BIDIRECTIONAL MULTI-STEP PREDICTION WITH AFFORDANCES

by

Utku Bozdoğan

B.S., Computer Engineering, Boğaziçi University, 2019

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University
2022

ACKNOWLEDGEMENTS

This research is supported by the TUBITAK BIDEB 2210-A program.

The numerical calculations reported in this work were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

ABSTRACT

BIDIRECTIONAL MULTI-STEP PREDICTION WITH AFFORDANCES

Affordances are action possibilities of an object, directly perceived by agents based on their capabilities. Affordances are learned from goal-free exploration of the agent's capabilities through observing the effects of their actions on objects in an environment. The agent can then use the learned affordances to make plans to reach a goal since the agent knows which actions on a certain object are possible and which action results in the desired effect. The affordance principle is also followed in robotics to learn to distinguish which actions in the repertoire of a robot are applicable to an object in its environment. This information can then be utilized in goal-directed planning, either directly or with the aim of reducing the search space for possible solutions. In this work, the problem of making multi-step predictions for object manipulation is investigated in the continuous domain. Several types of actions are defined in a robot's repertoire, and the interactions of the robot with a number of objects possessing differing qualities in a tabletop setting are recorded. Relative distance quantities are used for representing actions and effects which allow generalizability, alongside a top-down centered depth image of the object. This data is used to train a model which can be conditioned on actions to predict the effects, conditioned on effects to predict the applied actions, or conditioned on both to predict the actions and effects. By using a planner on top of this model, the capacity to chain together a correct sequence of actions for an object to reach the desired goal position is achieved. The model is verified in experiments, generating and executing reasonable plans efficiently. Setting it apart from previous work, using continuous effect and actions enables the planner to find solutions to configurations that were not observed in training using partial action executions.

ÖZET

SAĞLARLIKLAR İLE ÇİFT YÖNLÜ ÇOK ADIMLI TAHMİN

Sağlarlıklar, bir nesnenin bir aktör tarafından kapasitelerine dayalı olarak doğrudan algılanan eylem olanaklarıdır. Sağlarlıklar, aktörün kapasitelerini hedefsiz keşfetmesi sırasında nesneler üzerindeki aksiyonlarının etkilerini gözlemlemesinden öğrenilir. Aktör daha sonra, belirli bir nesne üzerinde hangi eylemlerin mümkün olduğunu ve hangilerinin istenen etkiyle sonuçlandığını bildiğinden, bir hedefe ulaşmak için planlar yapmak için öğrenilen sağlarlıkları kullanabilir. Bir robotun repertuarındaki hangi eylemlerin çevresindeki bir nesneye uygulanabilir olduğu ve hangi etkiye sahip olduğunu ayırt etmeyi öğrenmek için robotikte de sağlarlık ilkesi izlenir. Bu bilgi daha sonra, doğrudan veya olası çözümler için arama alanını daraltmak amacıyla hedefe yönelik planlamada kullanılabilir. Bu çalışmada, sürekli alanda nesne manipülasyonu için çok adımlı tahminler yapma problemi incelenmiştir. Bir robotun repertuarında çeşitli eylem türleri tanımlanır ve robotun bir masa üstü ortamında farklı niteliklere sahip bir dizi nesneyle etkileşimleri kaydedilir. Nesnenin yukarıdan ortalanmış derinlik görüntüsünün yanı sıra genelleştirilebilirliğe izin veren eylemleri ve etkileri temsil etmek için göreceli mesafe miktarları kullanılır. Bu veriler, etkileri tahmin etmek için eylemler üzerinde, uygulanan eylemleri tahmin etmek için etkiler üzerinde veya eylemleri ve etkileri tahmin etmek için her ikisi üzerinde koşullandırılabilen bir modeli eğitmek için kullanılır. Bu modelin üzerinde bir planlayıcı kullanarak, bir nesnenin istenen hedef konumuna ulaşması için doğru bir eylem dizisini zincirleme kapasitesi elde edilir. Model, deneylerle doğrulanmıştır, makul planları verimli bir şekilde üretir ve yürütür. Daha önceki çalışmalardan farklı olarak, sürekli etki ve eylemlerin kullanılması, planlayıcının, eylemlerin kısmen uygulandığı ve eğitimlerde görülmeyen konfigürasyonlara da çözüm bulmasını sağlar.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\mathbb{N}$	Set of natural numbers
$\mathbb{R}$	Set of real numbers
$SM(t)$	Sensorimotor data value at time t
β	Encoder network parameters
γ	External parameter
δ	Decoder network parameters
θ	Action angle
μ	Mean of output
σ	Standard deviation of output
ϕ	Output

LIST OF ACRONYMS/ABBREVIATIONS

CNP	Conditional Neural Processes
CNMP	Conditional Neural Movement Primitives
CNN	Convolutional Neural Network
DMBN	Deep Modality Blending Networks
GRU	Gated Recurrent Unit
LSTM	Long Short Term Memory
MVAE	Multimodal Variational Autoencoder
RGB	Red Green Blue
RGBD	Red Green Blue Depth
UR10	Universal Robots 10

1. INTRODUCTION

Affordance is a term coined by the ecological psychologist James J. Gibson. Gibson defined affordances as what an object affords to the observer, based on its properties, achieved via direct perception [1]. Both motor and cognitive capabilities of the animal affect its perception capability, which determines the perceived affordances.

Evidence has been reported in the fields of psychology and neurophysiology supporting the principle of affordances. Gibson put forward the groundwork, however since Gibson, a number of explanations have been proposed [2, 3] and there are multiple interpretations of qualities of affordances such as how affordances should be represented, or whether affordances are properties of the environment, the agent, or a relation between the two. A representation is shown in Figure 1.1. Roboticists also have their own interpretations [4], sometimes extending the original definition such that an action is considered to be afforded regardless observing an expected effect, and conflicting with the orthodox definition such that affordances are considered to be encoded with internal representations. There are developmental approaches among roboticists where the robot's possible interactions with the environment are represented with an affordance model, where the robot learns from its interactions. In time, the robot is shown to learn certain affordances, and also there exist works where the learned affordances are used to learn higher-level, more complex affordances, which also agrees with the principle of affordances.

Thinking from a human standpoint; mature humans are well capable of understanding how to act on and manipulate objects according to their goals, even if the objects or goals are novel to them and they are also good at predicting the outcomes of their actions. Humans acquire this capacity through learning over time.

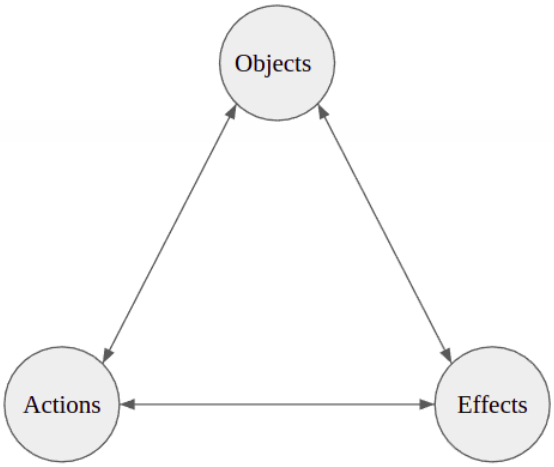


Figure 1.1. Affordance relation between objects, actions, and effects, adapted from [5].

Babies have certain reflexes from birth that they use for gaining experience, but in time the reflexes disappear and intentional actions take their place. These actions are generally more complex, such as manipulating objects after reaching them, and enable the humans to discover more complex skills [6].

From a robotics standpoint, learning to predict the effects of a robot's actions is useful since this can prevent potential failures and dangerous situations and enable planning to achieve certain goals. Planning for multi-step tasks in the real world is difficult, and a generalized approach to solving this problem is even more challenging. Due to its difficulty, previous works compromise on certain aspects, such as predefining the actions, effects, and/or object categories, simplifying the task. These simplifications include discretizing continuous sensorimotor information with the purpose of being able to make high-level planning with symbols. The aim of this work is to learn a deep model in the continuous sensorimotor space of the robot, taking into account the affordances of objects. This model can then be used to:

(i) predict the actions and effects, provided only with minimal observation, and
(ii) generate and execute multi-step object manipulation plans efficiently.

The contributions of this thesis are as follows:

- The robot acquires bidirectional prediction capability between actions and effects. In other words, given object information, it not only predicts effects of actions and but also predicts required actions to generate desired effects.
- The system learns predictors from continuous interactions in order to make multi-step planning.
- The system is able to consider the effects of partial actions executions in making plans. In other words, our system forms plans which may be composed of partial action executions as well.

In this work, affordances are learned through using deep neural networks. No intermediate object recognition step is used, following direct perception approach. Push and grasp actions are used in a continuous object manipulation domain for learning affordances. Depth images of objects are used as inputs to the neural network along with action parameters in order to predict the action outcome in terms of details motion trajectories of the manipulated objects.

Our approach utilizes a conditional model, namely Conditional Neural Processes [7], which is shown to achieve high performance in modeling temporal relationships in long-horizon prediction tasks. This model is paired with bidirectional prediction capability to create multi-step plans that can be executed effectively in order to accurately achieve given goals. Note that, our system not only predicts effects given objects and actions, but it can also predict actions given objects and observed effects.

2. RELATED WORK

In early works such as [8], affordances of specific objects were learned with respect to actions instead of generalized affordances, and objects were recognized before affordances were detected, conflicting with the direct perception principle of affordances. In [9], while the affordance principle was respected there was only a single affordance category, namely traversability, which was predefined and not discovered. In [10] the effects were discovered and categorized discretely via unsupervised clustering, enabling effect category prediction, but such unsupervised discretization step limits generalization in our opinion. In [11], the previous approach was extended by hierarchical clustering on top of various perceptual channels for more effective effect category discovery. Both [10] and [11] enabled forward chaining for multi-step planning, however effects of partial action executions were not considered. In [12], the objects were represented via point clouds, in a non-parametric manner, to provide direct perception. the grasping and pouring actions were shown to generalize well on novel objects but result in single-step plans only. Bayesian Networks were used [5, 13], enabling bidirectional predictive capabilities using the robot's own interaction experience, but clustering was performed on object features and effects in a predefined manner.

In [14], symbolic planning was studied. They applied clustering in order to find effect categories as in the previous works. However, object categories were considered to be a collection of effect categories obtainable by actions available to the robot. This enabled direct planning using learned object and effect categories. However, as only discrete full action effects were considered, the predictions were approximate, resulting in long-horizon planning errors. This work was validated on a real-world setup with simple actions such as poke, grasp, release and stack. The experience of the robot enabled it to gain experience from the single-object actions.

The object categories learned from single-object interactions were learned to generate valid plans that involved stack actions. In [15], probabilistic planning with symbols using parameterized actions was addressed in a real robotic task, showing that continuous tasks can be performed with discrete planners using parameterized behaviours.

Recently, use of RGB/RGBD images for predicting affordance classes or pixel-wise affordance labels for object manipulation has become popular [16–21]. This approach was shown to be effective on learning how to grasp different objects. [22] was able to learn the affordances of objects such that their system could place objects/humans in correct poses in a scene and also choose the correct object type to place in a given scene. [23] used point clouds to learn general geometric features from object interactions, enabling them to place objects in a scene correctly.

Integrating multimodal information was also shown to be a successful approach for learning tasks in general [24, 25] as some modalities might be missing during action execution and those modalities can be predicted by the available ones in these models. The same approach was also shown to be effective for robotics tasks that use sound, language, or tactile information alongside with joint inromation [18,26–30]. [31] studied bidirectional prediction and multimodal processing in order to ground affordances with human-provided language labels.

Affordances can also be used to reduce the search space in order to efficiently generate plans to solve long-horizon tasks. Recent approaches utilizing this idea extend the definition of affordances with the action feasibility [32] or action intent/goal [33]. However, the feasibility concept of [32] accepts a grasp action achieving nothing as afforded, and would also accept a grasp action as afforded regardless of whether it was an appropriate grasp for an object. While [33] overcomes this with intent representation, intents are specified a priori in their work and although a sub-goal discovery direction is proposed for learning, it is not well explored.

In [34], learning to predict the full motion trajectory of an object using the robot's own interaction experience is studied. Their neural-network based approach used the top-down images of the objects as input. In their work, they investigated the importance of different features such as hand-crafted shape features, CNN extracted features, or support point features extracted from a neural network. The authors used these features and the interaction experience to train a recurrent network and were able to accurately predict trajectories resulting from a lever-up action in a real-world setting with multiple objects. As our model also relies on predicting full motion trajectory of objects given action parameters, we compared the performance of our model with their model in the experiments section.

In [35], a generative model, which was trained using interaction images, was used to propose potential affordances. Their aim was to learn a generalizable prior from interaction data and then utilize it to propose reasonable goals for unseen objects. These goals were then attempted to be executed by an offline RL policy, which was also learned from interaction data, and tuned online efficiently to adapt to unseen objects.

A common shortcoming of the aforementioned methods, that could make predictions for action or effects, is related to the use of recurrent methods for long-horizon tasks. The use of recurrent networks such as LSTMs [36] or GRUs [37] is shown to be effective for short-term predictions. However, their recurrent structure causes any error in their prediction to accumulate over time, causing lower success rates in executing long-horizon tasks [38]. We aimed to overcome this problem using Conditional Neural Processes [7] in order to make effective predictions that can be used for planning.

3. METHODS

In this chapter, we present the background methods and our proposed method. Section 3.1 introduces the background methods which lead up to our method in given order, and Section 3.2 introduces our method.

3.1. Background

3.1.1. Conditional Neural Processes

Conditional Neural Processes (CNP) [7] is an approach bringing together the inference potential of Gaussian Processes and the training of neural networks with gradient descent by learning a prior from data. This allows the method to remain scalable and be able to make accurate predictions for desired targets sampled from a distribution conditioned with observations. A neural network h is used to encode varying numbers of sampled observations O into fixed-sized representations r_i. The order of observations can vary, in accordance with stochastic processes, and they are aggregated via a commutative operation into a single vector r which represents the prior knowledge available. This information is used for conditioning, and another neural network g decodes r and for targets $x_j \in T$ to generate predictions which are Gaussian distribution parameters. The formulation is

$$r_i = h_\beta(x_i, y_i), \quad \forall (x_i, y_i) \in O, \tag{3.1}$$

for the encoding of each observation, followed by

$$r = r_1 \oplus r_2 \oplus r_3 \oplus ... \oplus r_i, \tag{3.2}$$

where $\oplus$ is a commutative operation, like summation or dot product between all the encodings resulting in a single encoding.

After concatenation with target variables, the merged representation is decoded into the output by the decoder as

$$\phi_j = g_\delta(x_j, r), \quad \forall x_j \in T. \tag{3.3}$$

The output is expressed as

$$\phi_j = (\mu_j, \sigma_j^2), \tag{3.4}$$

which is a concatenated result of the mean and standard deviation of the output variable.

3.1.2. Conditional Neural Movement Primitives

Conditional Neural Movement Primitives (CNMP) [39] is an extension of CNPs, designed to work with temporal relations (t) and external task parameters collectively named as (γ). An observation is denoted as (t_i, γ) instead of x_i, and y_i are replaced by $SM(t_i)$ that corresponds to sensorimotor data at time t_i, which can be multi-dimensional variable. Averaging operation is chosen for aggregating r, and concatenation is used to incorporate (t_i, γ). A trajectory and a number of observation points are selected randomly from the training dataset, and target data is predicted using the observations during training. After training, the model can be conditioned with any number of observations in order to predict single or multiple targets. On a real-world task, as shown by the authors in their experiments, the network can be conditioned to current observations, making it robust to changes in the environment. The encoder and decoder networks are trained jointly, with the following loss function

$$\mathcal{L}(\beta, \delta) = -logP(y_j|\mu_j, softmax(\sigma_j)), \tag{3.5}$$

where both the mean and the standard deviation from the prediction are utilized.

3.1.3. Deep Modality Blending Networks

Deep Modality Blending Networks [40] is a further augmentation over CNMPs, specifically tailored to incorporate multimodality information. This is achieved by encoding information coming from different modalities into a common latent space, using a weighted average of encoded modality representations, facilitating information sharing. This also provides a regularization effect on the representations learned, similar to dropout [41].

Each modality is encoded separately by its own encoder, and the latent representations are subjected to a weighted averaging operation. This merged representation is then decoded by separate decoders, each corresponding to a different modality. After training, the authors show that the latent representations belonging to different modalities, initially encoded in different parts of the latent space, and started to overlap during training. This enabled the prediction of missing modalities.

3.2. Proposed Method

3.2.1. Our Model

We aim a framework that is able to learn complex temporal multi-modal relations with respect to external parameters. Our system should also be able to make continuous effect predictions given objects and actions. For this, we propose to use DMBNs. An overview of our architecture is provided in Figure 3.1. Our proposed model should be able to predict action related information (such as joint trajectory) and effect on an object (such as movement trajectory of an object) conditioned with object, action and effect related information. The latent space of the architecture is utilized to achieve a shared representation for actions a and effects e for different objects o in order to learn affordances. Learned affordances then can be used to predict the effects of actions or vice versa, enabling chaining of the model's predictions to create plans for solving multi-step tasks.

In our implementation, the object is represented with its top-down depth image that is centered around the object. The action is encoded with the relative distance of the robot's end-effector to the object. The effect is encoded as the displacement of the object from its starting position before the action execution. A crop from a single depth image centered on the object is included as an external parameter γ for the action. Our effect and object representations are similar to [42], but our action representation is relative.

For learning, a dataset of interaction trajectories is used. An interaction trajectory can be expressed as

$$D_d = (\{a_t, e_t, o, t\}_{t=0}^{t=1})_d, \tag{3.6}$$

where d corresponds to the interaction trajectory, m is the number of trajectories in the data set, and $0 \leq t \leq 1$ where $t \in \mathbb{R}$ is the variable that encodes passage of time.

At each training iteration, one of the action or effect channels or both action and effect channels are chosen randomly. The action channel predicts the action only, the effect channel predicts the effect only and both channels predict both action and effect. The input observations are provided according to the chosen channel. k observations are sampled uniformly at random from a randomly selected interaction trajectory D_d where $1 \leq k \leq obs_{max}$. $k \in \mathbb{N}$ is also sampled from a uniform distribution, and obs_{max} is a hyper-parameter denoting the maximum number of observations that the model is allowed to use during one iteration. These observations are then encoded and aggregated. A crop from a single depth image centered on the object is encoded separately using a CNN encoder network, and the resulting vector is concatenated at the end of this aggregated representation. Before a prediction is made, a target time step is also concatenated after the image features.

Next, representations of different observations are combined as follows:

$$r = \frac{\sum^{k} h_a(\{a_{t_{obs_k}}, o, t_{target}\})}{k} * w_1 + \frac{\sum^{k} h_e(\{e_{t_{obs_k}}, o, t_{target}\})}{k} * w_2, \tag{3.7}$$

where $w_1 = w_2 = 0.5$ in the current implementation. There is no averaging when a single channel (action or effect, not both) is chosen in the corresponding iteration.

Finally, this merged representation r is decoded at the action decoder by

$$g_a(r_{mrg}) = (\mu_{a_{t_{target}}}, \sigma_{a_{t_{target}}}). \tag{3.8}$$

Similarly, effect is decoded using the corresponding effect decoder

$$g_e(r_{mrg}) = (\mu_{e_{t_{target}}}, \sigma_{e_{t_{target}}}), \tag{3.9}$$

to yield predictions for action and/or effect for the target time-step shown in Fig. 3.1.

Gradient descent is used with the loss function provided Equation (3.5) with Adam optimizer [43]. After training, the network is able to predict the entire interaction trajectory given a single observation at $t = 0$.

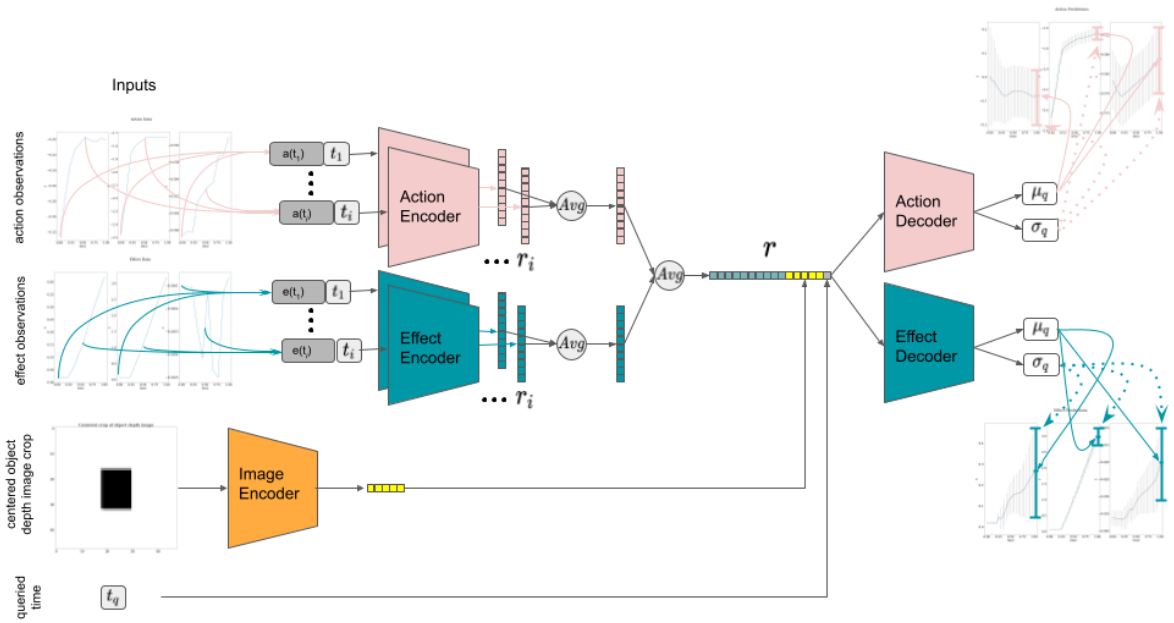


Figure 3.1. An overview of the proposed model.

3.2.2. Planner

An A* planner [44] is used to solve multi-step tasks where the prediction in each step is performed by the neural network structured explained in the previous subsection. The goal of the planner is to generate the sequence of actions that move an object from an arbitrary position to a goal position.

Two actions are available to the planner: push and grasp. The push action is specified by an angle $\theta \in [0, 2\pi]$, a push distance $l = 0.05$ meters and a $radius = 0.2$ meters. The parameterization is shown in Figure 3.2. The gripper starts on the red circumference of a circle of radius $radius$ centered on the object, at angle θ and pushes the object l meters.

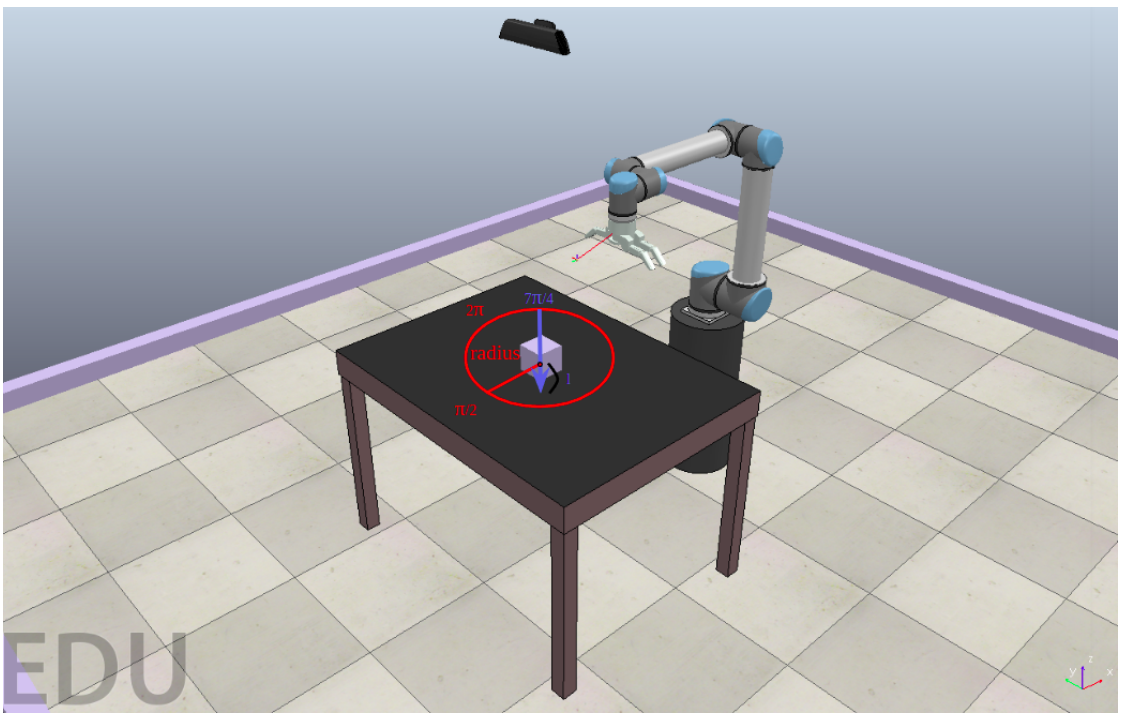


Figure 3.2. Scene showing the parameters of a push action around an object.

The push action can be applied in full, or only partially meaning that after a portion of the action is executed, the action can end without completing the full push distance. Thanks to the continuous predictive capabilities of our model, which is conditioned on time, the results of these partial actions can be predicted. These predictions, on the other hand, can be used to make plans that can move an object closer to the goal position with the corresponding partial actions.

The grasp action, on the other hand, moves the gripper on top of an object, then attempts to grasp and lift, resulting in either a lifted object grasped by the robot or an object that could not be grasped by the robot. The grasp attempt may move an object that could not be grasped, leading to uncertain outcomes.

While the push action can be applied in infinitely many ways, grasp action has no need for such versatility and therefore can only be applied in one way. The grasp action is also applied without partial execution, since half the grasp action is moving on top of an object, and the other half is lifting it.

The planner proposes a sequence of actions to move the object from its initial position to the goal position. Each branch in the search tree corresponds to either a push action or a grasp action. As grasp action is not parameterized, there is a single branch for grasp action. On other hand, as push action might be applied from different approach angles and for different push distances, the range of possible approach angles and push distances are discretized and used to create multiple branches from the same node in the search tree.

The planner uses predictions of actions with different θ parameters to update the predicted location of the object. The search is completed if the difference between the predicted and goal object position is within 2 cm. Our model is able to work with continuous inputs, however, the planner can only propose a finite amount of actions due to the mentioned discretization design choice.

The capacity of the model is thereby affected by the branching factor of the actions proposed by the planner. A depth limit is used during search.

The push action can be applied in many ways, so in order for the planner to choose the best candidate, a heuristic function is used. First, the distance between the target position and the current position is found by

$$\overrightarrow{p_d} = \overrightarrow{p_t} - \overrightarrow{p_c}. \tag{3.10}$$

This is followed by finding the angle of this position vector by applying the inverse trigonometric function arctan2. Here, arctan2 is preferred since the difference between z-axis coordinates is inconsequential as long as the action achieves contact with the object, which it does since the push action is applied close to the table. The result is added to π, expressed as

$$\theta_h = \text{arctan2}(\overrightarrow{p_d}) + \pi, \tag{3.11}$$

since the target of the function is $[-\pi, \pi]$ and being in $[0, 2\pi]$ is desired since this is also the domain of the action propositions by the planner. The angular difference between this angle and the proposed θ_p by the planner is then calculated and constrained within the interval $[0, 2\pi]$ by applying

$$\theta_d = \min\left((\theta_h - \theta_p) \pmod{2\pi}, 2\pi - ((\theta_h - \theta_p) \pmod{2\pi})\right), \tag{3.12}$$

yielding the angular difference θ_d.

Next, the distance to the target and the resulting displacement from a push action are used as sides of a triangle; and the angular difference is used as the angle between the sides to calculate where the object will end up after the push action as

$$\begin{aligned} a &= d(p_t, p_c), \\ c &= 0.1 \times t, \\ b &= \sqrt{a^2 + c^2 - 2ac\cos\theta_d}, \end{aligned} \tag{3.13}$$

where the quantity 0.1 is the minimum expected displacement of an object for a push action, multiplied with the duration variable $t \in [0, 1]$. This variable is related to the partial action execution where any action with $t < 1$ is a partial action. This way partial actions can have a lower costs of execution. The side length b of the triangle is used as the heuristic value. The true cost for the planner is the sum of the distances the object has moved since the start.

In a best-case scenario, the object will be pushed in a straight line towards the target, and our distance heuristic will not overestimate. In any other scenario, since the true cost increase is tied to the displacement of the object, due to triangle inequality our heuristic will always underestimate; making it both admissible and consistent.

4. EXPERIMENTS AND RESULTS

4.1. Planning with Push and Grasp Affordances

4.1.1. Experiment Setup

A simulated scene was constructed in CoppeliaSim [45] where a UR10 robot interacts with objects of 4 different shapes: cube, sphere, sideways cylinder, and upright cylinder. The robot interacts with one of these objects of different sizes using push or grasp actions in this tabletop setting. A Kinect sensor is placed above the table vertically such that the entire table is visible, and the table covers the sensor's field of vision. The parts of the interaction where the object is potentially going to be displaced are recorded. The recorded information consists of action and effect data from each 3 simulation step (a single step is 50ms), and a single depth image of the table with the object on top taken at the beginning of each interaction.

4.1.1.1. Actions. Push actions consist of the gripper following a trajectory starting on a circle with a radius of 20 cm. and at an angle θ parallel to the table, with the goal position 5 centimeters behind the object's center of mass. Larger-sized objects may be displaced more as a result of this setup. The setup can be seen in Figure 3.2. The model is expected to learn the pushability/rollability affordance from these interactions. It is also expected to predict the trajectories of both rollable objects (spheres or lying cylinders) and non-rollable objects (cuboids or upright cylinders) from their image.

There is an additional difficulty posed by the simulator for the push task. In our investigations, we have noticed that actions where the robot pushes objects towards itself (effectively pull actions) are performed slower, thus they taking longer time.

This results in rollable objects such as spheres not rolling away but instead getting dragged by the gripper. Since we are using the relative distance between the gripper and the object as our action input, a sphere behaving like a non-rollable object reduces the quality of the data collected. Due to certain push actions being executed slower than others, those actions create a different profile temporally, taking as much as double the amount of time to be executed for similar amounts of displacement.

Grasp actions consist of lowering the open gripper to grasp position, attempting to grasp an object by closing the gripper, and lifting the gripper up. The model is expected to learn the graspability affordance from these interactions based on size and shape of the objects. The motion trajectories of the objects are expected to be predicted for successful and failed grasp actions.

An important point is that since the model does not use any direct object position information, the predictions are made regardless of the location of an object. This is both desirable and undesirable because an object placed on certain parts of the table affords neither the push action nor the grasp action. This leads to the reachability affordance to be investigated, where an action can be applied only if the object is reachable by the robot with that action. Therefore actions are only applied if the object is reachable, otherwise, no action is applied and there is no effect to be observed, which is also recorded. For this purpose, a classifier is trained from this data to learn the reachability affordance. During planning, if the object is unreachable, the corresponding node in the search tree is not expanded.

4.1.2. Results

4.1.2.1. Prediction performance for single actions. We first evaluate the prediction accuracy of the model on single action tasks. Different models were trained for push and grasp actions separately. The simulation data sets were always split between 80% training, 10% validation, and 10% test data. For all the results reported in this work 10-fold cross-validation was applied unless otherwise specified.

The models were trained for one million iterations, without batches due to variable length of inputs and early stopping was employed. The learning rate was set to 1e−4. All errors reported in meters denote the distances to a specified goal position. The predictions for a single action is fixed to take 25 time steps.

For the push action, a data set made up of 500 trajectories was used. For each interaction, a fixed size objects were chosen randomly and placed at the center of the table. An angle for the push $\theta \in [0, 2\pi]$ was chosen randomly. The robot performed a complete push actions and the resulting interaction data was recorded.

In models where push action is the single action, the object has a fixed starting location at the center, and the z-axis coordinates are redundant since no significant changes occur in z-axis coordinates for the objects during an interaction. The recorded part of the interaction has a short duration relatively, so the interaction ends before a rollable object falls off the table in this case. In models where only grasp action is learned, z-axis information is used as well. The reason for this exception is that the changes in z-axis coordinates are essentially noise since no change is expected to occur during a push. However, the neural network still treats it as an equally important input to the x and y axes and attempts to fit z by sacrificing important information in x and y axes, leading to suboptimal results. A fixed location grasp action is secure from this effect due to the gripper always starting the action from the same relative location.

For the grasp action, a data set made up of 150 trajectories was used. Objects of sizes varying between 0.1 and 0.2 meters were randomly chosen to be placed at the center of the table. The robot then performed the grasp actions and the resulting interaction data was recorded. Larger objects were also attempted but there often were collisions with the robot due to limited space and the depth image crop became completely covered by objects of larger sizes, invalidating its use. Therefore larger objects were omitted. Objects used in the experiments were mostly graspable by the robot, leading to an imbalanced dataset.

Even so, the results for the grasp action in Table 4.1 show that the robot was successful in distinguishing between objects that it could or could not grasp. The evaluation was performed by checking the difference between the z-axis coordinates of the first and last time steps of the test trajectory and the prediction outputs.

In interpreting the performance and success of graspability prediction, if the change in height is larger than 0.1 meters, the grasp was assumed to be a success. If the change in height is less than 0.1 meters for both, it is a failed grasp that is predicted successfully. If the test data does not have a significant change in its z-axis coordinates but the predictions do, then this is a false positive and finally, if the test data has a significant change in z-axis coordinates but the predictions do not, then it is a false negative. By visual inspection of the data, we have found that the robot has more difficulty grasping non-rollable objects of equal size, most likely due to the fact that a cube and an upright cylinder are both grasped by straight surfaces whereas large spheres and sideways cylinders are grasped by curved surfaces that are above their center of mass, causing them to slip easier.

It is important to note that the grasp action error is significantly larger than the average push action error. This has two potential reasons. The first one is there might be many reasons for unsuccessful grasps. The object may slowly slip from the robot's hands and land on the table, close to its initial position. The object may topple or roll, leading to position changes that are uncertain beforehand and therefore cannot be accurately predicted.

Sometimes these position changes are large enough before the interaction finishes that objects roll off the table, leading to even greater displacements. Second, the simulator's physics engine is not a realistic representation of reality. Unrealistic interactions have been observed after failed grasps where the object flies off at high speed. However, even though the predictions are uncertain, the model is highly successful with 94% in distinguishing between graspable and not graspable objects, while also predicting a semi-accurate model of the continuous interaction.

It should also be kept in mind that the model is not measured on regular train/test metrics. Input to the model does not come from a test set that is made up of simulated data. The planner calculates push action propositions by choosing *theta* values by dividing 2π into equal intervals and then calculating a starting position for the gripper. The only input the model receives is the action input and the effect input at time $t = 0$. The effect input is always zero at time $t = 0$ since there is no movement at the start of interaction and therefore does not provide additional information to the model.

Table 4.1. Single grasp action prediction results on variable sized objects placed on a fixed location.

Error (m)	**True Positive (%)**	**True Negative (%)**	**False Positive (%)**	**False Negative (%)**
0.051 ± 0.032	80.00	14.00	4.00	2.00
	94.00		6.00	

In the previous analysis, the object types were chosen randomly. Next, we investigated how the type of the object affects the prediction errors. The results are presented in Figure 4.1. Sphere and sideways cylinder type objects have both higher errors and higher variance in their prediction error. This is expected since they are rollable objects and have higher displacement from an applied push action. Due to the planner proposing only a limited amount of action angles, the applied action angle can be different from the correct action angle. This angular difference affects the final distance of the object from the target, since rolling objects have higher displacement the distance can be higher than a non-rolling object with the same angular difference.

4.1.2.2. Incorporating Reachability Affordance. Up to now, only single actions with objects placed on a fixed location were investigated. This is because when the location of the object is not fixed, there is no guarantee whether an action can be performed on it. The robot may not be able to reach the object at all, or the robot may start applying the action but stop before completion when it can't reach the target anymore.

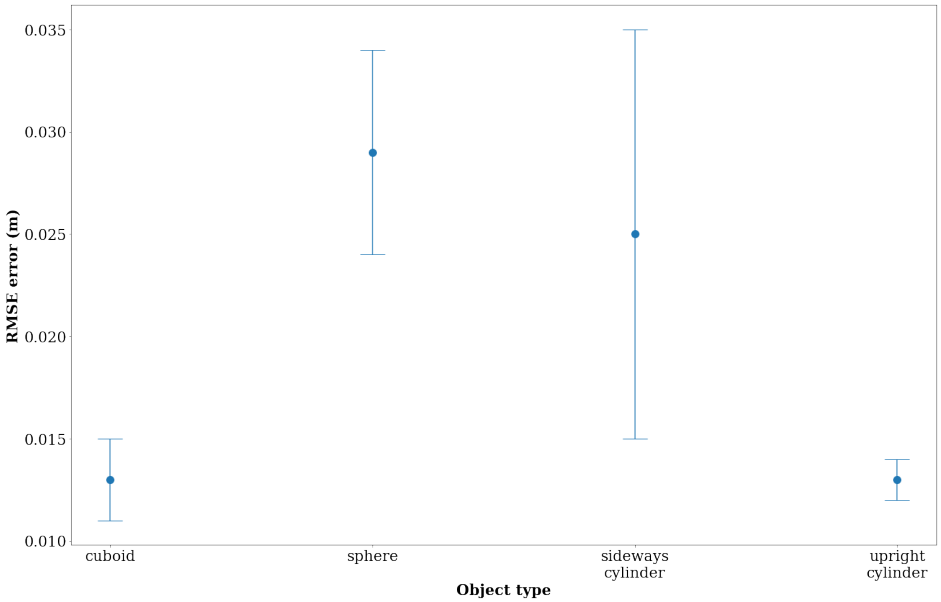


Figure 4.1. Errors based on object type for a single push action on fixed sized objects placed on a fixed location.

Our model does not take into account the position of the objects. Therefore, another component is required which takes into account the positions of the objects and the actions to be applied; which can decide whether an action is feasible or not.

A data set of 1000 interactions was collected where object size and location on the table could vary. The reachability classifier was trained on this data using binary cross-entropy loss with a simple feed-forward neural network.

The inputs to the network are chosen to be the starting object position, the starting gripper position and the partialness of the action. The output is binary, denoting whether the action is reachable or not. The results obtained for this classifier are presented in Table 4.2. When tested on the simulation environment, the classifier correctly identified and eliminated most of the unreachable configurations, enabling finding solutions faster.

In a separately gathered test data set, it was observed that for 34 out of 35 (97.14%) instances of actions being attempted towards objects and failing due to the object being unreachable, the reachability classifier was correct in its prediction that the specific action was indeed unreachable. Figure 4.2 shows one such example where the reachability classifier was tested with all proposals for a configuration. The efficiency gained from discarding unreachable proposals vary depending on the configuration. It was observed that it could be as high as predicting all the actions to be unreachable correctly, or it could be a smaller region as can be seen in the example provided.

Table 4.2. Results from the reachability classifier, outputting whether an action of certain partialness towards an object at a position is reachable.

Classification Accuracy (%)
97.80 ± 0.87

4.1.2.3. Prediction performance for multiple (full and partial) actions. Following the analysis over fully applied single actions, and by enabling reachability, interactions with single and multiple actions are generated, also including partial actions. Partial action corresponds to an action that is started to be executed, but its execution is ended before completion, i.e. before 5 cm motion after object contact. For example in a push action, the push may be cut short before the gripper even contacts the object, or the gripper may reach the coordinates of the center of mass of the object, meaning that the object has already started being pushed, but the push is not completed yet.

Our model was trained only on fully applied single-action interactions. No partial actions were specifically included in the training, but since the model is conditioned on time it can be requested to predict up to a certain time step, effectively providing a prediction for a partial action execution. We call this the temporal branching factor, and use 5 or 10 depending on the task.

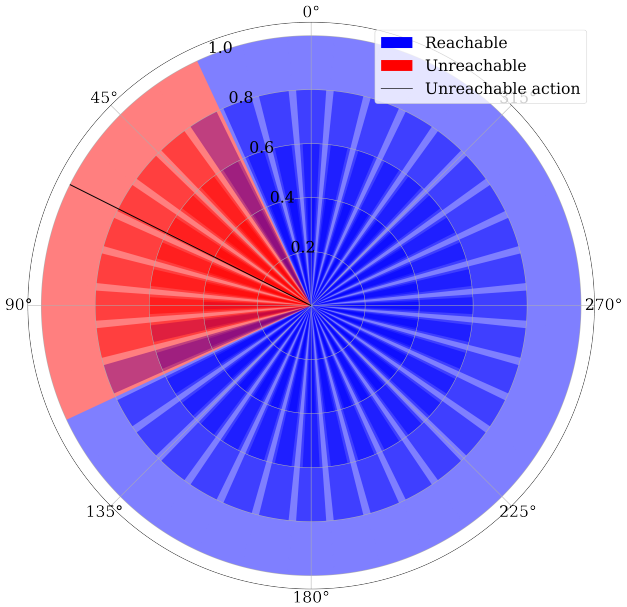


Figure 4.2. Example reachability analysis of a configuration. The same action is tested for reachability at different partialness levels which can be observed at the inner and outer regions of the circle. Actions with angles in the blue regions are predicted to be reachable by the classifier, red regions are predicted to be unreachable. The action with angle at the black line is known to be unreachable.

Our model is requested to generate plans for both single and multiple action interactions, including cases with partial actions. Results from a model trained with fully applied single push actions tested on different push settings are presented in Figure 4.3. Partial actions themselves are parameterized in the interval $[0, 1]$ randomly.

While the model has an advantage due to not having error accumulation within a single step prediction, the model itself cannot correct the accumulation of error between multiple steps. However, the partial actions can have an error reducing effect to a higher degree than that of a system with the capability of executing full actions only.

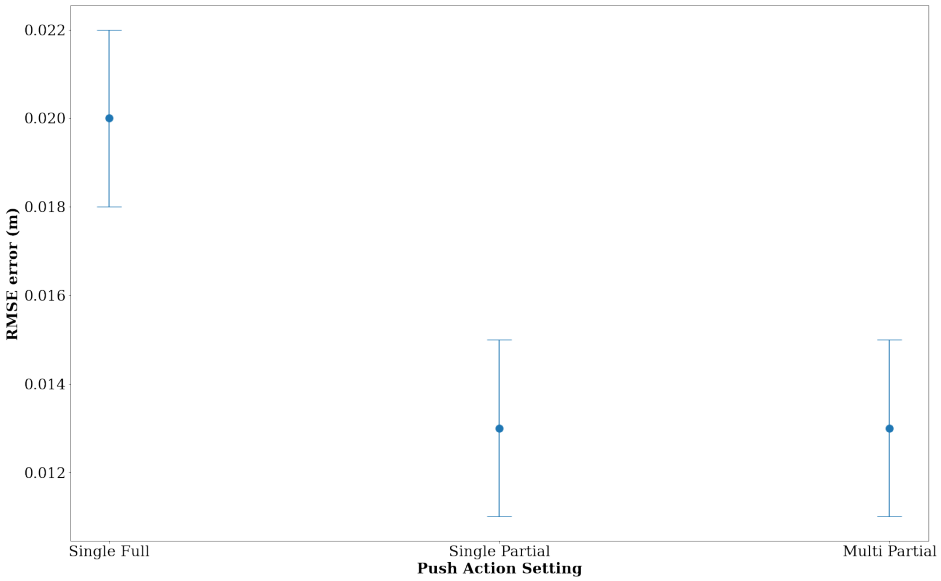


Figure 4.3. Results of applying push actions on objects on a fixed location with fixed size with different settings.

In the case that the prediction of the model is slightly off the goal, given another action our model can make a corrective adjustment using a partial action. Comparing this to a model capable of only applying full actions in the same situation, given more actions the model will require more actions and more effort by the robot to make a corrective adjustment. This phenomenon can be observed in Figure 4.3 where average single full action error is higher than multiple partial action error.

In the case that there are no extra actions available to the model, an example being pushing an object to the right 5 times to get close to the goal while only having 5 actions available. Since the action to push right is learned imperfectly, when it is applied in full, if it is predicted to cause less displacement than it is supposed to, this difference in displacement will be amplified 5 times due to taking 5 actions. Therefore, our model is not immune to compounding errors in multi-step actions.

The multi-step setting is 3 random partial push actions applied one after the other. But the task can be achieved in fewer steps since the robot will aim towards the target using its heuristic. We observe this in our results. We also note that since our model has no via-points or sub-goals, a goal that was generated by applying the actions $< left, left, up >$ can be achieved by our model in the same manner, or by applying any permutation of these actions, or for example with two diagonal push actions.

The planner is utilized by proposing actions and predicting the full action and effect trajectories for proposed actions in a classic forward-chaining manner. However, as explained, the model is not unidirectional and although the purpose of implementing the model in a bidirectional manner is achieving better latent representations, it can also be used to propose effects and calculate the full effect and action trajectories in the same way.

The predictions made by the model appear to have low errors, but there is no guarantee that the predictions will hold in the environment where the data was gathered from. For this reason, the same simulation scene was used to verify that the model's action and effect predictions are in fact realistic and hold out in the simulation world.

One such example is shown in Figure 4.4. The model did not predict orientations, therefore the orientation between the predictions and the ground truth have no reason to match.

4.1.3. Ablation Studies

4.1.3.1. Effect of Training Data Set Size. Reducing the size of the data set used is expected to have an adverse effect on the results. However, the model's generalization ability can help counter this effect up to a certain level. Our results are presented in Figure 4.5.

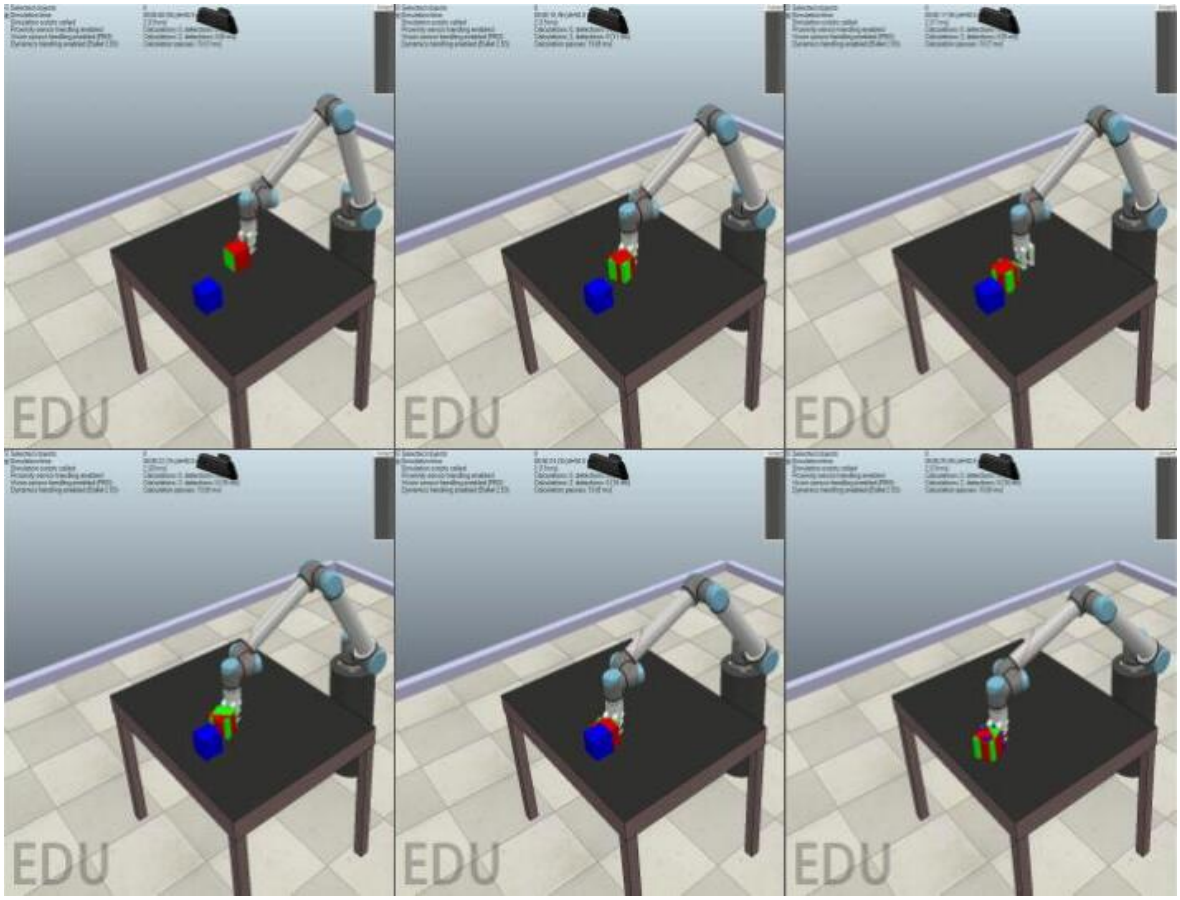


Figure 4.4. Results of applying the model's predicted actions in a scene. Images are ordered from left to right. The top row is from the first action execution, the bottom row is from the second action execution. The blue object denotes the target position, the red object is generated from the robot's effect predictions. The blue and red objects are not interactable by the robot. The green object is interactable and is acted upon by the robot, to show the ground truth results of the robot's predicted actions.

4.1.3.2. Effects of Different Training Regimes. An experiment was performed where the training regime of the network was altered. The proposed training regime is where between action, effect, or both channels were randomly selected at each iteration. To compare, training was done on the same data used to report our results, but only using both channels at each iteration. The results presented in Table 4.3 show that the difference is not significant at $p < .05$.

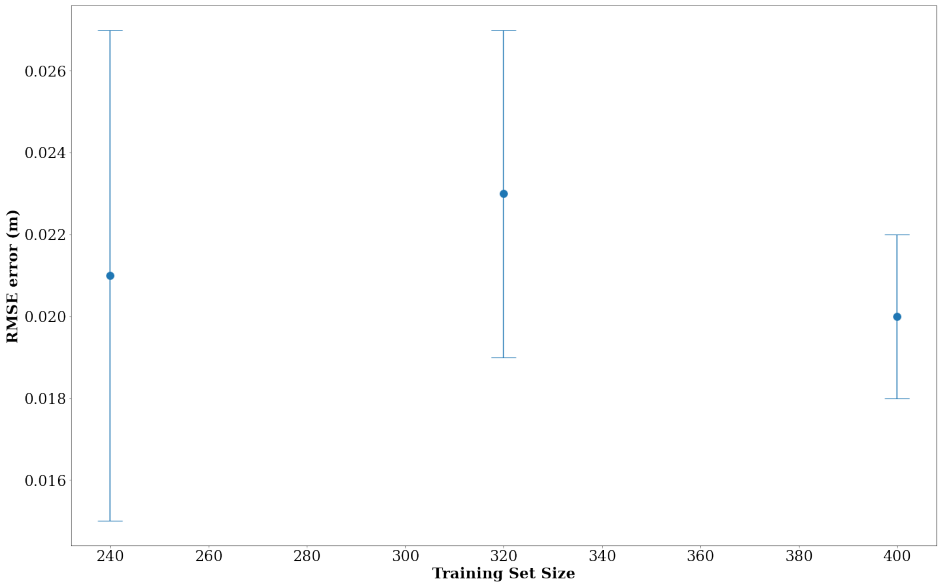


Figure 4.5. Results of using different amounts of training data for a single push action task on objects on a fixed location with fixed size.

Table 4.3. Training regime results comparison for randomly choosing between channels and using both channels all the time for a single push action task on objects on a fixed location with fixed size.

	Error (m)
Random channel	0.020 ± 0.002
Double channel	0.019 ± 0.006

4.1.3.3. Effects of Using Different Mixing Methods. Different mixing methods were applied to the encoded input channels to investigate whether there is a significant performance difference. A simpler dataset consisting of fixed-size objects, always placed at the center of the table and 16 specific available actions were used. Figure 4.6 shows that the model with the random weights has a slightly lower error than the other two models.

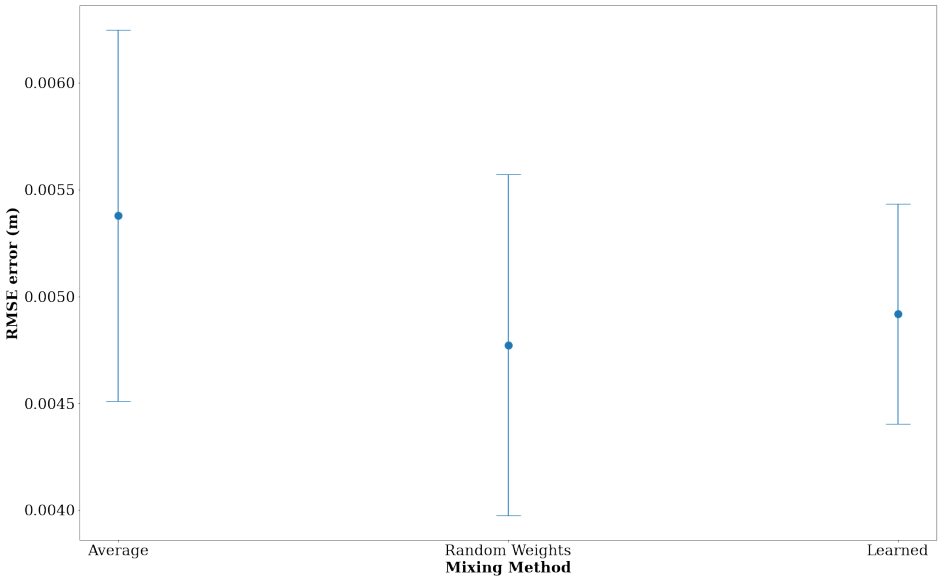


Figure 4.6. Results of applying different mixing methods to actions and effects for a single push action task on objects on a fixed location with a fixed size.

4.1.3.4. Effects of Changing Channel Structure. Several other architectures were also tried to solve this problem. A subset of the current model, where only action information as input and a crop from a single depth image centered on the object as an external parameter was provided was proposed. The model was expected to predict both action and effect information but failed to converge. Another model which is an extension of the current model, where along with action and effect information object images over time were provided as inputs; instead of a crop from a single depth image centered on the object as an external parameter was proposed. Training with RGB images, RGBD images, and depth images all failed to converge.

4.2. Comparison with State-of-the-Art CNN+LSTM Method in Effect Prediction

In this section, we compare the effect prediction performance of our model with [34]. In [34], the motion trajectories of various objects in response to lift action were predicted using a CNN+LSTM based neural network architecture.

An example lever-up action from [39] is shown in Figure 4.7. In their experiments, several geometric shapes (triangle, square, pentagon, hexagon, heptagon) of several sizes were used in lift interactions, along with their center of mass trajectories when subjected to a single lever-up action from different application points along the edges. We used the data set provided by the authors to train our model and compare the performance in predictions.

Two experiments are performed with the simulation data from [34], containing top-down images, support points which are alternative features extracted from top-down images, and center of mass trajectory data from a lever-up action applied from different points for objects of different geometric shapes. The data set is separated randomly into 80% training, 10% validation, and 10% testing, which is similar to the numbers reported in [34]. The network is trained with either single top-down images or support points belonging to an object, along with random numbers of observations from the trajectory, and the center of mass pose of a target time is predicted.

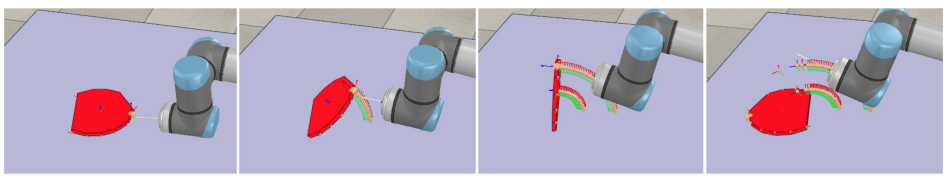


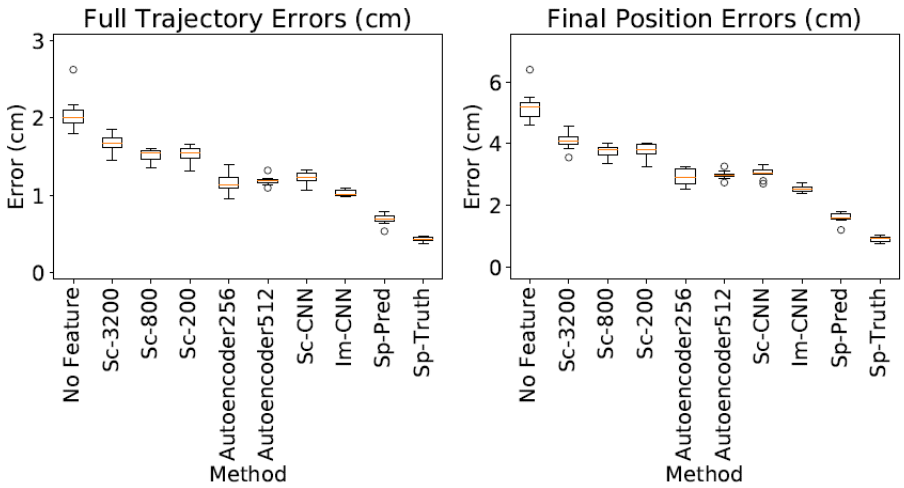
Figure 4.7. Example lever-up action in the simulator, reused with permission from [34].

Our model is tested with either a top-down image and an initial pose of the object, or with support points and an initial pose of the object, and is able to predict full effect trajectories, giving results agreeing with the properties of the object. In the first experiment, support points are used, and added as external parameters to our network; and in the second experiment, the top-down images are used in the same manner.

A CNN encoder is used to extract features from the images, which is trained together with our network. The CNN encoder for the image data is replicated from [34]. The input images are 128×128 grayscale, and during testing, the full center of mass trajectories for the objects are predicted. Figure 4.8 shows the results from [34], using an LSTM network, and Table 4.4 shows the results from our experiments. Both experiments are performed with 5-fold cross-validation and early stopping with one million iterations. Here Sp-Truth corresponds to use of ground truth support point features as inputs and Im-CNN corresponds to use of images, processed by CNNs, as inputs.

Table 4.4. Trajectory errors, results from our experiments.

	Full Trajectory Errors (cm)	**Final Position Errors (cm)**
Sp-Truth	1.146 ± 0.043	3.033 ± 0.090
Im-CNN	0.584 ± 0.044	1.372 ± 0.081



(a) Prediction error for full trajectory (b) Prediction error on final position

Figure 4.8. Trajectory error plots for the LSTM approach, reused with permission from [34].

Our results show higher errors than the compared model on support point data, and lower errors on image data for full trajectory and final position errors. It must be mentioned that the results reported in Figure 4.8 were achieved by one step ahead ($n = 1$) predictions whereas the results of our model are reported by using only a single observation at $t = 0$. A direct comparison cannot be made from these results, however considering that the image encoder is replicated, the difference should come from utilizing the information from the images better in our model.

In order to make a direct comparison, we gathered n-step predictions, taking 15 previous steps as observations, and compared the results with the ones reported in the same manner in [34]. A comparison between Figure 4.9 and Table 4.5 clearly shows that our model outperforms the compared model.

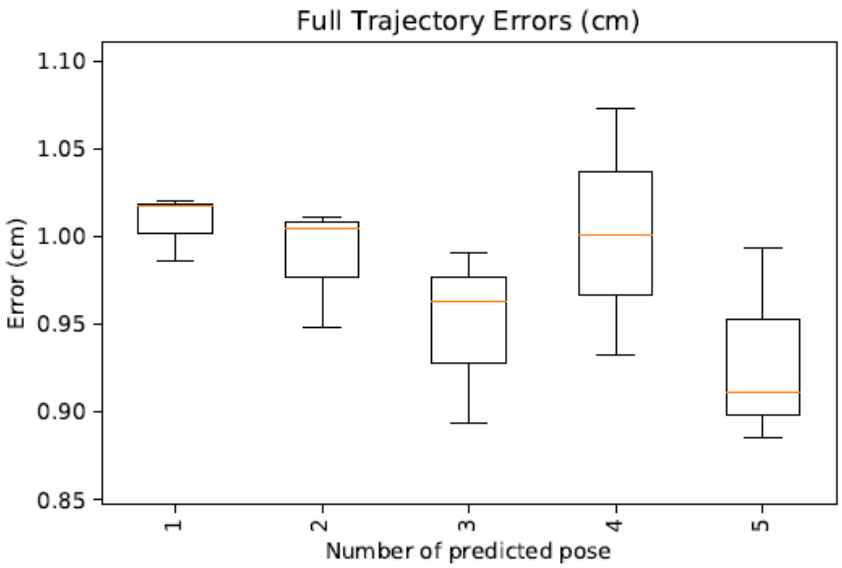


Figure 4.9. n-step error plots for the LSTM approach, reused with permission [34].

Our model is thus shown to perform well on both short and long-term predictions yielding lower error rates compared to a recurrent method. This is probably because the error does not accumulate in our model like in recurrent models.

Table 4.5. n-step prediction errors, results from our experiments.

	Sp-Truth	**Im-CNN**
1-step (cm)	0.354 ± 0.072	0.340 ± 0.044
2-step (cm)	0.390 ± 0.061	0.352 ± 0.046
3-step (cm)	0.417 ± 0.053	0.364 ± 0.049
4-step (cm)	0.445 ± 0.046	0.375 ± 0.051
5-step (cm)	0.474 ± 0.039	0.387 ± 0.054

5. CONCLUSION

5.1. Conclusion

We propose a model for multi-step action and effect prediction. While previous work's utilization of bidirectional learning is limited, our model specifically creates its latent representations using this concept and is able to make multi-step predictions that are in accordance with ground truth manipulations. We emphasize using object-centric inputs to achieve generalizability and investigate simple affordances of several classes of objects of different sizes. By using a network for single interaction predictions which can be interpreted similar to a state transition function and pairing it with a planner with heuristics to propose goal-directed actions the model is shown to have low error in reaching target positions. Our key observations from this work are as follows:

- Using object-centric action and effect input representations enables a level of generalization capability that could not be obtained without a large number of distinct data covering the object positions. This observation is obtained from earlier experiments where object and gripper position was used as inputs to the model and gave poor results.
- Using bidirectional learning in the model enables higher predictive capabilities than using a single channel to predict both action and effect information. This observation is obtained from our ablation studies where a single action channel network predicting both action and effect information learned from the same datasets used to report our results failed to converge.
- Using the proposed training regime does not appear to give a significant decrease in average error, however it may be utilized to reduce the variance of error. This observation is obtained from our ablation studies where effects of different training regimes is investigated.

- Using partial actions are shown to be effective in reducing compounding multi-step prediction errors. This observation is obtained as a result of our simulation experiments with single full, single partial and multiple partial push actions.

5.2. Future Work

While the results of our experiments are promising, the model still requires verification in the real world. Our next step is gathering data and testing our implementation with a real robot.

The reachability classifier is shown to successfully learn the reachability affordance. We are currently investigating a way to incorporate this capability into the network itself, so that when an action is proposed that is unreachable, the network's prediction will be to stay stationary and not act. Such a capability will show that the network can reason about multiple affordances, and more affordances can then be attempted, such as interactions with other objects. Such spatial affordances however require the network to be provided with certain position information. If this succeeds , improvements to the planner may need to be considered.

Our work uses a conditional architecture to avoid the compounding error problems of recurrent architectures and models that are used in a similar way by feeding their current step output as next step input. This advantage of using conditional models is shown in this work against using an LSTM network, and against a Multimodal Variational Autoencoder (MVAE) in [40]. However, transformer models recently gained popularity as being a good alternative to recurrent models. By using the attention mechanism they eliminate the need for recurrence, and attention can potentially be beneficial for our model as well.

Since our model is not recurrent, the benefit of attention can come from learning to attend to observations differently. In the current implementation, the model averages the information from all observations in different channels, then merges these channels by again averaging.

Important information may be lost during this process, which an attention layer may help with by focusing on important observations. An initial attempt has been made in this regard, and while the model is shown to have some learning progress, its prediction quality is significantly behind the currently used model. This is potentially due to using only a single observation which also provides limited important information.

REFERENCES

1. Gibson, J. J., *The Ecological Approach to Visual Perception*, Psychology Press, London, 1986.

2. Heft, H., "Affordances and the Body: An Intentional Analysis of Gibson's Ecological Approach to Visual Perception", *Journal for the Theory of Social Behaviour*, Vol. 19, No. 1, pp. 1–30, 1989.

3. Chemero, A., "An Outline of a Theory of Affordances", *Ecological Psychology*, Vol. 15, No. 2, pp. 181–195, 2003.

4. Sahin, E., M. Cakmak, M. R. Dogar, E. Ugur and G. Ucoluk, "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.

5. Montesano, L., M. Lopes, A. Bernardino and J. Santos-Victor, "Learning Object Affordances: From Sensory–Motor Coordination to Imitation", *IEEE Transactions on Robotics*, Vol. 24, No. 1, pp. 15–26, 2008.

6. Piaget, J. and B. Inhelder, *The Psychology of the Child*, Basic Books, New York, New York, USA, 2008.

7. Garnelo, M., D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. Rezende and S. A. Eslami, "Conditional Neural Processes", *International Conference on Machine Learning (ICML)*, pp. 1704–1713, Stockholm, Sweden, 2018.

8. Fitzpatrick, P., G. Metta, L. Natale, S. Rao and G. Sandini, "Learning About Objects Through Action-Initial Steps Towards Artificial Cognition", *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 3, pp. 3140–3145, Taipei, Taiwan, 2003.

9. Ugur, E., M. R. Dogar, M. Cakmak and E. Sahin, "The Learning and Use of Traversability Affordance Using Range Images on a Mobile Robot", *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1721–1726, Rome, Italy, 2007.

10. Ugur, E., E. Oztop and E. Sahin, "Going Beyond the Perception of Affordances: Learning How to Actualize Them Through Behavioral Parameters", *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4768–4773, Shanghai, China, 2011.

11. Ugur, E., E. Oztop and E. Sahin, "Goal Emulation and Planning in Perceptual Space Using Learned Affordances", *Robotics and Autonomous Systems (RAS)*, Vol. 59, No. 7-8, pp. 580–595, 2011.

12. Kroemer, O., E. Ugur, E. Oztop and J. Peters, "A Kernel-Based Approach to Direct Action Perception", *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2605–2610, St Paul, Minnesota, USA, 2012.

13. Montesano, L., M. Lopes, A. Bernardino and J. Santos-Victor, "Modeling Affordances Using Bayesian Networks", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4102–4107, San Diego, California, USA, 2007.

14. Ugur, E. and J. Piater, "Bottom-up Learning of Object Categories, Action Effects and Logical Rules: From Continuous Manipulative Exploration to Symbolic Planning", *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2627–2633, Seattle, Washington, USA, 2015.

15. Ames, B., A. Thackston and G. Konidaris, "Learning Symbolic Representations for Planning With Parameterized Skills", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 526–533, Madrid, Spain, 2018.

16. Nguyen, A., D. Kanoulas, D. G. Caldwell and N. G. Tsagarakis, "Detecting Ob-

ject Affordances With Convolutional Neural Networks", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2765–2770, Daejeon, Korea, 2016.

17. Do, T.-T., A. Nguyen and I. Reid, "Affordancenet: An End-To-End Deep Learning Approach for Object Affordance Detection", *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5882–5889, Brisbane, Australia, 2018.

18. Mi, J., S. Tang, Z. Deng, M. Goerner and J. Zhang, "Object Affordance Based Multimodal Fusion for Natural Human-Robot Interaction", *Cognitive Systems Research*, Vol. 54, pp. 128–137, 2019.

19. Hamalainen, A., K. Arndt, A. Ghadirzadeh and V. Kyrki, "Affordance Learning for End-To-End Visuomotor Robot Control", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1781–1788, Macau, China, 2019.

20. Chu, F.-J., 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 (RAL)*, Vol. 4, No. 4, pp. 4070–4077, 2019.

21. Thermos, S., G. Potamianos and P. Daras, "Joint Object Affordance Reasoning and Segmentation in RGB-D Videos", *IEEE Access*, Vol. 9, pp. 89699–89713, 2021.

22. Zhang, L., W. Du, S. Zhou, J. Wang and J. Shi, "Inpaint2Learn: A Self-Supervised Framework for Affordance Learning", *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2665–2674, Waikoloa, Hawaii, USA, 2022.

23. Ruiz, E. and W. Mayol-Cuevas, "Geometric Affordance Perception: Leveraging Deep 3D Saliency With the Interaction Tensor", *Frontiers in Neurorobotics*, Vol. 14, p. 45, 2020.

24. Aytar, Y., C. Vondrick and A. Torralba, "See, Hear, and Read: Deep Aligned

Representations", ArXiv:1706.00932 [cs], 2017.

25. Arandjelovic, R. and A. Zisserman, "Objects That Sound", *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 435–451, Munich, Germany, 2018.

26. Noda, K., H. Arie, Y. Suga and T. Ogata, "Multimodal Integration Learning of Robot Behavior Using Deep Neural Networks", *Robotics and Autonomous Systems (RAS)*, Vol. 62, No. 6, pp. 721–736, 2014.

27. Yamada, T., H. Matsunaga and T. Ogata, "Paired Recurrent Autoencoders for Bidirectional Translation Between Robot Actions and Linguistic Descriptions", *IEEE Robotics and Automation Letters (RAL)*, Vol. 3, No. 4, pp. 3441–3448, 2018.

28. Mori, H., M. Masuda and T. Ogata, "Tactile-Based Curiosity Maximizes Tactile-Rich Object-Oriented Actions Even Without Any Extrinsic Rewards", *Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pp. 1–7, Valparaíso, Chile, 2020.

29. Saito, N., D. Wang, T. Ogata, H. Mori and S. Sugano, "Wiping 3D-Objects Using Deep Learning Model Based on Image/Force/Joint Information", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10152–10157, Las Vegas, Nevada, USA, 2020.

30. Saito, N., T. Ogata, S. Funabashi, H. Mori and S. Sugano, "How to Select and Use Tools?: Active Perception of Target Objects Using Multimodal Deep Learning", *IEEE Robotics and Automation Letters (RAL)*, Vol. 6, No. 2, pp. 2517–2524, 2021.

31. Allevato, A., E. Short, M. Pryor and A. Thomaz, "Learning Labeled Robot Affordance Models Using Simulations and Crowdsourcing", *Robotics: Science and Systems (RSS)*, Corvalis, Oregon, USA, 2020.

32. Xu, D., A. Mandlekar, R. Martin-Martin, Y. Zhu, S. Savarese and L. Fei-Fei,

"Deep Affordance Foresight: Planning Through What Can Be Done in the Future", ArXiv:2011.08424 [cs], 2020.

33. Khetarpal, K., Z. Ahmed, G. Comanici, D. Abel and D. Precup, "What Can I Do Here? A Theory of Affordances in Reinforcement Learning", *International Conference on Machine Learning (ICML)*, pp. 5243–5253, Vienna, Austria, 2020.

34. Seker, M. Y., A. E. Tekden and E. Ugur, "Deep Effect Trajectory Prediction in Robot Manipulation", *Robotics and Autonomous Systems (RAS)*, Vol. 119, pp. 173–184, 2019.

35. Khazatsky, A., A. Nair, D. Jing and S. Levine, "What Can I Do Here? Learning New Skills by Imagining Visual Affordances", *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14291–14297, Xi'an, China, 2021.

36. Hochreiter, S. and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

37. Cho, K., B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation", ArXiv:1406.1078 [cs], 2014.

38. Pekmezci, M., E. Ugur and E. Oztop, "Learning System Dynamics via Deep Recurrent and Conditional Neural Systems", *29th IEEE Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, Istanbul, Turkey, 2021.

39. Seker, M. Y., M. Imre, J. H. Piater and E. Ugur, "Conditional Neural Movement Primitives.", *Robotics: Science and Systems (RSS)*, Freiburg im Breisgau, Germany, 2019.

40. Seker, M. Y., A. Ahmetoglu, Y. Nagai, M. Asada, E. Oztop and E. Ugur, "Imitation and Mirror Systems in Robots Through Deep Modality Blending Networks", *Neural Networks*, Vol. 146, pp. 22–35, 2022.

41. Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks From Overfitting", *The Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958, 2014.

42. Sener, M. I., Y. Nagai, E. Oztop and E. Ugur, "Exploration with Intrinsic Motivation using Object-Action-Outcome Latent Space", ArXiv:2008.11503 [cs], 2020.

43. Kingma, D. P. and J. Ba, "Adam: A Method for Stochastic Optimization", ArXiv:1412.6980 [cs], 2014.

44. Hart, P., N. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, pp. 100–107, 1968.

45. Rohmer, E., S. P. N. Singh and M. Freese, "CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013.

46. Elsner, B. and B. Hommel, "Effect Anticipation and Action Control", *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 27, No. 1, p. 229, 2001.

47. Goodale, M. A. and A. D. Milner, "Separate Visual Pathways for Perception and Action", *Trends in Neurosciences*, Vol. 15, No. 1, pp. 20–25, 1992.

48. Griffith, S., J. Sinapov, M. Miller and A. Stoytchev, "Toward Interactive Learning of Object Categories by a Robot: A Case Study With Container and Non-container Objects", *IEEE 8th International Conference on Development and Learning (ICDL)*, pp. 1–6, Shanghai, China, 2009.

49. Cakmak, M., M. Dogar, E. Ugur and E. Sahin, "Affordances as a Framework for Robot Control", *Proceedings of the 7th International Conference on Epigenetic Robotics (EpiRob)*, Lund, Sweden, 2007.

50. Ugur, E., E. Sahin and E. Oztop, "Affordance Learning From Range Data for Multi-Step Planning", *Proceedings of the 9th International Conference on Epigenetic Robotics (EpiRob)*, Venice, Italy, 2009.

51. Hogman, V., M. Bjorkman and D. Kragic, "Interactive Object Classification Using Sensorimotor Contingencies", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2799–2805, Tokyo, Japan, 2013.

APPENDIX A: PERMISSIONS

In this thesis, Figure 4.7, Figure 4.8 and Figure 4.9 are reused from [34]. The license to reuse is presented in Figure A.1.

ELSEVIER LICENSE
TERMS AND CONDITIONS

Aug 01, 2022

This Agreement between Mr. Utku Bozdoğan ("You") and Elsevier ("Elsevier") consists of your license details and the terms and conditions provided by Elsevier and Copyright Clearance Center.

License Number	5360280438571
License date	Aug 01, 2022
Licensed Content Publisher	Elsevier
Licensed Content Publication	Robotics and Autonomous Systems
Licensed Content Title	Deep effect trajectory prediction in robot manipulation
Licensed Content Author	M. Yunus Seker,Ahmet E. Tekden,Emre Ugur
Licensed Content Date	Sep 1, 2019
Licensed Content Volume	119
Licensed Content Issue	n/a
Licensed Content Pages	12
Start Page	173
End Page	184
Type of Use	reuse in a thesis/dissertation
Portion	figures/tables/illustrations

Figure A.1. License document obtained from the publisher.

Number of figures/tables/illustrations	3
Format	both print and electronic
Are you the author of this Elsevier article?	No
Will you be translating?	No
Title	BIDIRECTIONAL MULTI-STEP PREDICTION WITH AFFORDANCES
Institution name	Boğaziçi University
Expected presentation date	Aug 2022
Order reference number	20220808
Portions	Figure 8, Figure 10, Figure 12
Requestor Location	Mr. Utku Bozdoğan Boğaziçi University North Campus BM31 İstanbul, 34337 Turkey Attn: Mr. Utku Bozdoğan
Publisher Tax ID	GB 494 6272 12
Total	0.00 USD
Terms and Conditions	

INTRODUCTION

1. The publisher for this copyrighted material is Elsevier. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at http://myaccount.copyright.com).

GENERAL TERMS

Figure A.1. License document obtained from the publisher (cont.).

2. Elsevier hereby grants you permission to reproduce the aforementioned material subject to the terms and conditions indicated.

3. Acknowledgement: If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source, permission must also be sought from that source. If such permission is not obtained then that material may not be included in your publication/copies. Suitable acknowledgement to the source must be made, either as a footnote or in a reference list at the end of your publication, as follows:

"Reprinted from Publication title, Vol /edition number, Author(s), Title of article / title of chapter, Pages No., Copyright (Year), with permission from Elsevier [OR APPLICABLE SOCIETY COPYRIGHT OWNER]." Also Lancet special credit - "Reprinted from The Lancet, Vol. number, Author(s), Title of article, Pages No., Copyright (Year), with permission from Elsevier."

4. Reproduction of this material is confined to the purpose and/or media for which permission is hereby given.

5. Altering/Modifying Material: Not Permitted. However figures and illustrations may be altered/adapted minimally to serve your work. Any other abbreviations, additions, deletions and/or any other alterations shall be made only with prior written authorization of Elsevier Ltd. (Please contact Elsevier's permissions helpdesk here). No modifications can be made to any Lancet figures/tables and they must be reproduced in full.

6. If the permission fee for the requested use of our material is waived in this instance, please be advised that your future requests for Elsevier materials may attract a fee.

7. Reservation of Rights: Publisher reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

8. License Contingent Upon Payment: While you may exercise the rights licensed immediately upon issuance of the license at the end of the licensing process for the transaction, provided that you have disclosed complete and accurate details of your proposed use, no license is finally effective unless and until full payment is received from you (either by publisher or by CCC) as provided in CCC's Billing and Payment terms and conditions. If full payment is not received on a timely basis, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.

9. Warranties: Publisher makes no representations or warranties with respect to the licensed material.

10. Indemnity: You hereby indemnify and agree to hold harmless publisher and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

11. No Transfer of License: This license is personal to you and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.

Figure A.1. License document obtained from the publisher (cont.).

12. No Amendment Except in Writing: This license may not be amended except in a writing signed by both parties (or, in the case of publisher, by CCC on publisher's behalf).

13. Objection to Contrary Terms: Publisher hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and publisher (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

14. Revocation: Elsevier or Copyright Clearance Center may deny the permissions described in this License at their sole discretion, for any reason or no reason, with a full refund payable to you. Notice of such denial will be made using the contact information provided by you. Failure to receive such notice will not alter or invalidate the denial. In no event will Elsevier or Copyright Clearance Center be responsible or liable for any costs, expenses or damage incurred by you as a result of a denial of your permission request, other than a refund of the amount(s) paid by you to Elsevier and/or Copyright Clearance Center for denied permissions.

LIMITED LICENSE

The following terms and conditions apply only to specific license types:

15. **Translation**: This permission is granted for non-exclusive world **English** rights only unless your license was granted for translation rights. If you licensed translation rights you may only translate this content into the languages you requested. A professional translator must perform all translations and reproduce the content word for word preserving the integrity of the article.

16. **Posting licensed content on any Website**: The following terms and conditions apply as follows: Licensing material from an Elsevier journal: All content posted to the web site must maintain the copyright information line on the bottom of each image; A hyper-text must be included to the Homepage of the journal from which you are licensing at http://www.sciencedirect.com/science/journal/xxxxx or the Elsevier homepage for books at http://www.elsevier.com; Central Storage: This license does not include permission for a scanned version of the material to be stored in a central repository such as that provided by Heron/XanEdu.

Licensing material from an Elsevier book: A hyper-text link must be included to the Elsevier homepage at http://www.elsevier.com . All content posted to the web site must maintain the copyright information line on the bottom of each image.

Posting licensed content on Electronic reserve: In addition to the above the following clauses are applicable: The web site must be password-protected and made available only to bona fide students registered on a relevant course. This permission is granted for 1 year only. You may obtain a new license for future website posting.

17. **For journal authors:** the following clauses are applicable in addition to the above:

Preprints:

A preprint is an author's own write-up of research results and analysis, it has not been peer-reviewed, nor has it had any other value added to it by a publisher (such as formatting, copyright, technical enhancement etc.).

Figure A.1. License document obtained from the publisher (cont.).

Authors can share their preprints anywhere at any time. Preprints should not be added to or enhanced in any way in order to appear more like, or to substitute for, the final versions of articles however authors can update their preprints on arXiv or RePEc with their Accepted Author Manuscript (see below).

If accepted for publication, we encourage authors to link from the preprint to their formal publication via its DOI. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help users to find, access, cite and use the best available version. Please note that Cell Press, The Lancet and some society-owned have different preprint policies. Information on these policies is available on the journal homepage.

Accepted Author Manuscripts: An accepted author manuscript is the manuscript of an article that has been accepted for publication and which typically includes author-incorporated changes suggested during submission, peer review and editor-author communications.

Authors can share their accepted author manuscript:

- immediately
  - via their non-commercial person homepage or blog
  - by updating a preprint in arXiv or RePEc with the accepted manuscript
  - via their research institute or institutional repository for internal institutional uses or as part of an invitation-only research collaboration work-group
  - directly by providing copies to their students or to research collaborators for their personal use
  - for private scholarly sharing as part of an invitation-only work group on commercial sites with which Elsevier has an agreement
- After the embargo period
  - via non-commercial hosting platforms such as their institutional repository
  - via commercial sites with which Elsevier has an agreement

In all cases accepted manuscripts should:

- link to the formal publication via its DOI
- bear a CC-BY-NC-ND license - this is easy to do
- if aggregated with other manuscripts, for example in a repository or other site, be shared in alignment with our hosting policy not be added to or enhanced in any way to appear more like, or to substitute for, the published journal article.

Published journal article (JPA): A published journal article (PJA) is the definitive final record of published research that appears or will appear in the journal and embodies all value-adding publishing activities including peer review co-ordination, copy-editing, formatting, (if relevant) pagination and online enrichment.

Policies for sharing publishing journal articles differ for subscription and gold open access articles:

Subscription Articles: If you are an author, please share a link to your article rather than the full-text. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help your users to find, access, cite, and use the best available version.

Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.

If you are affiliated with a library that subscribes to ScienceDirect you have additional private sharing rights for others' research accessed under that agreement. This includes use for classroom teaching and internal training at the institution (including use in course packs and courseware programs), and inclusion of the article for grant funding purposes.

Figure A.1. License document obtained from the publisher (cont.).

Gold Open Access Articles: May be shared according to the author-selected end-user license and should contain a CrossMark logo, the end user license, and a DOI link to the formal publication on ScienceDirect.

Please refer to Elsevier's posting policy for further information.

18. **For book authors** the following clauses are applicable in addition to the above: Authors are permitted to place a brief summary of their work online only. You are not allowed to download and post the published electronic version of your chapter, nor may you scan the printed edition to create an electronic version. **Posting to a repository:** Authors are permitted to post a summary of their chapter only in their institution's repository.

19. **Thesis/Dissertation**: If your license is for use in a thesis/dissertation your thesis may be submitted to your institution in either print or electronic form. Should your thesis be published commercially, please reapply for permission. These requirements include permission for the Library and Archives of Canada to supply single copies, on demand, of the complete thesis and include permission for Proquest/UMI to supply single copies, on demand, of the complete thesis. Should your thesis be published commercially, please reapply for permission. Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.

Elsevier Open Access Terms and Conditions

You can publish open access with Elsevier in hundreds of open access journals or in nearly 2000 established subscription journals that support open access publishing. Permitted third party re-use of these open access articles is defined by the author's choice of Creative Commons user license. See our open access license policy for more information.

Terms & Conditions applicable to all Open Access articles published with Elsevier:

Any reuse of the article must not represent the author as endorsing the adaptation of the article nor should the article be modified in such a way as to damage the author's honour or reputation. If any changes have been made, such changes must be clearly indicated.

The author(s) must be appropriately credited and we ask that you include the end user license and a DOI link to the formal publication on ScienceDirect.

If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source it is the responsibility of the user to ensure their reuse complies with the terms and conditions determined by the rights holder.

Additional Terms & Conditions applicable to each Creative Commons user license:

CC BY: The CC-BY license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article and to make commercial use of the Article (including reuse and/or resale of the Article by commercial entities), provided the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the work. The full details of the license are available at http://creativecommons.org/licenses/by/4.0.

CC BY NC SA: The CC BY-NC-SA license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article, provided this is not done for commercial purposes, and that the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the

Figure A.1. License document obtained from the publisher (cont.).

Figure A.1. License document obtained from the publisher (cont.).

APPENDIX B: NETWORK INFORMATION

The model architecture is presented below. The object encoder is not visible in Figure B.1, it is a part of action and effect encoders; since the model can also function as a single channel. The action encoder architecture is presented in Figure B.2, the effect encoder in Figure B.3, the object encoder in Figure B.4 and the decoder in Figure B.5. The reachability classifier model is presented in Figure B.6.

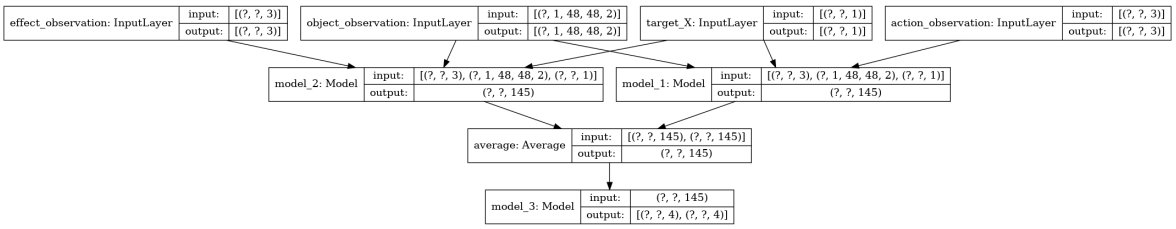


Figure B.1. Overview of the architecture.

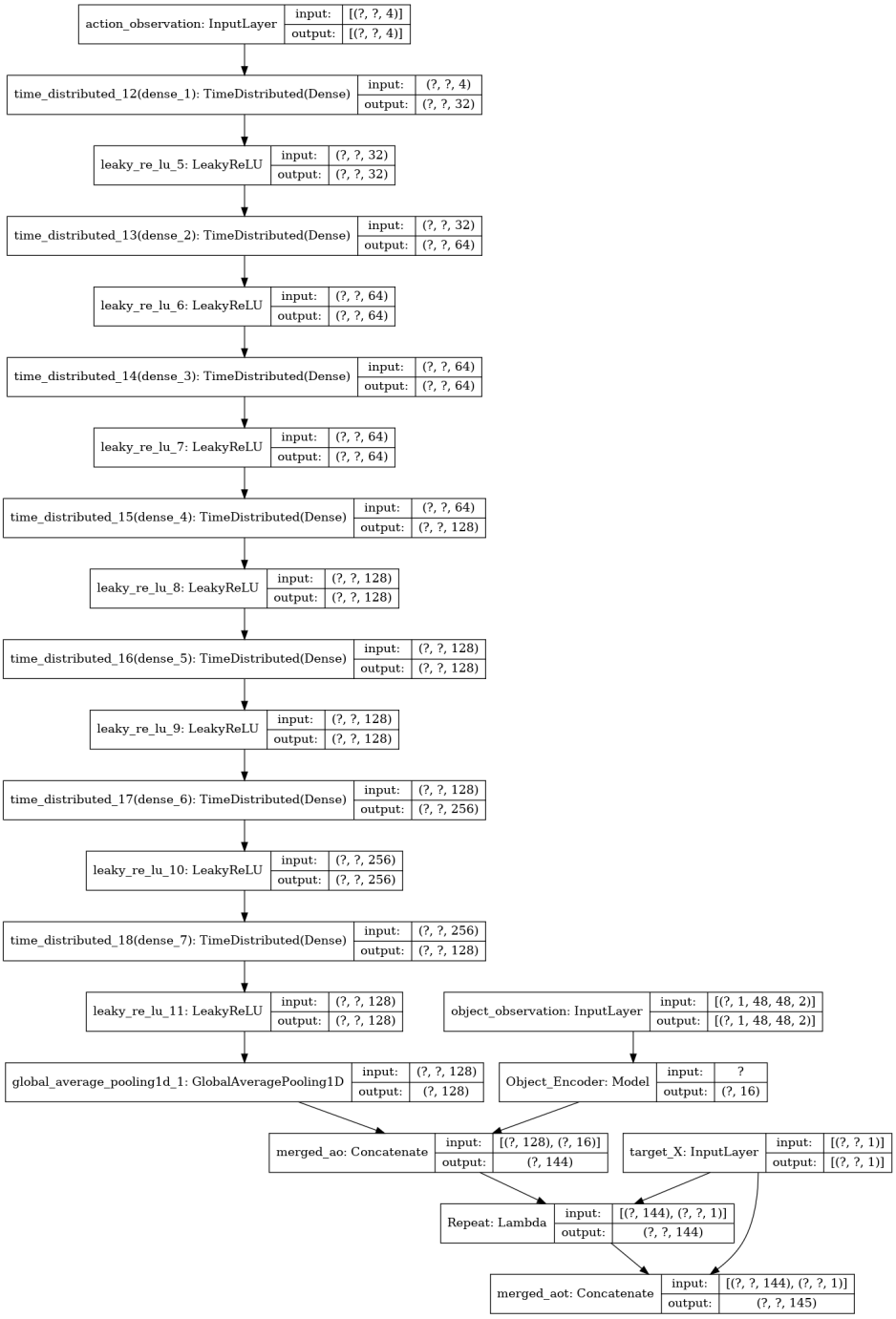


Figure B.2. Action encoder architecture.

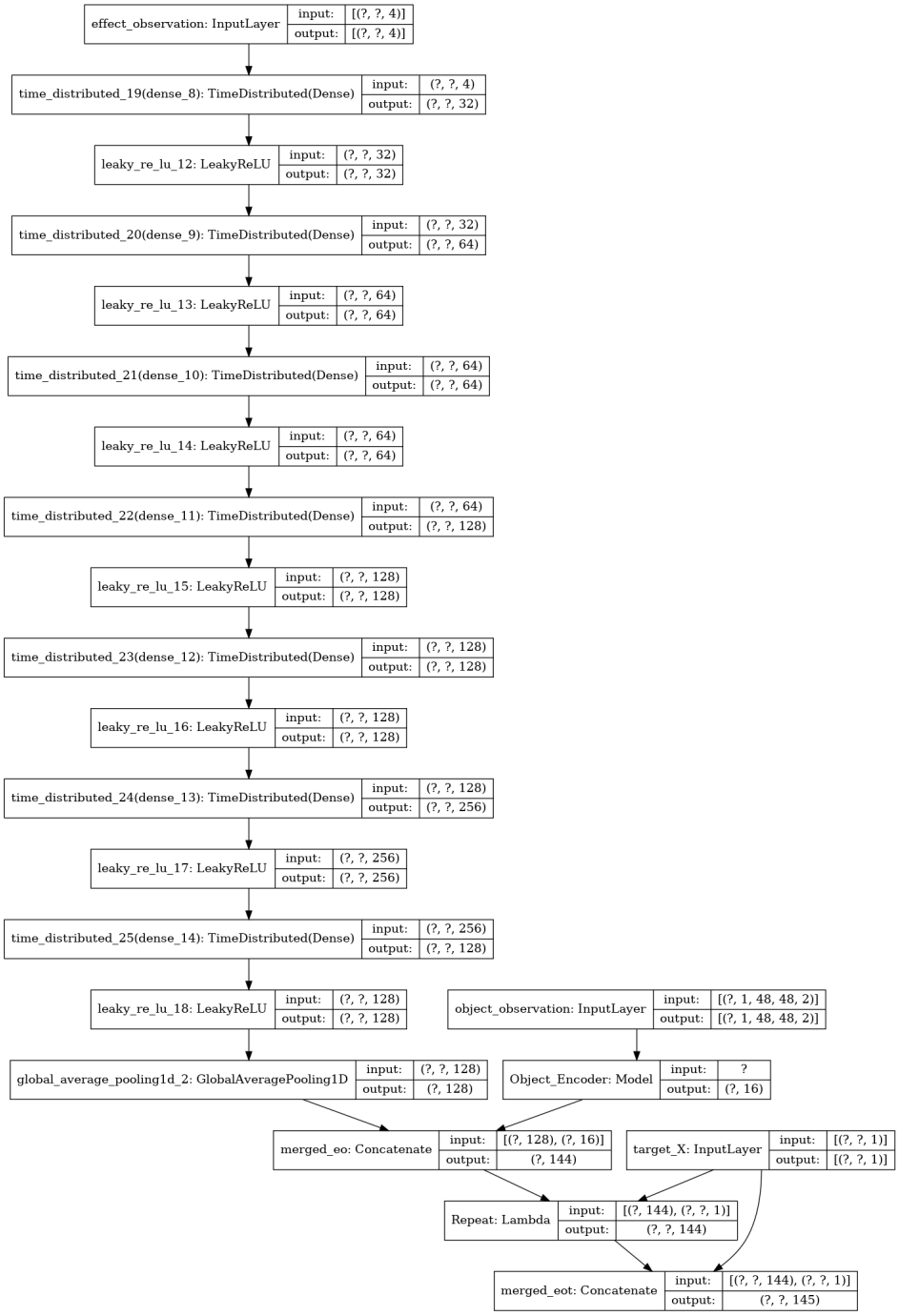


Figure B.3. Effect encoder architecture.

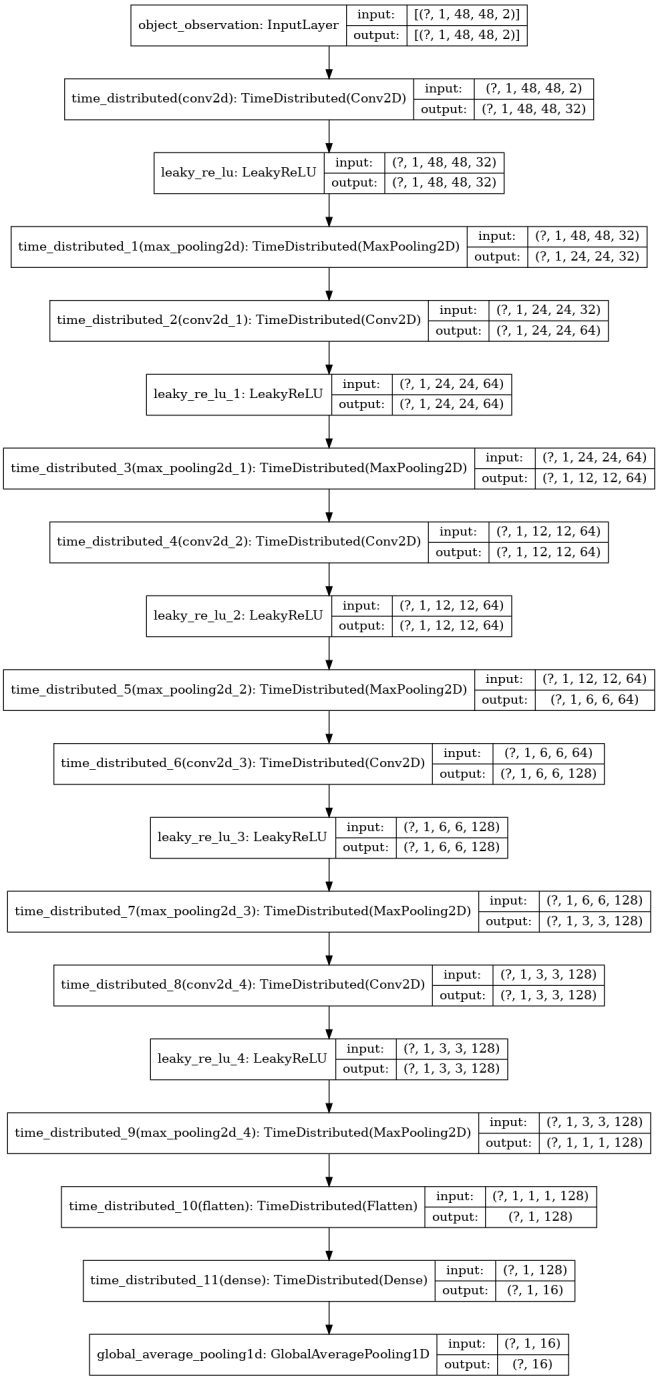


Figure B.4. Object encoder architecture.

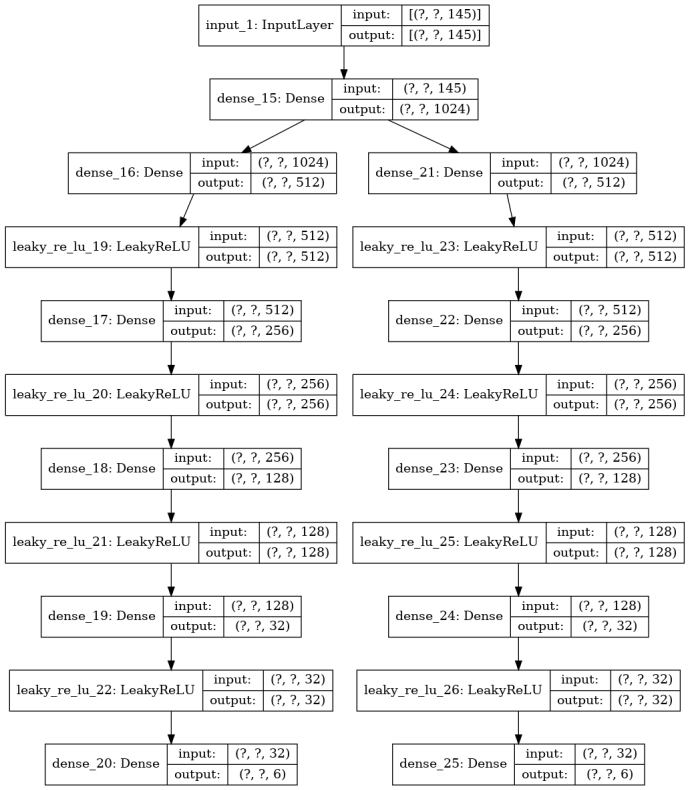


Figure B.5. Decoder architecture showing both the action and effect decoders.

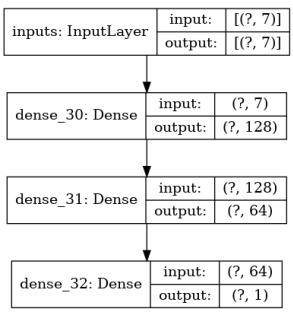


Figure B.6. Overview of the reachability classifier.