4th International Conference on
Development and Learning and on Epigenetic Robotics
October 13-16, 2014. Palazzo Ducale, Genoa, Italy

Emergent Structuring of Interdependent Affordance Learning Tasks

Emre Ugur and Justus Piater

Intelligent and Interactive Systems, Institute of Computer Science,
University of Innsbruck

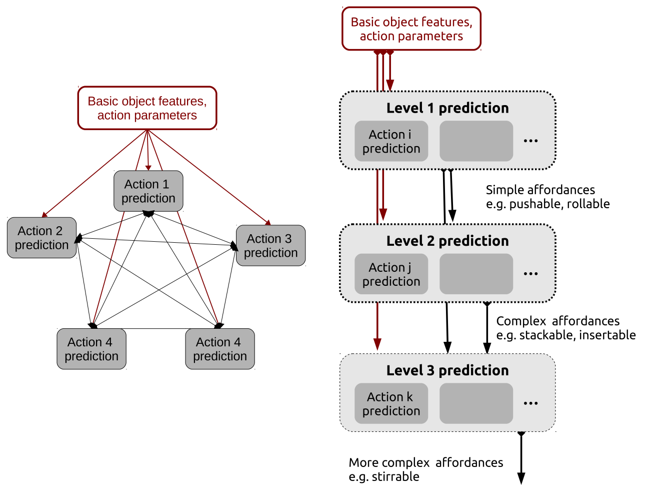
Abstract— **In this paper, we study the learning mechanisms that facilitate autonomous discovery of an effective affordance prediction structure with multiple actions of different levels of complexity. A robot can benefit from a hierarchical structure where pre-learned basic affordances are used as inputs to bootstrap learning of complex affordances. In a developmental setting, links from basic affordances to the related complex affordances should be self-discovered by the robot, along with a suitable learning order. In order to discover the developmental order, we use Intrinsic Motivation approach that can guide the robot to explore the actions it should execute in order to maximize the learning progress. During this learning, the robot also discovers the structure by discovering and using the most distinctive object features for predicting affordances. We implemented our method in an online learning setup, and tested it in a real dataset that includes 83 objects and the discrete effects (such as pushed, rolled, inserted) created by three poke and one stack action. The results show that the hierarchical structure and the development order emerged from the learning dynamics that is guided by Intrinsic Motivation mechanisms and distinctive feature selection approach.**

I. Introduction

Studies with infant chimpanzees[1] and human infants[2] revealed that there is a dramatic increase in exploration and success of object-object combinatory actions at around 1.5 years of age while such actions were at a very low frequency before that period. This data suggests that the infants first develop basic skills and affordances that are precursors of combinatory manipulation actions. They also probably use the learned action grounded object properties in further development of complex action affordances.

In learning complex action affordances, i.e. affordances that are provided by pairs of objects, we proposed a learning framework where a developmental robotic system learns object affordances[1] in two-stages [3]. In the first stage, the robot learns predicting single-object affordances (such as pushability and rollability) by pushing single objects in different directions, and learning the relations between visual object features and the created discrete effects. In the second stage, these single-object affordance predictions, i.e. effects predicted to be obtained by the single-object actions, were used along with other object features to learn paired-object affordances in stacking task. In this context, we showed how complex affordance learning was bootstrapped by using pre-learned basic-affordances encoded as additional features. While such an approach was effective in efficient learning of increasingly more complex affordances, the development order and hierarchical prediction structure was manually designed based on the pre-defined complexity levels of actions and affordances. A truly developmental system, on the other hand, should be able to self-discover such a structure (see Fig. 1(b)), i.e. links from basic to related complex affordances, along with a suitable learning order.

[1]In this study, the affordances provided by an object is defined as the list of discrete effects (e.g. pushed, rolled, inserted) predicted to be obtained by the discrete actions such as 'poke a single object' and 'stack a pair of objects'. Learning affordances refers to building a multi-category classifier for each action that predicts the effect of that action given continuous visual features and other predicted affordances of the object(s) involved. The discrete actions and discrete effects are assumed to be discovered before.



(a) Flat prediction (b) Hierarchical prediction

Fig. 1. (a) shows a flat affordance learning structure, where the affordances are predicted based on low level object features, action parameters, and all other perceived affordances. (b) shows a simple hierarchical structure where simple affordance predictions can be used to detect complex affordances. This paper aims automatic discovery of such a hierarchical structure along with the corresponding development order of its components.

E. J. Gibson argued that learning affordances is neither the construction of representations from smaller pieces, nor the association of a response to a stimulus. Instead, she claimed, learning is "discovering distinctive features and invariant properties of things and events" [4]. Learning is not "enriching the input" but discovering the critical perceptual information in that input. We will argue that learning and prediction based on the most distinctive features not only provide perceptual economy (as in [5]), but can be used to autonomously determine the structure of the learning problem.

Affordance learning through exploration requires the embodied agent to deal with the high-dimensional nature of the sensorimotor development in an open-ended learning framework. Intrinsic Motivation approach, which can be regarded as a set of active learning mechanisms for developmental robots, enables efficient and effective learning in such environments by guiding the robot learning with intelligent exploration strategies[6]. Intrinsic Motivation (IM) approach in developmental robots [7] was inspired from curiosity based motivation mechanisms in human development, and has recently been effectively applied to cognitive robots where object knowledge is developed through self-exploration and social guidance [8]. This approach adaptively partitions agent's sensorimotor space into regions of exploration and guides the agent to select the regions that are in intermediate level of difficulty. This is achieved by maximizing reduction in prediction error, in other words by maximizing the learning progress. In this paper, we propose to use this approach to guide the robot to explore different affordances by adaptively selecting the actions to execute, and updating the models of the affordance predictions based on the results of these actions. Through IM approach, we aim to achieve a developmental progression similar to those of infants in learning simple-to-complex skills and affordances.

In summary, we study the mechanisms that enable autonomous structuring of affordance learning problem along with development order of its components. Our prediction system starts in a flat form as shown in Fig. 1(a) with no assumption on the relative complexity of actions and predictions. In each learning step, the robot actively selects the most "interesting" action to explore based on Intrinsic Motivation[9], and updates the prediction model of the corresponding action based on the observed effect. The robot also distinguishes "the most distinctive features" for prediction of each different affordance in order to "discover the information that specifies an affordance"[10] in training the prediction model. We expect these two mechanisms, namely (i) the Intrinsic Motivation based selection of actions to explore, and (ii) the use of the most distinctive features in affordance predictions, enable emergence of a hierarchical structure, similar to the one shown in Fig. 1(b) along with the corresponding developmental order of its components.

II. Active Learning of Affordances with Distinctive Features

This section gives the outline of the online learning algorithm. In our scenario, the robot needs to interact with the objects using its action repertoire in order to learn their affordances. Learning affordances corresponds to training a classifier for each action that predicts the effect of that action given object features. Thus, in each learning episode, the robot selects an action, executes this action on a number of objects, observes the effects created on these objects, and updates the predictor of the explored action with the acquired experience.

Algorithm 1 gives the online learning outline. At line 1, visual object features are computed for the objects observed in the environment. Next, predictors and their learning-progresses are initialized with an initial phase of random exploration which correspond to 10 interactions for each action. The first step of the main loop (line 4) is to select the next action to explore with the highest learning progress based on Intrinsic Motivation criteria (see Section II-C). Next, a number of objects are selected for exploration by this action (Section II-D). The selected action is executed on each selected object and the effects generated by these executions are observed (line 6). Based on the observed effects, the predictor of the executed action, along with its learning progress, is updated (lines 7 and 8). The most distinctive features used for predicting the effect of this action are also updated by finding the relevant features of the updated predictor at the same step (Section II-B). Finally, the effect predictions for all objects are updated. Note that we described the algorithm with single-object actions in order to provide a clear overall picture, thus omitted several details.

Algorithm 1 Active learning of affordances with distinctive features

```
 1: compute object features
 2: initialize predictors and affordance predictions
 3: for each online learning time-step do
 4:     select action based on Intrinsic Motivation
 5:     select objects to explore
 6:     execute the selected action on the objects and observe effects
 7:     update the effect predictor of the selected action
 8:     find the most distinctive features for the updated predictor
 9:     update learning progress of the selected action
10:     update the effect predictions for all objects for the selected
        action
11: end for
```

A. Learning of affordances

Learning of affordances corresponds to learning the relations between objects, actions and effects [11]. In this study, object affordances are encoded as the list of effects achievable by executing different actions of the robot:

$$\mathit{affordances}^o = (\varepsilon^o_{a_1}, \varepsilon^o_{a_2}, ...)$$

where $\varepsilon^o_{a_1}$ is the discrete effect created on object o by action a_1.

Predicting the effect of each action is learned by executing the corresponding action on different objects. The resulting effect of one action depends on various features of objects, and is related to the other affordances the object provide. For example, stackability affordance can be related to rollability affordance and some other object features such as the object sizes. Here, object features corresponds to general-purpose basic ones computed mostly from visual perception with no explicit link to robot's actions. These may include standard features used in literature, related to size and shape properties of the objects. On the other hand, as we defined above, affordances encode object-action interaction dynamics for the available actions.

In order to learn affordances, and acquire the ability to predict action effect based on object features and affordances, the following classifier ($Pred$) is trained for each action. Specifically, we use Support Vector Machine (SVM) classifiers with Radial Basis Function (RBF) kernel and optimized parameters to learn these predictors[12]. The multi-category classifier, after training, can predict the effect category given features and affordances as follows:

$$\varepsilon^o_{a_i} = Pred_{a_i}(\mathit{features}^o, \mathit{affordances}^o \backslash \varepsilon^o_{a_i})$$

Here, $\mathit{affordances} \backslash \varepsilon^o_{a_i}$ denotes 'other affordances', i.e. the effect predictions of other classifiers. The general input/output structure is provided in Fig. 2.

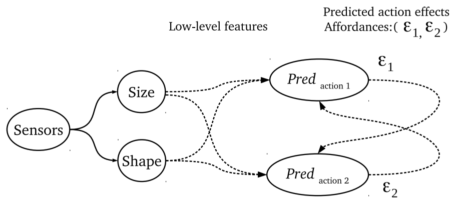


Fig. 2. Input/output links of the affordance predictors.

The recurrent connections in Fig. 2 might seem counter-intuitive in a non-dynamical system, where both of the predictors expect input from each other. This is achieved by keeping the predictions for all the objects in memory, and using the input values from the memory. Before training, the predictions for all objects are fixed to non-existing effect categories (-1). When a predictor is updated with exploration and learning, its predicted effects on all objects are updated in the memory as well. However these updated predictions are not propagated to other predictors immediately; they are used only by a new predictor that is being updated in the next time-step. In this way, we avoided potential instability issues of interdependent predictors.

Finally, this prediction mechanism can be generalized to actions that involve more than one object, by simply including all object features and affordances as the input attributes of the predictors. In this paper, we indeed use an action that involves two objects, where the predictor takes the following form:

$$\begin{aligned} \varepsilon^{(o_1,o_2)}_{a_i} = Pred_{a_i}(&\mathit{features}^{o_1}, \mathit{features}^{o_2}, \\ &\mathit{affordances}^{(o_1,o_2)} \backslash \varepsilon^{(o_1,o_2)}_{a_i}) \end{aligned} \qquad (1)$$

Paired-object affordances, i.e. affordances offered by the corresponding two objects, correspond to collection of the effects (expected to be) obtained by the execution of all available actions:

$$\mathit{affordances}^{(o_1,o_2)} = (\varepsilon^{o_1}_{a_1}, \varepsilon^{o_1}_{a_2}, ...\varepsilon^{o_2}_{a_1}, \varepsilon^{o_2}_{a_2}, ..., \varepsilon^{(o_1,o_2)}_{a_i}, \varepsilon^{(o_1,o_2)}_{a_j})$$

B. Discovering the most distinctive features

The most distinctive features that specify an affordance correspond to the minimal set of inputs of the corresponding effect predictor with the maximum achievable prediction accuracy. We used *Sequentialfs* (sequential features selection) method to select these features. The Sequentialfs method generates near-optimal relevant feature sets in a way similar to the one used in Schemata Search[13]. Starting from an empty relevance feature set, it selects one feature and adds it to the feature set of previous iteration. At each iteration, a candidate feature set for each not-yet-selected feature is formed by adding the corresponding feature to the previous feature set. Then, the candidate feature sets are evaluated through 10-fold cross-validations on SVM classifiers that use these candidate feature sets. The best performing candidate set is then transferred to the next iteration. In the experiments, we empirically observed that not more than 10 features were necessary to achieve best accuracy, thus we limited the iteration number to 10. We also eliminated the ones that have no effect in accuracy increase, finalizing the most distinctive features for each trained predictor $Pred$.

C. Action selection with Intrinsic Motivation

Intrinsic Motivation, in its original formulation by Oudeyer et. al.[7], is used to adaptively partition agent's sensorimotor space into regions of exploration, and to guide the agent to select the regions that provide maximal learning progress. In our study, Intrinsic Motivation is used to guide our robot to select actions in order to maximize the learning progress, which is defined as the increase in prediction accuracy of the corresponding action.

The robot keeps learning progress of each action and in each time-step, it selects an action to explore based on the learning progress using ϵ-greedy strategy[14] where ϵ is set to 0.05. If an action (a_i) and a number of objects are selected for exploration at time-step t, the robot first computes the effects predicted to be achieved on these objects using $Pred_{a_i}$. Next, the action is executed on these objects and the generated effects are observed. The success of the robot in predicting the effects, denoted by $\gamma_{a_i}(t)$, is defined as the ratio of the correct predictions on objects explored by a_i, and is used to update the learning progress of the action.

The learning progress (LP) of action a_i is formally defined as the actual increase in the mean prediction accuracy of the predictor ($Pred_{a_i}$) of the corresponding action:

$$LP_{a_i}(t+1) = \overline{\gamma}_{a_i}(t+1) - \overline{\gamma}_{a_i}(t+1-\tau)$$

where $\overline{\gamma}_{a_i}(t+1)$ and $\overline{\gamma}_{a_i}(t+1-\tau)$ are defined as the current and previous mean prediction accuracies of the effect predictor, and τ is a time window, set to 2.

Here we define mean prediction accuracy by setting a smoothing parameter θ to 5:

$$\overline{\gamma}_{a_i}(t+1) = \frac{\sum_{j=0}^{\theta} \gamma_{a_i}(t+1-j)}{\theta+1}$$

This is only a local measure that approximates the real accuracy. We used this local accuracy measure in our online incremental learning setup as the robot cannot access to ground truth, i.e. it cannot know the effect categories of the objects without actually executing its actions on all of them in a real setting.

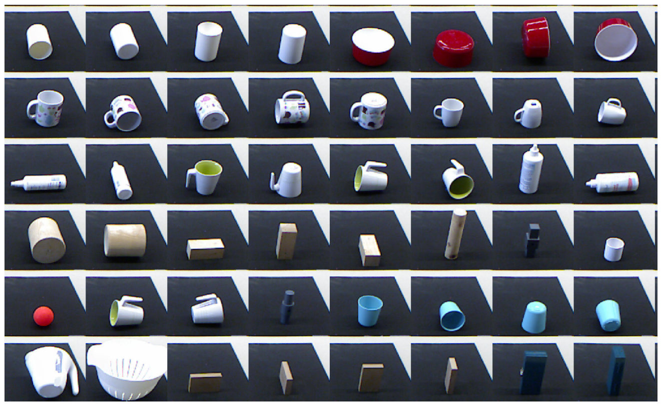


Fig. 3. A subset of the objects used in the experiments.

D. Active object selection

The aim is to select the next object set so that the diversity of the objects in the training set is maximized. For this purpose, the Euclidean distance between objects are computed using one of the three feature types randomly (as computing distance in the joint space of different types would be sensitive to relative weighting of the features). We select the next object from the set of possible objects (PossObjs) by maximizing the total distance from the next object to the already explored objects (ExpObjs) as follows:

$$nextObj = \underset{o_1 \in \text{PossObjs}}{\arg\max} \sum_{o_2 \in \text{ExpObjs}} dist_t(o_1, o_2)$$

where $dist_t(o_1, o_2)$ is the Euclidean distance between two objects in space t, which is sampled uniformly from the set of feature types {size, shape, distance}.

III. Experiment Setup

1) Interaction Dataset: We collected data from 83 objects (Fig. 3) by placing them on the table in front of our robot. Using these objects, we aimed to create an interaction database composed of (object, action, effect) tuples. In order to collect such a dataset, the robot, for example, was required to make $(83 \times 83) = 6889$ interactions for an action that involves two objects, which is not feasible in the real world. Thus, we used a human expert to fill-up the effect field of the complete table[2]

2) Actions: The robot is equipped with a number of manually coded actions that enable single and multi object manipulation. The robot can poke a single object from different sides using *front-poke*, *side-poke*, and *top-poke* actions. It can also stack one object on the other using *stack* action, where it grasps the first object, move it on top of the other one and release it.

[2]Guessing the effects of actions and filling up the table without any reference to robot's real world performance have the risk of creating a human-biased interaction dataset. In order to reduce this risk, we implemented poke and stack in our hand-arm robot system and let the expert observe the robot action executions on a number of different sample objects; and generalize his observations to other objects.

3) Action effects: The effect of stacking objects on top of each other depends on their relative size. For example, while 'inserted-in' effect is generated when a small box is stacked on a hollow cylinder, 'piled-up' effect is observed when the box is larger than the opening on top of the cylinder. Using the objects, we marked the interaction results for each object pair for stack action. Different poke actions also generate different effects even on the same objects. For example, when poked from side, lying cylinders will roll away, boxes will be pushed, objects with holes in poke direction will not be affected as finger would go through the hole without any interaction, and tall objects will topple down. The set of manually encoded actions and their effects are as follows

- Actions: {side-poke, top-poke, front-poke, stack}
- Poke-effects: {pushed, rolled, toppled, resisted, nothing}
- Stack-effects: {piled-up, inserted-in, covered, tumbled-over}

Note that all the effects can be differentiated based on changes in visual features of the objects, except for the effects 'resisted' and 'nothing', which require force readings from the end effector of the robot.

4) Object features: The objects are segmented based on depth information of Kinect sensor that is placed over the torso of the robot. Features are encoded in a continuous vector composed of shape, size and local distance related properties for object o:

$$features^o = (shape^o, dim^o, dist^o)$$

Shape features are encoded as the distribution of local surface normal vectors from object surface. Specifically histograms of normal vectors along each axis, 8 bins each, are computed to form $3 \times 8 = 24$ sized feature vector. dim encodes the object size in different axes. $dist$ features encode the distribution of the local distance of all pixels to the neighboring pixels. For this purpose, for each pixel we computed distances to the neighboring pixels along each 4 direction on Kinect's 2D depth image. For each direction, we created a histogram of 20 bins with bin size of $0.5cm$, obtaining a $4 \times 20 = 100$ sized vector for the $dist$.

IV. Experiment Results

Using the database of 83 objects, 4 actions, and their corresponding effects, we applied active learning of affordances with distinctive features method (Algorithm 1) to discover the structure and development order of the affordance learning system.

A. Discovered development order

This section provides the obtained development order of the affordance predictors. Recall that development order refers to maturation order of the action predictors, and can be analyzed by examining the order and frequency of actions, selected during each iteration of the online learning of the complete system (Algorithm 1, Step 7). The action selected for exploration in each iteration step is shown in Fig. 4. As shown, the less complex poke actions are learned first, and more complex stack action is learned later. As the effect of paired-object actions depend on the relations between properties of two objects, stack is a more complicated action, difficult to learn. Prediction of stack action can also benefit from simple-affordances (as we will show in the next section). Thus, stack action is explored and learned automatically after all other simpler actions are explored. In the figure, the stack action is observed to be explored also in the beginning of the learning in a number of steps either because of momentarily increases in local accuracy or due to the ϵ-greedy strategy.

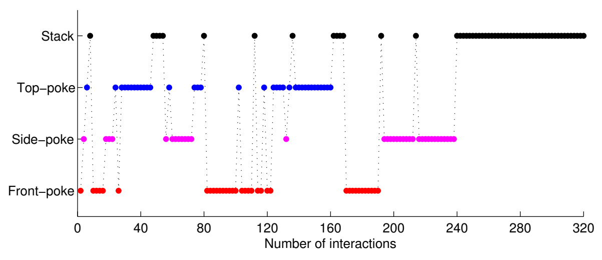


Fig. 4. The action selected for exploration and learning in each iteration of online learning of affordances. As shown, single-object affordances, i.e. prediction of effects of top-poke, side-poke and front-poke. are learned initially. As prediction of the effect of stack action requires learning of features (and probably affordances) of both objects, paired-object affordances are explored later.

We also plotted the local prediction accuracy γ evolution of each action in Fig. 5. The actions that are selected in the corresponding iteration step is illustrated with a mark along with its accuracy plot. As we defined in Section II-A, the accuracy of the predictors are computed using the small number of objects (denoted by objs in Algorithm 1; 4 objects in this experiment) explored in that iteration. This causes a jerky performance evolution as shown in the figure. We run the same algorithm with different initial conditions (initial objects), and observed that the exact shape of each accuracy plot and the exact order among single-object actions change. However, the tendency of learning single-object affordances first, and paired-object affordances later, remained consistent.

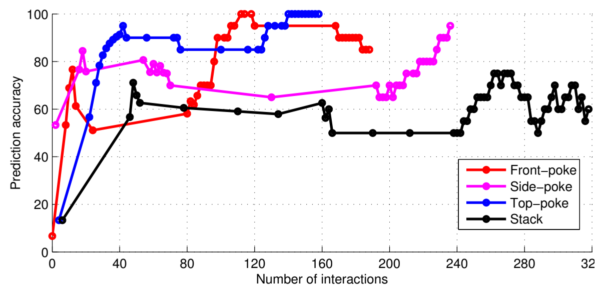


Fig. 5. The evolution of prediction accuracy of each predictor during online learning. In each time-step, one of the four predictors is being updated depending on the selected action, where this selection is illustrated by the marks on the plots.

B. Discovered affordance prediction structure

This section gives the results of the structure evolution of the affordance prediction system. Recall that the prediction

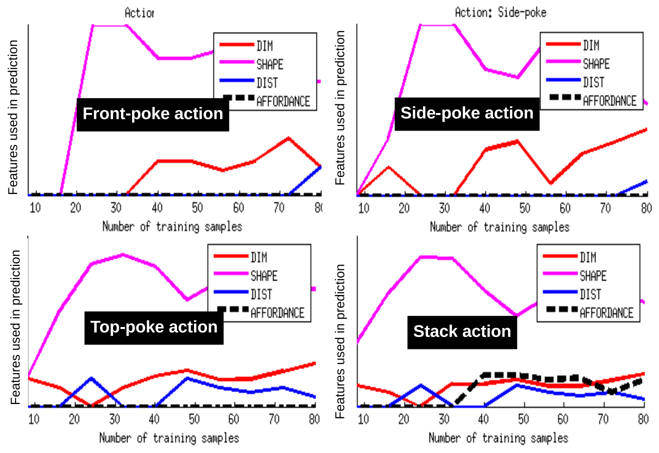


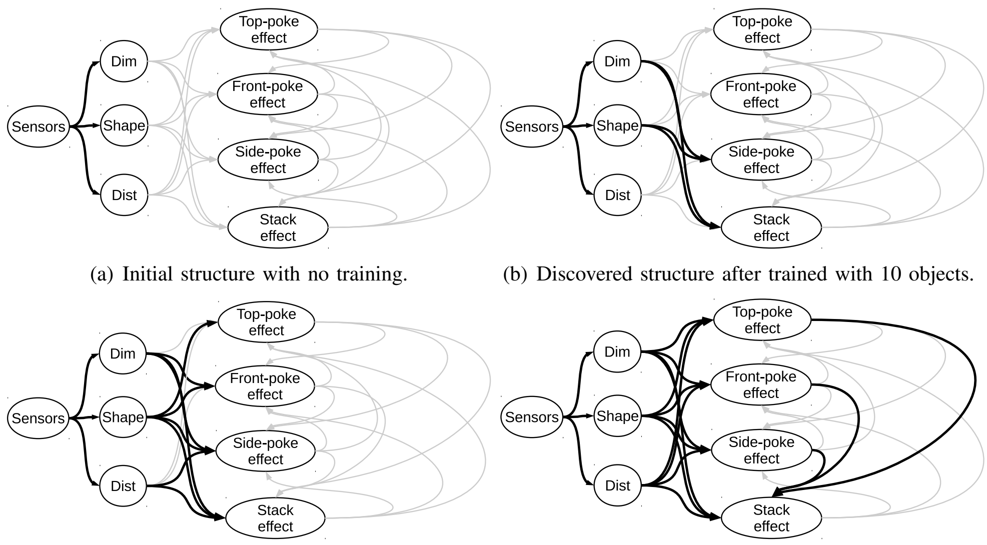
Fig. 6. Evolution of the distinctive features for prediction, where solid lines correspond distance, shape and dimension features, and dashed lines correspond to the predicted affordances. As shown, only effect prediction of the stack action uses affordances as distinctive features. Note that use of affordances for prediction starts after step 30, probably after effect predictors of poke actions (and their affordance corresponding predictions) have developed.

structure is defined over the most distinctive features that are discovered to be most effective in predicting affordances (Section II-B). The robot learns the affordances similar to the previous experiment, but in order to analyze the discovered structure independent of an action selection strategy, the next action in each iteration is selected randomly in this experiment. The ratio of the types of distinctive features used in prediction in different phases of the online learning are shown in Fig.6. Each plot in this figure corresponds to evolution of the used features and affordances for a different action. As shown in the plots, each low-level feature affects affordance predictions in different levels, and shape features are observed to be the first discovered distinctive features especially in the initial phases of development for all actions. However, more important in the context of this paper, affordances are observed not to be used in the initial phases, and are only found to be used in predicting effect of stack action, i.e. predicting stackability affordances. Note that stack predictor starts using single-object affordances after around 30 samples, probably because the single-object affordance prediction was not good enough before that time-point.

We also plotted the exact structure, i.e. features and affordances used by different effect predictors by highlighting the links in the prediction system in Fig.7. Different plots provide the structure in different iterations of learning. As shown, at the end, a hierarchical structure is formed as expected, where learned simple affordances are used in learning and prediction of more complex affordances.

V. Conclusion

In this paper, we studied how interdependent affordance learning tasks can be autonomously structured along with its developmental order. In an online learning framework, we showed that intrinsic motivation mechanism, which select



(a) Initial structure with no training. (b) Discovered structure after trained with 10 objects.

(c) Discovered structure after trained with 20 objects. (d) Discovered structure after trained with 80 objects.

Fig. 7. Evolution of the structure discovery of the affordance prediction system. Black lines correspond to the "distinctive" features that are used by affordance predictors. As shown, a hierarchy gradually emerges where at the end (d), single-object affordances are predicted based on only object features, and further used in predicting paired-object affordances.

the next action to explore, based on learning progress of the model of that action, can discover such a development order where paired-object affordance learning follows maturation of single-object affordances learning. Next, we showed that by using the most discriminative features for affordance prediction, the expected hierarchical structure emerged autonomously where the learning system discovered that predictions of the single-object affordances are connected to the paired-object affordances. We validated our approach in a real dataset composed of 83 objects and pairs of these objects along with the effects of three poke actions and one stack action. The results show that hierarchical structure and development order emerged from the learning dynamics that is guided by Intrinsic Motivation mechanisms and feature selection approach. In order to further verify our approach, we are currently working on realizing the learning cycle in the real robot with the aim of self-discovering the effect categories autonomously and analyzing the results with multiple independent trials.

In this paper, we assumed existence of discrete action primitives and effect categories. We safely made such simplifications and assumptions in the developmental setting of this paper, as we already showed that a set of basic primitive actions can be self-discovered through in interaction based on observed tactile profiles in [15], and effect categories can be autonomously found for different actions, such as rolled-out-of-table, pushed, no-change, grasped in [11].

Acknowledgements

This research was supported by European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience.

References

[1] M. Hayashi and T. Matsuzawa, "Cognitive development in object manipulation by infant chimpanzees," *Animal Cognition*, vol. 6, pp. 225–233, 2003.

[2] M. Ikuzawa, *Development diagnostic tests for children*, 2000.

[3] E. Ugur, S. Szedmak, and J. Piater, "Bootstrapping paired-object affordance learning with learned single-affordance features," in *4th International Conference on Development and Learning and on Epigenetic Robotics*, Genoa, Italy, 2014.

[4] E. J. Gibson, "Perceptual learning in development: Some basic concepts," *Ecological Psychology*, vol. 12, no. 4, pp. 295–302, 2000.

[5] E. Ugur and E. Şahin, "Traversability: A case study for learning and perceiving affordances in robots," *Adaptive Behavior*, vol. 18, no. 3-4, 2010.

[6] M. Lopes, P.-Y. Oudeyer, *et al.*, "Guest editorial active learning and intrinsically motivated exploration in robots: Advances and challenges," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 2, pp. 65–69, 2010.

[7] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp. 265–286, 2007.

[8] S. Ivaldi, S. Nguyen, N. Lyubova, A. Droniou, V. Padois, D. Filliat, P.-Y. Oudeyer, and O. Sigaud, "Object learning through active exploration," *IEEE Transactions on Autonomous Mental Development*, pp. 56–72, 2013.

[9] P.-Y. Oudeyer and F. Kaplan, "What is intrinsic motivation? a typology of computational approaches," *Frontiers in neurorobotics*, vol. 1, pp. 1–6, 2007.

[10] A. Szokolszky, "An interview with Eleanor Gibson," *Ecological Psychology*, vol. 15, no. 4, pp. 271–281, 2003.

[11] E. Ugur, E. Oztop, and E. Sahin, "Goal emulation and planning in perceptual space using learned affordances," *Robotics and Autonomous Systems*, vol. 59, no. 7–8, pp. 580–595, 2011.

[12] C.-C. Chang and C.-J. Lin, "Libsvm : a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 27, pp. 1–27, 2011.

[13] A. W. Moore and M. S. Lee, "Efficient algorithms for minimizing cross validation error," in *Proceedings of the 11th International Conference on Machine Learning*, R. Greiner and D. Schuurmans, Eds. Morgan Kaufmann, 1994, pp. 190–198.

[14] R. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.

[15] E. Ugur, E. Sahin, and E. Oztop, "Self-discovery of motor primitives and learning grasp affordances," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3260–3267.