Real)	Planning
[29]	Erasers; Pegs; Holes	6DoF PJG, wipe, insert	RGB-D	SL	Whiteborad wiping, peg-in-hole insertion (Real)	Planning
[30]	Shoes; Mugs	6DoF PJG, place & hang	RGB-D	SL	Same as [28] with shape completion (Real)	Planning
[31]	Hammers	6DoF PJG, hammer, push, reach	PCD	SSL	Object hammering, pushing & reaching (Sim)	Planning
[32]	Hammers	4DoF PJG, hammer, push, hook	RGB-D	SSL	Object hooking, reaching, hammering (Sim)	MPC
[33]	UMD+GT dataset	UMD+GT dataset	RG-D	SL	PJG, pouring, arranging, cutting (Sim & Real)	Planning

the works leveraged human knowledge to create a pattern of keypoints and trained deep neural networks to predict them for a category of objects [28]–[31], [33], while only one work, GIFT, proposed to discover functional keypoints using task-completion signals [32]. The main benefits of using keypoints to infer afforded actions include but are not limited to:

- keypoints can capture the common properties of a category of objects;
- keypoints can support the inference of various afforded actions;
- keypoints can be used to reduce the search space of afforded actions for the action inference processes.

Limitations: The primary limitation of keypoint-based methods is that pre-defining a fixed pattern of keypoints requires a relatively large amount of human prior. This eases the keypoint prediction model from the difficulty of learning from scratch but limits the generalisability of the learnt keypoint patterns. In reality, one specific pattern of keypoints is unlikely to be sufficient and flexible enough for the diverse manipulation tasks that may need to be performed on the objects. The aforementioned papers have evaluated their methods on tasks with relatively simplified geometric constraints and manipulation skills [28]–[31], [33]. For example, when a robot could only reach a hammer's head, it could not grasp the head and use the handle as a hammering point if it can only recognise the head as a hammering point. Learning to predict keypoint patterns with free interactions and task-completion signals is promising for reducing such human biases [32].

Secondly, sparse keypoint representation is not very compatible with tasks that are sensitive to object shapes and sizes, when compared to a full point cloud representation. For example, when manipulating a deformable object like a soft plastic cup, keypoints are not enough for the robot to determine the grasping force and track the deformation of the cup [44]. In this regard, multi-modal representations may be required, such as using keypoints along with a shape-completion procedure [30]. In the future, other observation modalities, such as tactile sensors, force sensors, etc., may be incorporated with keypoints to better infer afforded actions in real-world manipulation tasks.

Last but not least, the primary method to infer afforded actions using keypoints, namely motion planning, is difficult and expensive in environments with complex dynamics and large action and state spaces. It poses two problems to classic methods: 1) user-specified dynamic models have difficulties representing highly stochastic and non-linear real-world systems and generalising to high-dimension inputs like images and 2) planning over large action and state spaces is very expensive and difficult. Researchers have proposed to address them by learning a system dynamic model from data [12], [45]–[48], though they did not explicitly consider the concept of affordances. We elaborate in the next subsection on recent works that propose to plan robot motions using a learnt affordance-aware dynamic model.

V. Modelling Action Possibilities and Effects

As defined in Section II, the effects of afforded actions can be modelled by a partial dynamic model $\hat{P}_I(s'|s,a)$, which predicts the next system states given a pair of state and *afforded* action. The motivation for building a dynamic model is to equip a robot with a safer and more efficient method to generate motion plans or learn from imagined data. A dynamic model releases the robot from expensive and potentially unsafe interactions with the real world [12], [48]. Previous works on action effect modelling have relied extensively on manually-abstracted state representations and dynamics [49], [50], which has a deep connection to the field of symbolic planning [51]. It is difficult, however, to handcraft dynamic models for real-world systems with complex observations. Therefore, in recent years researchers have proposed deep learning methods to represent and learn the dynamic model from data, demonstrating the value of having access to a dynamic model over the space of complex sensory observations [12], [48], [52], [53].

Among many recent advances of learnt world models, Khetarpal *et al.* proposed to integrate the concept of affordances in the model-based reinforcement learning (MRL) paradigm (as rephrased in section II) [10]. They first learnt a binary classification model to predict whether some actions were afforded given an observation, which was essentially estimating the action possibilities $\hat{p}^{\mathcal{AF}}$ as binary variables. Different from methods discussed in Section III, they did not infer the afforded actions from the estimated action possibilities. Rather, they proceeded to learn a dynamic model of the world for only actions that were classified as possible or effective. Data on non-effective actions are regarded as redundant and ignored. The resultant model was a partial dynamic model (PDM) of the system. During planning, the PDM is only queried for effective actions according to $\hat{p}^{\mathcal{AF}}$. In short, the benefits of such a framework are twofold: 1) it accelerates planning by only considering the afforded actions and 2) it accelerates dynamic model learning by focusing on learning part of the system dynamics concerning the afforded actions of interests. They were demonstrated first in a continuous 2D navigation task in [10] and later in unseen long horizon manipulation tasks in simulation with image inputs (Fig. 4) [34]. This affordance-aware model-based reinforcement learning framework was later extended to develop temporally abstract partial dynamic models, considering options (sub-policies) that are only afforded in certain situations. The authors empirically demonstrated the success of learning option affordances and partial option models online, resulting in more efficient learning and planning in a 2D Taxi task [35].

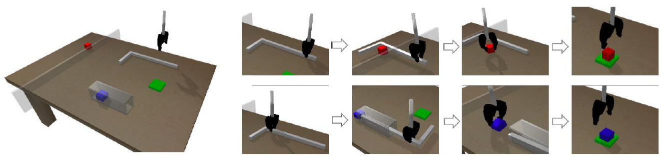


Fig. 4: The multi-step tool-use task designed to evaluate the Deep Affordance Foresight method proposed in [34]. The robot needs to decide which end of the L-shape stick to grasp for reaching the red block or push the blue block out of the tube.

Limitations: As a relatively new direction, the first limitation is the lack of evaluation in more realistic examples. Most previous works are performed in simulation using synthetic data. Tasks with image or point cloud observations from real robots with longer time horizons would increase the complexity considerably. More efforts are required to design more realistic tasks.

Secondly, the predicted action effects in the proposed examples are more short-term or instant effects of single-step action commands. In practice, planning is often more valuable with macro actions that consist of a series of single-step control commands, exhibiting a particular kind of skill, such as pushing for a certain distance, approaching and grasping an object, lifting up for a certain height, etc. This requires the algorithm to reason about long-term action possibilities and consequences. Though an attempt was made to incorporate affordances with temporally abstract partial models for more efficient planning at a more abstract level, it was only evaluated in a 2D Taxi task [35]. More effort is needed to evaluate and improve its performance on robotic tasks in the future.

Thirdly, the proposed method focuses on the affordances of a given state, which is likely to be computationally inefficient for tasks with complex observations containing diverse information irrelevant to the manipulation goal. From a human perspective, we typically only attend to some parts of the observation that are most relevant to the task of interest, saving energy and improving planning efficiency and accuracy.

VI. Discussions and Challenges

According to the reviewed papers, this section summarises the limitations of deep robotic affordance learning (DRAL) and identifies its bottlenecks at the current stage. We conduct the discussion and pose future research challenges from the following angles: observations, actions, affordance representations, data collection and real-world deployment.

A. Observation

For most tasks, especially real-world tasks, a robot relies on sensors to perceive the environment *without access to the true system dynamics,* such as the velocities of objects. This is one of the most common assumptions adopted by robotic researchers. Previous works have made efforts to develop symbolic representations for the observations of the system to simplify the mapping from sensory observations to affordances [4], [54]. In recent DRAL literature, the types of observations have become more complex, including object states (normally in simulation), object point clouds, and RGB/RGB-D images.

Another important assumption made by these works is that *the observation contains enough information to reason about affordance*. However, this does not always hold true. For example, a heated plate may be detected as graspable from RGB-D or point cloud observations though it may be actually too hot to hold by a human. Some affordances may require information about temperature, softness, transparency, reflection, etc., that is difficult for (depth) cameras to capture. It is also worth noting that languages are becoming more popular to provide instructions or extra information about the desired tasks and skills for affordance learning [27], [55], due to the rise of large language models (LLMs). Information about the robot itself, such as sensorimotor states, could also help to reason about affordances like reachability. On the other hand, affordances of occluded objects are difficult to detect from a fixed camera viewpoint. Combining all these, a promising direction for future research is to apply multi-modal and multi-viewpoint observations for affordance detection [54], [56].

The third assumption about observations, especially for deep learning-based methods, is that *the mapping from inputs to actions or action possibilities can be found through gradient descent*. However, given the large space of observations in the real world, it is very challenging to find such a mapping even if it does exist. Some works applied pre-processing methods to help the robot focus on the most relevant information for affordance learning or action inference, such as applying object masks [25] or extracting object keypoints [28]. Such ideas make computation more efficient by shrinking the size of observation space, whereas more or less lose some degree of generality due to human priors. In this regard, future research could focus on representation design or learning, giving special attention to the trade-off between generalisability and learning efficiency (or computational cost) for affordance detection or afforded action inference.

B. Action

Noticeably, researchers preferred motion primitives in recent DRAL works, for example, grasping primitives that move a gripper towards an identified grasping location and close the fingers [24], [25], and placing primitives that move a gripper with an object to a location and release the fingers [28], [30]. Note that these primitives can be motion planned by a planner [24], [28], [30], [31] or parameterised motor skills [34]. These primitives exhibit relatively simple motions, such as pick-and-lift [18], [19], [23]–[25], pick-and-place [22], [28], [30], pushing [34] and hammering [31]–[33]. The use of motion primitives as actions exhibits a trend that the community is more interested in the affordances of high-level skills, rather than low-level control commands. To follow this trend, we pose some challenges and future directions to consider.

The adaptability of the primitive motions considered by recent works could be improved, as they were mostly designed for open-loop control. For example, given a grasping point, a grasping motion moves the gripper to the grasping point and closes the fingers, without any adaptation in between. However, the detected grasping affordance may be inaccurate or changed during the execution of the motion due to occlusion, human factors, collision with the robot arm or finger slippery, etc. To cope with such challenges, one may consider a feedback control style method for action inference [29], [32]. Another interesting direction to consider is an algorithm that is permitted to stop and re-select motion primitives. For example, when an insertion motion changes from affordable to unaffordable, the robot may select a re-position motion without waiting for the insertion motion to reach its execution time limit. The notion of *interrupted options* based on the option framework [36] may serve as a good theoretic foundation.

Predefined primitive motions are very useful when the manipulation task is in a rather structured environment without unexpected factors. However, the real world is highly unstructured and uncertain. A robot needs to generalise its skills to novel situations quickly or sometimes finds new skills to manipulate an object. This means the robot may be required to discover new afforded actions. To achieve this, the action space needs to be general enough. One promising direction is the study of option or subgoal discovery in hierarchical reinforcement learning [13], in which skills (in the form of sub-policies) are discovered instead of predefined.

C. Affordance Representations

According to Gibson [1], perceiving affordance does not need information processing or any internal representations, but only requires the extraction of fundamental physical properties of the target object or environment. For example, perceiving that a needle has a pointed end leads to the perception that the needle affords to pierce. This reasoning is theoretically sound [57], but is however practically limited as, in practice, some forms of mathematical representations of affordances are required to facilitate action inferences [6]. Also, it is important to note that there are so far no known widely-adopted benchmarking metrics for qualitative or quantitative comparative studies of different representations proposed in the field. What intermediate representations are needed in the spectrum between end-to-end learning and manually constructing everything is mostly specific to the problem of interest.

As this review is inclined to the recent practical applications of affordances in DL-powered RL and robotics, it is more graspable and plausible from the practical standpoint to discuss the representations of affordances in recent literature according to *how the action inference method works*. Afforded actions are inferred in mainly three manners: 1) from the action possibility estimates, 2) by a direct mapping from the observations, and 3) by planning with a partial dynamic model. The first and third classes require an explicit representation of the action possibilities and effects, while the second one may need an intermediate representation that constrains the action space, such as object keypoints.

Action possibilities for primitive motions were often represented by an *affordance map*, which is typically a matrix that has the same size as the observation image. Its entries indicate the success rates or possibilities of executing certain primitive motions at the corresponding pixel locations [22]–[25]. Segmentation masks can be regarded as a special case with binary variables [14], [17]–[20]. It can also be applied to point clouds in the 3D space [15], [26]. This representation is efficient as it estimates the possibilities for a set of actions simultaneously, but is limited to primitive motions that operate over the discrete image pixels or object points. It may not easily generalise to continuous observations such as sensorimotor states, force feedback, etc. For actions that are not parameterised on images or point clouds, one may need to represent the action possibilities as a classifier [10]. In order to scale to real-world tasks, it is promising to develop methods to accelerate the learning of the action possibility estimator with large and continuous action space, such as learning from demonstrations [58].

Representing and predicting the effects of actions is another difficult topic. Though an action possibility estimator helps to reduce the learning data requirement and increase the planning efficiency for dynamic models [10], [34], the difficulty of reconstructing high dimensional observations (e.g., images or point clouds) remains. Experiences and methods from other fields could be considered, such as video prediction [59]. There is also a large body of work devoted to the learning of dynamic models [48]. Abstract representation for system observations is another closely related topic [60]. Future research may focus on applying general dynamic model learning methods to partial dynamic models with an action possibility estimator. Another challenge in the long term may be how the learning of affordances affects the learnt representation of the world, which is related to the topic of understanding the world through interaction.

Another way to compute afforded actions in the literature is through a direct mapping from observations to a set of afforded actions. The crucial question is how to represent the scene/object in a way that relates to their afforded actions. One popular solution is to use object keypoints that geometrically capture some functions of a category of objects, such as grasping points of mugs [28]–[33], as discussed in Section IV. From the keypoint methods, we can identify some criteria to be satisfied when considering other types of representations. These include: 1) intuitive or convenient for generating robot motions; 2) able to generalise across robot hardware (grippers, arms, etc.); 3) able to capture the common properties of many objects. Notice that such a representation should be designed as an abstraction of the observations of a scene or an object that relates to the afforded actions. The keypoint-based methods rely on motion planning or model predictive control to generate the desired motions (see Table II), while one may come out with representations that suit other motion generation techniques (e.g., reinforcement learning, imitation learning, etc.).

D. Data collection

Deep learning methods require a considerable amount of data to achieve good generalisation performances [61]. Previous papers in DRAL have used supervised learning, self-supervised learning and reinforcement learning as their core training methods, each of which has a unique data collection process.

Supervised learning methods rely fully on human prior to collect and generate data, which is expensive for large datasets (e.g., ImageNet [41]). Most papers use the UMD dataset [14] for evaluation. However, it only provides segmentation labels. To alleviate the difficulty of collecting manipulation-specific data (e.g., grasping points, motion trajectories, etc.), some papers adopt self-supervised learning to collect data automatically through simulations [23]–[25], [31], [32]. Reinforcement learning (RL)-based methods generate training data by interacting with the environment using a learnt policy with some degree of randomness [11]. In addition, the performance of the RL policy is evaluated directly on task return or success rate, without intermediate metrics (e.g., the accuracy of predicting segmentation masks or keypoints). However, off-policy RL methods can benefit from data generated from other sources, such as human demonstrations [58].

A limitation, at the current stage, is the lack of a consensus on which benchmark should be used to generate the data and evaluate the algorithms for DRAL. Ideally, such a benchmark should provide handy Application Programming Interfaces (APIs) and functions to support the data collection processes for supervised, self-supervised and reinforcement learning. Common functionalities, such as capturing RGB/RGB-D images and point clouds, classic planning algorithms, popular RL baselines, etc. are also considered helpful. It could be more valuable if tasks that feature multiple manipulation objectives and multi-step manipulation are designed and built-in. There are several open-source datasets, simulation environments and benchmarks that may be extended for such purposes [26], [62]–[64]. The community has not yet seen a large-scale dataset for DRAL that covers the mentioned aspects.

E. Real-world deployment

For methods that use real-world data, the main difficulty is primarily the expensive data-collection process, which was covered in the last subsection. The main concern that arises during the final deployment or evaluation is then the insufficient generalisation ability, which is largely caused by the limited amount of training data.

1) *Supervised learning* methods are easier to be deployed in the real world after being trained, though their performances rely extensively on the quality of the dataset. In the past few years, many datasets that support the learning of stable grasping have been constructed [14], [22], [38], [39], [65], [66]. However, very few are built for multiple manipulation objectives or multi-step tasks [67], [68]. Consequently, more efforts are needed to collect data that cover diverse background textures, viewpoints, objects (in terms of types, shapes, dimensions, etc), and manipulation skills (trajectories) in order for supervised learning-based DRAL to work in the real world.

2) *Reinforcement learning* in the real world is even more difficult due to the high risk of hardware damages during exploration and a considerable amount of human labour for resetting the environment [69].

3) *Sim2real transfer* is another stepping stone for successful real-world deployment, as researchers have resolved the simulation training to avoid the painful and impractical data-collection process in the real world. Inevitably, deploying models trained in simulation onto real-world systems will have to face the simulation-to-reality gap. To cope with such differences, researchers have proposed to use domain randomisation to extend the distribution of training data [70]. It can be applied to image textures [20], [25], [70], [71], camera parameters [72] and physical properties [69]. Recent DRAL works limit their real-world applications within a relatively unchanged and structured environment. Long-horizon tasks that require the reasoning of the long-term effects of diverse skills or objects have mainly been studied in simulation. More efforts are needed to evaluate and adapt existing methods to real-world data.

VII. Conclusion

This review paper looks into the recent advances in the topic of deep robotic affordance learning (DRAL). DRAL aims to develop data-driven (deep learning) approaches to apply the concept of affordance to robotic tasks. We suggest in this review to summarise and analyse these works based on the reinforcement learning (RL)-based definition of affordances [10]. We briefly recall this definition in Section II, where we classify recent DRAL papers and discuss the connections between RL and affordances. Accordingly, they are categorised into three classes of works:

- 1) infer afforded actions from the estimated action possibilities;
- 2) learn an abstract object/scene representation that relates to the set of afforded actions;
- 3) generate afforded actions through planning with a learnt partial dynamic model and an action possibility classifier.

Advances and limitations of the three lines of works are discussed in Sections III, IV and V, respectively. A more general discussion of the field and its challenges are given in Section VI.

Final remark: We further propose here a promising direction to extend the RL-based affordance definition. In [10], the intent captures the desired resultant state of an action taken at a system state. Subsequently, the corresponding affordance is defined as a subset of state and action pairs, in which the intent is satisfied. In [35], the definitions of intent and affordance are extended to include multiple timestep predictions in the MDPs. Here, we propose to extend the theory by generalising the definition of intent to capture *an arbitrary kind of consequence* of an action taken at a state, generalising beyond state prediction.

Such intents could be called *general intent*. For example, the intent of a grasping action may include the desired success rate, object dropping rate, the weight of the object that can be held, etc. Subsequently, affordance is defined to include a subset of state and action pairs, in which the intent is satisfied. Such affordances may be called *general affordances*.

More importantly, this direction is promising, if a thorough mathematical definition is developed based on the RL framework. A set of new algorithms can be developed to infer actions according to the predictions of arbitrary action consequences, instead of simple system states. Similar to the dynamics-based affordances, general affordances can help with exploration, value function or policy learning, model learning and planning by constraining the action space, but for arbitrary action consequences, beyond state prediction. However, this is outside of the scope of this review, and considerably more future efforts are required to derive and experiment with the theory.

VIII. Acknowledgement

Xintong Yang thanks the Chinese Scholarship Council (CSC) for providing the living stipend for his PhD programme (No. 201908440400). This work was partially supported by the Engineering and Physical Sciences Research Council (grant No. EP/X018962/1).

References

[1] E. J. Gibson and W. Collins, "The concept of affordances in development: The renascence of functionalism," in *The concept of development: The Minnesota symposia on child psychology. Vol*, vol. 15, 1982, pp. 55–81.

[2] J. M. Good, "The affordances for social psychology of the ecological approach to social knowing," *Theory & Psychology*, vol. 17, no. 2, pp. 265–295, 2007.

[3] L. Jamone, E. Ugur, A. Cangelosi, L. Fadiga, A. Bernardino, J. Piater, and J. Santos-Victor, "Affordances in psychology, neuroscience, and robotics: A survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 1, pp. 4–25, 2016.

[4] N. Yamanobe, W. Wan, I. G. Ramirez-Alpizar, D. Petit, T. Tsuji, S. Akizuki, M. Hashimoto, K. Nagata, and K. Harada, "A brief review of affordance in robotic manipulation research," *Advanced Robotics*, vol. 31, no. 19-20, pp. 1086–1101, 2017.

[5] P. Ardón, È. Pairet, K. S. Lohan, S. Ramamoorthy, and R. Petrick, "Building affordance relations for robotic agents-a review," 2021.

[6] M. Hassanin, S. Khan, and M. Tahtali, "Visual affordance and function understanding: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–35, 2021.

[7] D. Abel, G. Barth-Maron, J. MacGlashan, and S. Tellex, "Toward affordance-aware planning," in *First Workshop on Affordances: Affordances in Vision for Cognitive Robotics*, 2014.

[8] F. Cruz, S. Magg, C. Weber, and S. Wermter, "Training agents with interactive reinforcement learning and contextual affordances," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 4, pp. 271–284, 2016.

[9] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[10] K. Khetarpal, Z. Ahmed, G. Comanici, D. Abel, and D. Precup, "What can i do here? a theory of affordances in reinforcement learning," in *International Conference on Machine Learning*, PMLR, 2020, pp. 5243–5253.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[12] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153–173, 2017.

[13] M. Wulfmeier, D. Rao, R. Hafner, T. Lampe, A. Abdolmaleki, T. Hertweck, M. Neunert, D. Tirumala, N. Siegel, N. Heess, *et al.*, "Data-efficient hindsight off-policy option learning," in *ICML*, PMLR, 2021, pp. 11 340–11 350.

[14] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 1374–1381.

[15] M. Kokic, J. A. Stork, J. A. Haustein, and D. Kragic, "Affordance detection for task-specific grasping using deep learning," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, IEEE, 2017, pp. 91–98.

[16] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Object-based affordances detection with convolutional neural networks and dense conditional random fields," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 5908–5915.

[17] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 5882–5889.

[18] F.-J. Chu, R. Xu, L. Seguin, and P. A. Vela, "Toward affordance detection and ranking on novel objects for real-world robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4070–4077, 2019.

[19] P. Mandikal and K. Grauman, "Learning dexterous grasping with object-centric visual affordances," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[20] A. Hämäläinen, K. Arndt, A. Ghadirzadeh, and V. Kyrki, "Affordance learning for end-to-end visuomotor robot control," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 1781–1788.

[21] J. Borja-Diaz, O. Mees, G. Kalweit, L. Hermann, J. Boedecker, and W. Burgard, "Affordance learning

from play for sample-efficient policy learning," *arXiv preprint arXiv:2203.00352*, 2022.

[22] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *2018 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2018, pp. 3750–3757.

[23] J. Cai, H. Cheng, Z. Zhang, and J. Su, "Metagrasp: Data efficient grasping by affordance interpreter network," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 4960–4966.

[24] H. Wu, Z. Zhang, H. Cheng, K. Yang, J. Liu, and Z. Guo, "Learning affordance space in physical world for vision-based robotic object manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 4652–4658.

[25] S. Yang, W. Zhang, R. Song, J. Cheng, and Y. Li, "Learning multi-object dense descriptor for autonomous goal-conditioned grasping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4109–4116, 2021.

[26] K. Mo, L. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," *arXiv preprint arXiv:2101.02692*, 2021.

[27] A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on Robot Learning*, PMLR, 2023, pp. 287–318.

[28] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "Kpam: Keypoint affordances for category-level robotic manipulation," in *Proceedings of the 17th international symposium on Robotics Research*, 2019.

[29] W. Gao and R. Tedrake, "Kpam 2.0: Feedback control for category-level robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2962–2969, 2021.

[30] ——, "Kpam-sc: Generalizable manipulation planning using keypoint affordance and shape completion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 6527–6533.

[31] Z. Qin, K. Fang, Y. Zhu, L. Fei-Fei, and S. Savarese, "Keto: Learning keypoint representations for tool manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 7278–7285.

[32] D. Turpin, L. Wang, S. Tsogkas, S. Dickinson, and A. Garg, "Gift: Generalizable interaction-aware functional tool affordances without labels," in *Robotics: science and systems*, 2021.

[33] R. Xu, F.-J. Chu, C. Tang, W. Liu, and P. A. Vela, "An affordance keypoint detection network for robot manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2870–2877, 2021.

[34] D. Xu, A. Mandlekar, R. Martín-Martín, Y. Zhu, S. Savarese, and L. Fei-Fei, "Deep affordance foresight: Planning through what can be done in the future," in *NeurIPS 2020*, PMLR, 2021.

[35] K. Khetarpal, Z. Ahmed, G. Comanici, and D. Precup, "Temporally abstract partial models," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[36] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[37] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. arXiv: `http://arxiv.org/abs/1312.6114v10 [stat.ML]`.

[38] G. Du, K. Wang, S. Lian, and K. Zhao, "Vision-based robotic grasping from object localization, object pose estimation to grasp estimation for parallel grippers: A review," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1677–1734, 2021.

[39] Q. M. Marwan, S. C. Chua, and L. C. Kwek, "Comprehensive review on reaching and grasping of objects in robotics," *Robotica*, pp. 1–34, 2021.

[40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, Springer, 2014, pp. 740–755.

[41] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[42] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 22 243–22 255. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/fcbc95ccdd551da181207c0c1400c655-Paper.pdf.

[43] M. CHEGIN, J. Bernard, J. Cui, F. Chegini, A. Sourin, K. Keith, and T. Schreck, "Interactive visual labelling versus active learning: An experimental comparison," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 4, pp. 524–535, 2020.

[44] V. E. Arriola-Rios, P. Guler, F. Ficuciello, D. Kragic, B. Siciliano, and J. L. Wyatt, "Modeling of deformable objects for robotic manipulation: A tutorial and review," *Frontiers in Robotics and AI*, vol. 7, p. 82, 2020.

[45] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International Conference on Machine Learning*, PMLR, 2019, pp. 2555–2565.

[46] S. Eiffert, H. Kong, N. Pirmarzdashti, and S. Sukkarieh, "Path planning in dynamic environments using generative rnns and monte carlo tree search," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 10 263–10 269.

[47] A. H. Qureshi, Y. Miao, A. Simeonov, and M. C. Yip, "Motion planning networks: Bridging the gap between learning-based and classical motion planners," *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 48–66, 2020.

[48] T. M. Moerland, J. Broekens, and C. M. Jonker, "Model-based reinforcement learning: A survey," *arXiv preprint arXiv:2006.16712*, 2020.

[49] E. Şahin, M. Cakmak, M. R. Doğar, E. Uğur, and G. Üçoluk, "To afford or not to afford: A new formalization of affordances toward affordance-based robot control," *Adaptive Behavior*, vol. 15, no. 4, pp. 447–472, 2007.

[50] E. Ugur and J. Piater, "Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 2627–2633.

[51] E. Karpas and D. Magazzeni, "Automated planning for robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 417–439, 2020.

[52] C. Finn and S. Levine, "Deep visual foresight for planning robot motion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 2786–2793.

[53] A. Ahmetoglu, M. Y. Seker, J. Piater, E. Oztop, and E. Ugur, "Deepsym: Deep symbol generation and rule learning from unsupervised continuous robot interaction for planning," *Journal of Artificial Intelligence Research*, vol. 75, pp. 709–745, 2022.

[54] F. Cruz, G. I. Parisi, and S. Wermter, "Multi-modal feedback for affordance-driven interactive reinforcement learning," in *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2018, pp. 1–8.

[55] Y. Jiang, S. S. Gu, K. P. Murphy, and C. Finn, "Language as an abstraction for hierarchical deep reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[56] M. A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Learning multimodal representations for contact-rich tasks," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 582–596, 2020.

[57] R. A. Brooks, "Intelligence without representation," *Artificial intelligence*, vol. 47, no. 1-3, pp. 139–159, 1991.

[58] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 6292–6299.

[59] B. Wu, S. Nair, R. Martin-Martin, L. Fei-Fei, and C. Finn, "Greedy hierarchical variational autoencoders for large-scale video prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2318–2328.

[60] D. Abel, "A theory of state abstraction for reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9876–9877.

[61] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[62] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[63] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi, *et al.*, "Rearrangement: A challenge for embodied ai," *arXiv preprint arXiv:2011.01975*, 2020.

[64] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, "Gpu-accelerated robotic simulation for distributed reinforcement learning," in *Conference on Robot Learning*, PMLR, 2018, pp. 270–282.

[65] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 444–11 453.

[66] D. Morrison, P. Corke, and J. Leitner, "Egad! an evolved grasping analysis dataset for diversity and reproducibility in robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4368–4375, 2020.

[67] Y. Huang and Y. Sun, "A dataset of daily interactive manipulation," *The International Journal of Robotics Research*, vol. 38, no. 8, pp. 879–886, 2019.

[68] A. Mandlekar, J. Booher, M. Spero, A. Tung, A. Gupta, Y. Zhu, A. Garg, S. Savarese, and L. Fei-Fei, "Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pp. 1048–1055.

[69] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: Lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021.

[70] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30. DOI: 10.1109/IROS.2017.8202133.

[71] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 627–12 637.

[72] X. Ren, J. Luo, E. Solowjow, J. A. Ojea, A. Gupta, A. Tamar, and P. Abbeel, "Domain randomization for active pose estimation," in *2019 International Confer-*

ence on Robotics and Automation (ICRA), IEEE, 2019, pp. 7228–7234.