This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

Multi-Object Graph Affordance Network: Goal-Oriented Planning through Learned Compound Object Affordances

Tuba Girgin[1,2] and Emre Uğur[1]

***Abstract*—Learning object affordances is an effective tool in the field of robot learning. While the data-driven models investigate affordances of single or paired objects, there is a gap in the *exploration* of affordances of compound objects composed of an arbitrary number of objects. We propose the Multi-Object Graph Affordance Network which models complex compound object affordances by learning the outcomes of robot actions that facilitate interactions between an object and a compound. Given the depth images of the objects, the object features are extracted via convolution operations and encoded in the nodes of graph neural networks. Graph convolution operations are used to encode the state of the compounds, which are used as input to decoders to predict the outcome of the object-compound interactions. After learning the compound object affordances, given different tasks, the learned outcome predictors are used to plan sequences of stack actions that involve stacking objects on top of each other, inserting smaller objects into larger containers and passing through ring-like objects through poles. We showed that our system successfully modeled the affordances of compound objects that include concave and convex objects, in both simulated and real-world environments. We benchmarked our system with a baseline model to highlight its advantages.**

***Index Terms*—Robot learning, affordance learning, graph neural network (GNN) architecture.**

I. Introduction

THE affordances concept, introduced by J.J. Gibson to refer to the action possibilities provided by the environment [1], has been significantly influential in robotics research [2], [3]. The developmental aspects of affordances have been widely adopted in robot learning research [4]–[7]. While previous works have examined the affordances of single or paired simple object interactions, the affordances of compound objects composed of an arbitrary number of objects with concave shapes and varying sizes have not been sufficiently studied [8].

Consider an infant trying to build a tower with its toys. Because of the different shapes and sizes of the objects, each toy would afford different actions, and with each action, various effects would be generated. The affordances of the objects may change according to their relations with the other objects in the environment, i.e., while an empty cup affords insertability, it might loose this affordance if one or more objects are put inside the cup or a large box is stacked on the cup.

This work was supported by INVERSE project (101136067) funded by the European Union.

[1]Department of Computer Engineering, Bogazici University, Istanbul, Turkey. [2]Robotics and Autonomous Systems Laboratory, TUBITAK BILGEM, 41470 Kocaeli, Turkey

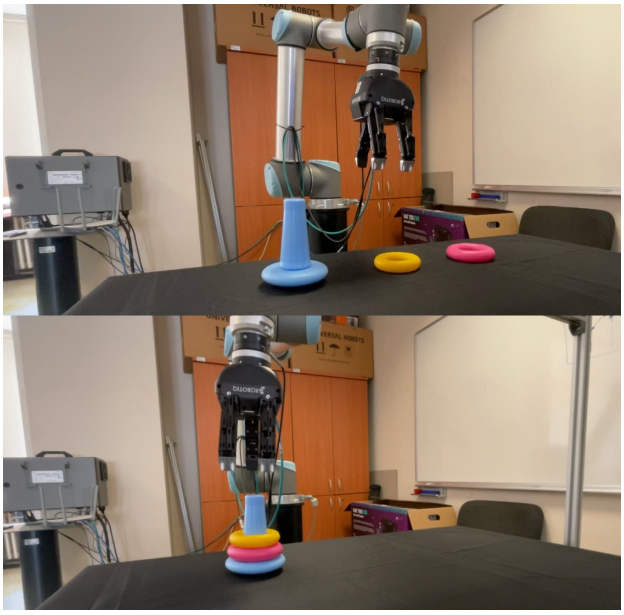

Fig. 1. Execution of the plan generated using our MOGAN model to build the shortest compound object given a pole and two rings in the real world setup. The agent uses the pole as the base and stacks the rings, as they do not change the height of the compound.

However, if a small object is inserted in a big cup or several large rings are stacked on the cup, the cup would remain insertable. Predicting the affordance of the compound object is not straightforward, as the affordance of a compound object not only depends on the affordances of the included objects but is also determined based on in which order these objects are placed (e.g. via releasing the objects on the compound) and the relative positions of all objects. In order to address the challenge of learning compound object affordances, we propose to represent the objects in the compound as a graph, as the graph representation preserves spatial relations between objects, and can be used to propagate information along the chain of objects in the compound, enabling effective reasoning for the complete structure.

Graph Neural Networks (GNNs) [9] are effective for learning meaningful representations of structures and their relations. Consequently, they have gained extensive adoption in action recognition problems [10], natural language processing [11], navigation problems to learn relations between pedestrians, objects, and robots [12]–[14], as well as reasoning about relations between multi-object systems [15]–[18]. In these studies, the

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

representation capacity of GNNs for spatial relations involving an unlimited number of inputs is exploited, whereas other commonly used feed-forward networks lack this property.

Affordances refer to the relations between objects, actions and effects [4], we aim to learn to predict effects given objects and actions. In our study, we propose the Multi-Object Graph Affordance Network (MOGAN), which learns affordances of compound objects, i.e. learns predicting effects of actions applied on objects and/or object compounds. The prediction is done using features obtained from graph representations utilizing GNNs. In this study, we focus on actions that facilitate interactions between objects. For this purpose, we used actions that pick up objects and place them on top of other objects or object compounds. The effects are encoded as the spatial displacements between the released objects and the objects in the compound structure. With concave objects, visibility can be altered when stacking, as the line of sight may intersect another object. In contrast, with convex objects, there is no need to measure visibility. Consequently, when dealing with concave objects, traditional effect representation based solely on their center points is insufficient to reason about visibility effects. Therefore, more complex effect representations are required to accurately convey information about the visibility of concave objects. As a result, a suitable novel effect representation is used.

We designed six different tasks using an inventory of convex and concave objects of varying sizes, including poles, cups and rings of different sizes, boxes, and balls. The learned affordances correspond to forward predictors, and therefore can be used for goal-oriented action selection and planning. We first presented the effect prediction results of compound object interactions in the Pybullet simulation environment. Subsequently, we discussed the success rates of plans generated through model predictions. Finally, we demonstrated the realization of the generated plans in the simulation environment, resulting in the construction of unseen structures composed of available objects. The results of our model were compared with those of the baseline model which corresponds to DeepSym model [19]. While Deepsym is the state-of-the-art model for learning action-object-effect relational categories, it is limited to paired object interactions. We modified the Deepsym model to handle multi-object representations to benchmark it against our proposed model. We also demonstrated the applicability of our system by executing tasks with the UR10 manipulator in a real-world setting.

In summary, this paper introduces the MOGAN model, a novel approach for learning compound object affordances. Our contributions can be summarized as follows:

- **Proposal of Multi-Object Graph Affordance Network:** We introduce a novel model, MOGAN, which *encodes* the affordances of compounds composed of varying number of objects by representing them as a graph structure without the need for supervision from experts or labeled affordances. Our model learns these affordances through the observed effects of robot interactions.
- **Introduction of a Novel Effect Encoding Method:** We represent the effects of robot interactions with a particular encoding that takes into account 3D spatial relations. Widely used effects, such as displacement of the centers of the objects, are insufficient to explain the semantic behavior of concave objects.
- **Demonstration of Applicability:** We showed the applicability of our system by successfully accomplishing various tasks in both the Pybullet simulation environment and the real world using the UR10 manipulator.

II. Related Work

A. Affordances

The study of affordances [3], [4], [20], [21] has attracted significant attention in recent years, with embodied AI studies utilizing affordances to evaluate language model generations. For instance, Ahn et al. [22] proposed SayCan, where they combined the skill affordances of a robot with Language Models (LLM) to ground the instructions to the environment. Additionally, Ahn et al. [23] introduced AutoRT, where affordance filtering is utilized to align the tasks generated by vision-language models (VLMs) with the robot's capabilities and safety regulations. However, these studies primarily focused on extracting skill affordances by combining language model outputs with predefined instructions and rules, such as "do not lift heavy objects." In contrast, the concept in our study revolves around exploring the affordances of objects through robot interactions and observed effects, which are influenced by also the weights of the objects.

Some approaches learned visual affordances to understand applicable actions through neural networks employed in computer vision. [24] Qian et al. [25] combined large-scale vision language models with an image encoder and an affordance decoder network to predict an affordance map based on the queried action. Birr et al. [26] extracted affordances of detected objects through queries using a predefined prompt list of affordances in ChatGPT, an AI tool developed by OpenAI. Do et al. [27] detected affordances of objects in images along with their classes, with affordances being labeled at the pixel level. Cuttano et al. [28] proposed a model based on CLIP [29] that grounds task-agnostic affordances of texts from an open vocabulary onto images. Depth values of object images are also utilized to define affordance relations. Toumpa and Cohn [30] defined affordance relations concerning the concaveness of objects. While they measure concaveness by inspecting depth values based on [31], we use deep autoencoders to learn features of objects that are not limited to concavity.

Various robotics research benefited affordances to enhance precision in grasping, picking, and placing operations [32]–[34]. Hart et al. [35] designed a ROS package enabling the operators to specify grasp poses. Corona et al. [36] proposed a comprehensive network called GanHand, where the hand shape and pose are predicted through the reconstruction of the object alongside the predicted grasp type. [37] studies mechanisms that produce hierarchical structuring of affordance learning tasks of different levels of complexity. Mandikal and Grauman [38] learned grasping policies utilizing 3D thermal affordance maps. Zeng et al. [39] benefitted from labeled affordance maps for improved grasping. Learning contact information as affordances, as studied in [40] and [41], is

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

another way to tackle the existing problems. Cheng et al. [42] also learned the contact points of two objects for picking, grasping, and regrasping operations. Lin et al. [43] learned pixel-wise pick-and-place affordances by generating 3D imaginary scenes from 2D images using an annotated dataset. Borja-Diaz et al. [44] designed a self-supervised affordance learning model that labels gripper open and close points while the robot is controlled through human teleoperation. Mees et al. [45] extended this work, grounding large language models to robotic applications. While these studies supervise their models to learn affordances using expert annotations, contact points, and gripper signals, we observe the effects of the manipulator's actions to explore the affordances. Cruz et al. [46] used affordances in an interactive reinforcement learning setup in order to speed up the learning of skills from other agents.

Multiple approaches have studied the exploration of affordances learning effects through interactions [47], [48]. Mar et al. [49] explored affordance categories according to the effects of tool usage. They mapped the features extracted from the observations and affordance classes discovered by clustering the effects with the k-means algorithm [50]. Antunes et al. [51] defined affordances as the probability of effects given the object features, tool features, and action. With the formulation of goals as symbols, they achieved probabilistic planning. Saponaro et al. [52] exploit the affordances learned from robot interactions to interpret and describe human actions in light of their own experience. [19], [53] also performed sub-symbolic and symbolic planning [54], using affordances of only single or paired objects.

While the affordances of objects with concave and convex shapes, such as mugs and spoons, are learned with the supervision of experts, *exploring* and *discovering* the affordances of complex structures generated by combining (inserting, passing through, stacking) of a sequence of such concave and convex objects through observed effects has not been studied to the best of our knowledge. We study the discovery of affordances of compounds, including concave shapes like rings, poles, and cups. In our study, we also adapt GNNs to exploit their representation capacity for compound object affordances.

B. Graph Neural Networks

Graph Neural Networks have been used in a wide range of domains to model systems composed of multiple parts such as the human body for human action understanding [55], the human hand for gesture inference [56], the electroencephalogram (EEG)-related measurements for emotion recognition [57], and the images with multiple entities for inferring the relations between them [58]. In robotics, on the other hand, the representation capability of GNNs for an unlimited number of objects and their relationships has enabled their widespread adoption [59]–[62]. Lou et al. [63] depicted densely clustered objects as graph structures and extracted the adjacency and occlusion relations. They then utilized GNNs to learn grasping poses for target objects, taking into account the spatial relations with other objects. Wilson and Hermans [64] utilized GNNs in conjunction with CNNs to encode their multi-object environment for more accurate reward calculation during policy training. Lin et al. [65] devised a graph structure in which objects and the goal positions for pick and place tasks are connected by edges. Subsequently, the learned GNN policy selects object and goal nodes to execute the steps of desired tasks. Huang et al. [66] represented multi-object scenes as fully connected graph structures based on partial observations and learned the relations between nodes as logical symbols using GNN classifiers. In contrast, our study reasons relations between objects by learning observed effects of robot actions without defining logical symbols.

GNNs are also commonly used for modeling dynamics of multi-object systems [67]. Driess et al. [68] employed GNNs to capture the dynamics between multiple objects for novel scene synthesis using Neural Radiance Fields (NeRF) [69]. Tekden et al. [70], [71] introduced a learnable physics engine where objects are represented as graph structures, and the relations between them are classified at the edge level. With estimated relations between objects, future states are predicted based on the applied actions. However, the objects are simple-shaped, and their features are restricted to position and radius values. Additionally, only actions that push the objects in the horizontal plane are considered.

Overall, in our study, compound objects are represented as graph structures, their features are learned utilizing GNNs, and the affordances are learned through effect predictions. Our system plans a sequence of actions (selecting an object to place it on the compound object) with a search algorithm utilizing the learned affordances.

III. Method

Our proposed method models the affordances of compound objects, which are composed of an arbitrary number of objects that are placed on top of each other. Given the compound object and a new object, it learns to predict the effects generated by placing the new object on top of the compound object. In our framework, an affordance, which is denoted as A, is defined as the relation between the compound object (T) that resides on the table, the object (o) to be placed on top of the compound object, and the effects (E_1, E_2, E_3) generated: $A = (T, o, (E_1, E_2, E_3))$. Given T and o, our system is expected to learn E_1, E_2, and E_3. For learning, at the start of each exploration cycle, the size of the object compound (T) is initialized as 0. Then, the robot randomly selects and picks up an object (o), places it on top of the current object compound, and observes the effects (E_1, E_2, E_3) until either the new object falls down or the object compound collapses.

In the rest of this section, we first describe how compound and single objects (T and o) and effects (E_1, E_2, E_3) are represented, and the details of the learning algorithm. Finally we describe how the learned affordances can be used to make plans in order to achieve different goals.

A. Single Object Representation

The single objects are represented by features extracted from their depth images using an autoencoder. The encoder component of the autoencoder takes in a 32x32 normalized

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

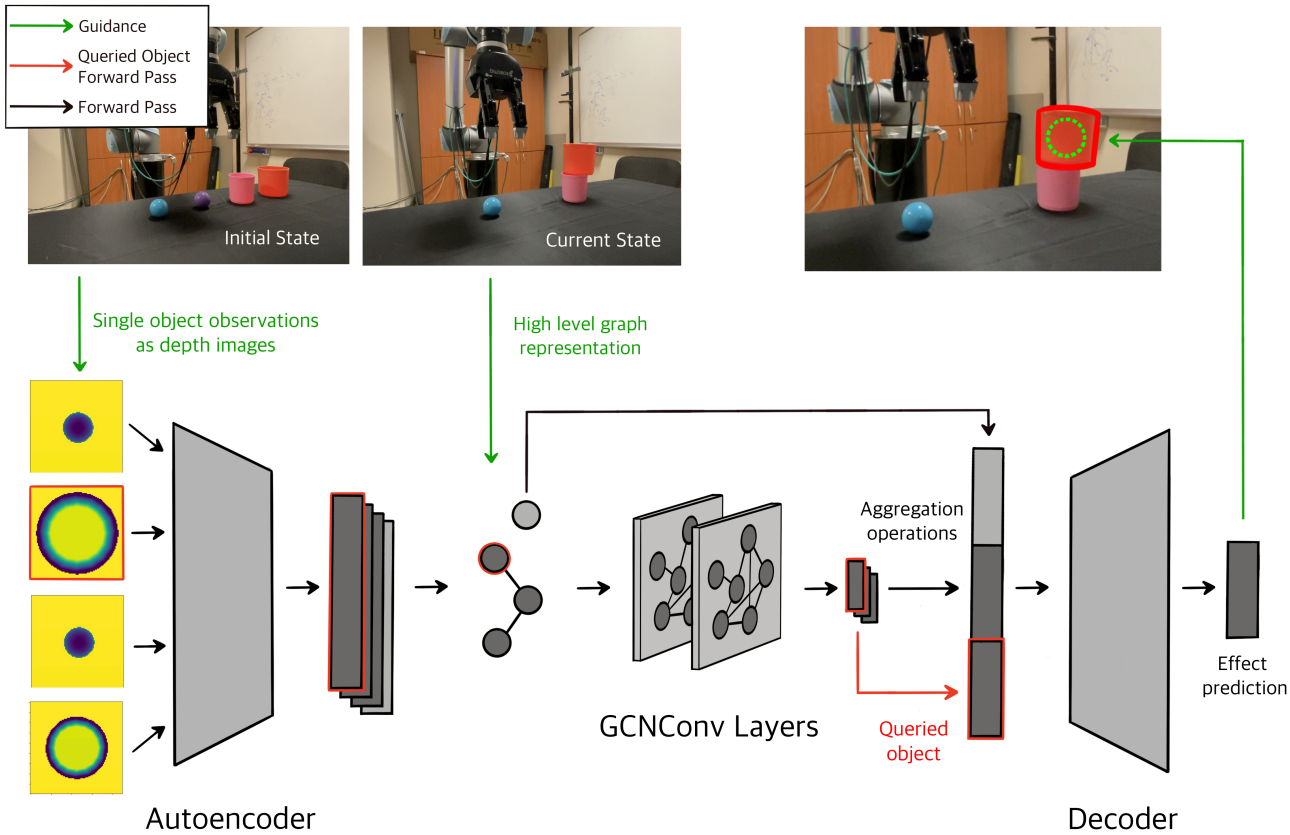

Fig. 2. MOGAN: Multi-Object Graph Affordance Network Architecture, along with the pretrained autoencoder. The depth images of single objects are encoded with the autoencoder. It then constructs the graph representation of the compound object. The proposed model, MOGAN, extracts meaningful features from the graph and predicts the resulting effect between a single object and a queried object within the compound object. The predicted effect is visually depicted in the rightmost image with a dashed green circle.

depth image and comprises three linear layers with neuron sizes of 256, 256, and 64, respectively. Empirically, a latent space size of 4 was found to be sufficient for representing the images of the object set used in this study. The decoder part of the autoencoder is not utilized in the MOGAN model because we only need the latent vector; therefore, it is not shown in Figure 2. The decoder depicted in the figure represents the decoder component of the MOGAN model. The autoencoder is trained with the single depth images collected from both the simulation environment and the real world until convergence.

In our system, the encoder component of the autoencoder is used to extract latent representations from the single images. These latent representations are then used to construct graph representations for the compounds, allowing the MOGAN model to learn the effects from them. The maximum and minimum values of the depth images are appended to the latent representations to prevent the information loss caused by the normalization operation.Therefore, single objects o and their features, such as size, shape, and concaveness, are represented by a learned feature vector of size 6.

B. Compound Object Representation

The compound objects are composed of different objects placed on top of each other. In order to represent a compound object, both the features of the single objects inside the compound and the spatial relations between the objects are required to be used. For this, we utilize a graph-based structure. A graph, denoted as G, is defined as a tuple of nodes N and edges E.

$$G = (N, E) \tag{1}$$

$$N = \{n_1, n_2, ..., n_k\}, n_i \in \mathbb{R}^p, 1 \leq i \leq k \tag{2}$$

$$E = \{e_1, e_2, ..., e_{k-1}\}, e_i \in \mathbb{R}^q, 1 \leq i \leq k-1. \tag{3}$$

Each node, n, consists of the object features acquired through the autoencoder. p and q denote the size of the feature vector of a node and an edge, respectively, while k indicates the number of nodes within the graph, with no specific limits on this count. A directed edge between two nodes is defined when objects are placed consecutively in the tower, with the direction going from one object to the one placed before it, providing a hint for the spatial relations between the objects in the compound. In Section V-C, we also designed an edge creation algorithm for non-linear compounds to preserve these spatial relations. All nodes form self-connections.

C. Effect Representation

When an object is placed on the compound object, different types of effects, such as insertion in different ways, stacking, or toppling, are observed. Instead of categorizing each effect instance into a pre-defined effect category, we propose a

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

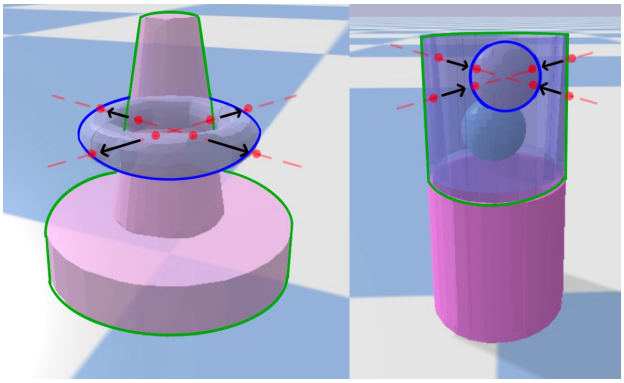

Fig. 3. Visualization of the calculation of lateral spatial displacements: Imaginary rays are projected through the center of the new object. Red points illustrate the intersections with both the compounding object and newly added object. The black arrows are calculated by the function s.

generic continuous effect representation that captures the 3D spatial relations between the placed object and each object in the compound. In other words, the effect represents the spatial outcome of placing the new object on the compound and is encoded as a combination: (E_1, E_2, E_3). E_1 describes the height differences between the top and bottom surfaces of each object pairs, considering their bounding boxes.

$$E_1 = \{E_1^1, E_1^2, .., E_1^k\}, E_1^i \in \mathbb{R}^2, 1 \leq i \leq k \tag{4}$$

$$E_1^i = \{s(|z_i^+ - z_{k+1}^+|), s(|z_i^- - z_{k+1}^-|)\} \tag{5}$$

E_1^i correpsponds to the effect between the new object and the ith object in the object compound. z^+ and z^- describe the maximum and minimum height values of an object derived from the bounding box of the object. s is a sign function that assigns signs to the effect values. The function s creates a vector starting from the top surface of the ith object to the top surface of the new object, using the center points of the surfaces. If the vector points toward the center of the ith object, the sign is considered negative. This comparison is performed for all the bounding box faces. E_2 encodes the lateral spatial differences between objects. The differences are calculated by sending imaginary rays through the new object, as shown in Fig. 3. If the ray does not intersect with the interested object (outlined with green color), the relevant effect becomes 0. The signs of the differences are calculated with the sign function s considering the points that the imaginary rays cut. The intersection points for each object are calculated using the $rayTestBatch$ function in PyBullet. This function provides the coordinates and object IDs where the ray intersects at each point. We use this function to find the red points indicated in Figure 3.

$$E_2 = \{E_2^1, E_2^2, .., E_2^k\}, E_2^i \in \mathbb{R}^4, 1 \leq i \leq k \tag{6}$$

$$E_2^i = \{s(|x_i^+ - x_{k+1}^+|), s(|x_i^- - x_{k+1}^-|), \tag{7}$$

$$s(|y_i^+ - y_{k+1}^+|), s(|y_i^- - y_{k+1}^-|)\} \tag{8}$$

$$E_3 = \begin{cases} 1, & \text{if } pos(o_i) \geq t_1 || ori(o_i) \geq t_2, 1 \leq i \leq k+1 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Finally, E_3 encodes whether the newly placed object falls down or the compound object collapses/topples when the new object is placed on top. The pos and ori functions get the x-y position and orientation of a given object o_i and compare them with the base of the compound. They return the sum of the differences in these values. The thresholds t_1 and t_2 are used to determine whether the position and orientation values indicate a collapse.They are chosen empirically as 20 cm and 60 degrees, respectively.

D. Multi-Object Graph Affordance Network (MOGAN)

The proposed MOGAN model, shown in Fig. 2, outputs the effects (E_1, E_2, E_3) expected to be generated when a new object (o_{k+1}) is placed on the compound object (T). As the compound object was formed by placing the objects one by one on top of each other, the depth images and the corresponding autoencoder features $(n_1, n_2, ..n_k)$ were already collected and available for processing for a compound with a size of k. The autoencoder features of the new object to be placed on the compound object are also processed and is represented as n_{k+1}. Our system, MOGAN, comprises the encoder part of the pretrained autoencoder, GCNConv layers, and a linear decoder. The depth image features are extracted by the encoder. A high-level graph representation is then constructed based on the current compound (T), as explained in Section III-B, utilizing corresponding depth image features. Subsequently, two GCNConv [72] layers process the graph representation of the compound to generate a latent representation. The mean and maximum values of these latent representations for all nodes are calculated and aggregated. The features of the newly added object (n_{k+1}), the aggregated latent representation, and the latent representation of the queried object are concatenated. The decoder, consisting of three linear layers, takes this concatenated input to predict the effects between the queried object $(o_i, 1 \leq i \leq k)$ and the new object (o_{k+1}) placed on top of the compound. In our system, the parameter size for the network is 46786, and Leaky ReLU is utilized as the activation function between the layers.

E. Planning and Tasks

We aim to provide a variety of tasks to demonstrate the prediction capacity of the MOGAN for planning to achieve different goals. The goals include obtaining object compounds of specified heights, structures, and sub-structures. A tree search algorithm is realized to discover the optimal plan to achieve a specific goal. At each iterative step, the graph representation of the existing object compound is generated, and the object that will be placed on the tower is encoded. Three MOGAN networks predict the effects E based on the graph representation of the compound and the feature vector of the new object. If predicted E_3 indicates a fall/collapse, the current branch of the search operation is terminated.

In detail, six different tasks can be specified. The first two tasks correspond to building the tallest and shortest compounds/towers. In order to predict the height of the object compounds, the E_1 effect predictions are summed up. The third and fourth tasks correspond to obtaining structures where

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

TABLE I
COMPARISON OF PLAN SUCCESS RATES IN THE SIMULATION ENVIRONMENT WITH THE MULTI-OBJECT DEEPSYM AS BASELINE (MDS)

Size	**Task 1: Tallest**		**Task 2: Shortest**		**Task 3: Occluded**		**Task 4: Occluding**		**Task 5: Specific Height**		**Task 6: Condition**	
	MOGAN	MDS	MOGAN	MDS	MOGAN	MDS	MOGAN	MDS	MOGAN	MDS	MOGAN	MDS
2	100	80	100	80	100	80	100	80	100	90	100	100
3	100	60	100	80	100	90	100	70	90	60	100	60
4	90	50	90	80	80	80	90	60	90	60	90	60
5	80	10	90	60	90	40	80	60	80	50	90	60

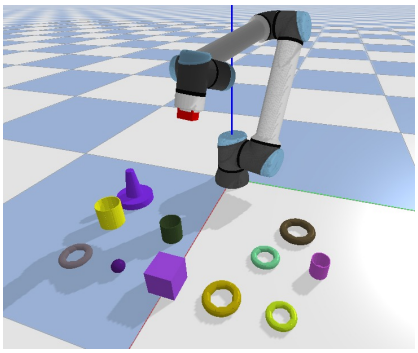

Fig. 4. A PyBullet environment featuring a UR10 robot and various objects, including cubes, poles, balls, cups, and rings.

TABLE II
OBJECT DETAILS

Name	Size (Height, Width, Depth) (cm)	Number
Pole	(17, 14, 14)	1
Ball	(5, 5, 5)	5
Cube	(10, 10, 10)	1
Ring	(3, 12, 12)	1
Ring	(2.5, 10.5, 10.5)	1
Ring	(2.4, 9.7, 9.7)	1
Ring	(2, 9, 9)	1
Ring	(1.5, 8, 8)	1
Cup	(10, 10.5, 10.5)	1
Cup	(8.5, 7.5, 7.5)	1
Cup	(7.5, 6.5, 6.5)	1

the placed objects are required to enclose the top part of the object compound and become obstructed in the compound (inserted inside). The accumulated E_2 predictions are used for this purpose. The fifth task corresponds to building a tower of a specific height. Finally, the sixth task enables the selection of two objects from the set of objects that will be used in the object compound and puts constraints on their relative placements, such as maximizing or minimizing their relative distances.

Fig. 5. Various objects used in the real-world setup: a pole, rings, cups, a cube, and balls.

IV. Experimental Setup

In the real-world experiment setup, we employ a 7-DOF UR10 manipulator equipped with a Robotiq 3-Finger Adaptive Robot Gripper. The objects chosen for the experiment are selected from a variety of toys commonly played with by infants, enabling the exploration of affordance relations involving concave, convex objects, and compounds. As indicated in Table II, there are more small objects than large ones. This is because it becomes infeasible to find a suitable plan for the end-effector of the manipulator when dealing with compounds that are too tall. However, our approach does not impose a limitation on the number of objects in a compound. Positioned 1 meter above the table, centered, a Realsense RGBD camera is installed, with its lens directed downward to optimize capture. The Pybullet environment is used for simulating actions and interactions. A custom gripper is attached to the wrist of the UR10 manipulator in the simulator in order to speed up the pick and place action executions. The objects used in the simulator are created using Blender, taking into account their size information. These objects are depicted in Fig. 5 and Fig. 4. The depth images of the simulated scene are captured using a virtual depth camera positioned at the same relative location as in the real-world experimental setup.

During experiments, a subset of the inventory is spawned in a rectangular area at random positions. The depth image of the scene is segmented to acquire the depth images of the objects individually. In order to segment the depth image, the lowest values are grouped according to the pixel positions. The image is cropped according to the center pixel positions for each group. The positions of the objects are calculated using the center pixel positions and values and used during pick and place action executions. The trajectory for the manipulator to pick up and place an object is planned using MoveIt in the real-world setup, while built-in inverse kinematics functions are utilized in the simulation. The calculated positions for the objects are used as the positional goal for the end-effector during the pick operation. For the place operation, an additional 15 cm height is added to the goal position to eliminate potential object collisions. The orientation of the end-effector remains the same during operations.

A data point consists of: 1) a 32x32 single object depth image, 2) 32x32 individual depth images for the objects in the compound, and 3) effects as explained in the Method Section. The depth images for the objects in the compound are derived from previous iterations. The training dataset comprises of

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

TABLE III
PREDICTION ERRORS FOR THE UNSEEN SIMULATION DATA IN DECIMETERS

Tower Size	Test Errors		
	Effect 1 (dm)	Effect 2 (dm)	Effect 3
1	0.000	0.000	0.010
2	0.008	0.000	0.099
3	0.022	0.004	0.149
4	0.043	0.003	0.177
5	0.063	0.002	0.177
6	0.085	0.002	0.184
7	0.093	0.001	0.240
8	0.109	0.001	0.145
9	0.126	0.000	0.185
10	0.141	0.000	0.231
11	0.134	0.000	0.306
12	0.123	0.000	0.322
13	0.092	0.000	0.330
14	0.108	0.000	0.252

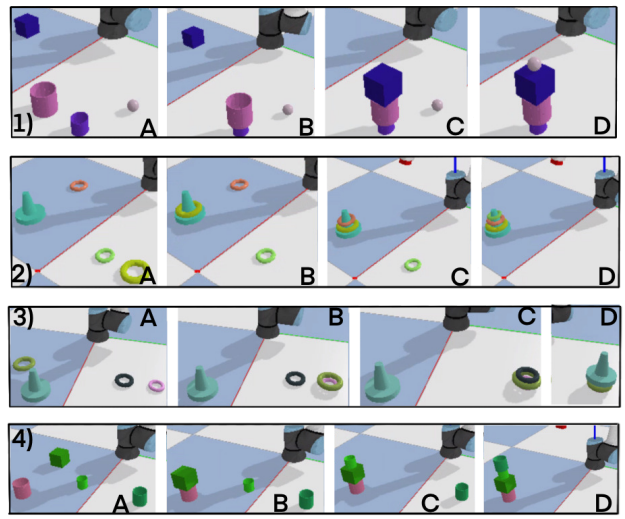

Fig. 6. A number of sample plan executions in the simulator. The tasks are (1) to minimize the invisibility of the given objects, (2) to build the shortest compound object using a pole and different sized rings, (3) to build the tallest compound object using a pole and different sized rings, and (4) to build a compound object given a constraint between the pink and dark green cups.

5000 data points acquired from simulation experiments. Prior to training, representations of both single and compound objects are acquired using the pretrained autoencoder.

A MOGAN model is initialized with two GCNConv layers and three linear layers. The parameter size of the model is 46786 which is empirically found to prevent over fit. The model weights are randomly initialized with a torch seed value of 42. Mean Squared Error (MSE) loss and a custom sign loss are utilized as the loss functions. The sign loss, used for E_1 and E_2, penalizes predictions that do not align with the correct signs compared to the ground truth data. The Adam optimizer is employed as optimization algorithm. The model is trained for 600 epoch with a batch size of 1. The learning rate starts from 10^{-4} and gradually decreased with the learning rate scheduler. The gamma value is set to 0.95, and the step size is 500.

To demonstrate the efficiency of our proposed model, we compared it with a modified version of DeepSym [19]. While Deepsym, a deep encoder-decoder network, learns relational symbols through effect predictions, the model is limited to paired object interactions. In the modified version, we encoded individual depth images and concatenated them. The size of the tensor is the multiplication of the feature size and the maximum object number in a compound. The maximum object number extracted from the dataset is 14 in our case. The remaining part of the input tensor remains 0 for the smaller-sized object compounds. In contrast to DeepSym, we did not utilize the Gumbel-Sigmoid function in the latent space, as our study does not focus on discrete symbol learning. Since we define our actions as adding a new object, we did not query additional actions. The decoder part learns the concatenation of all effects for each node in a compound.

The parameter size for the baseline model is 50178, which is close to but not less than the parameter size of our proposed model. Training and test results are compared with the MOGAN model in the Experiments and Results Section.

V. Results

In this section, we analyze the prediction error of our model for the unseen combinations of the composite objects and provide the results in Table III. The errors are grouped according to the compound object sizes to analyze the relation between compound object sizes and prediction errors. Effect 1 is the predicted height differences between two objects, as explained in the Method Section. The inventory contains objects with maximum, minimum, and mean height values of 17 cm, 1.5 cm, and 6.5 cm, respectively. The errors in Effect 1 predictions result in deviations of less than 1 cm in predicted height differences when the compound object size is 8 or less. If the object size exceeds 8, we observe a maximum error of 1.41 cm. Although these prediction errors do not significantly impact the majority of predictions due to the presence of considerably larger objects, they can lead to failures when predicting effects between smaller objects, such as small rings. The error for Effect 2 does not increase along with the compound object size. We can confidently state that our model is capable of predicting x and y displacements of objects without being affected by the number of objects. The ground truth value of Effect 3 is 1 when the tower collides, 0 otherwise. When we inspect the prediction errors for Effect 3, we see that it increases as the number of objects increases. However, the errors in Effect have minimal impact on the overall results due to the margin between ground truth values.

A. Simulation Experiments & Comparison with Baseline

We evaluate the generated plans for six different tasks, we sample 10 different configurations for each compound object size, ranging from 2 to 5 in the simulator, as shown in Table I. For the fifth task, which is to build a compound object with a desired height, we calculated possible height values for the sampled configuration, selected one as the goal, and compared it to the resulting height. For the last task, we randomly selected two objects from the sampled set of objects to maximize or minimize their distances. Please see the generated and executed plans for a number of sample tasks

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

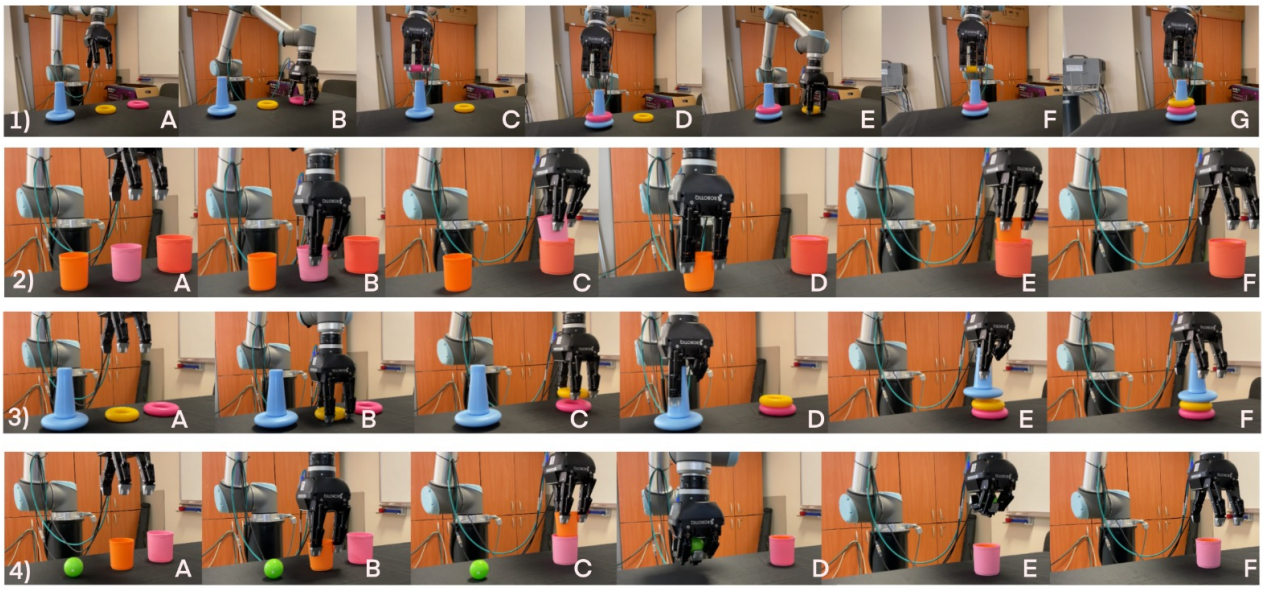

Fig. 7. A number of snapshots from real-world planning experiments. In the first, second, and fourth images, the objective is to construct the shortest compound objects. In the third image, the goal is to create the tallest compound object. The system observes the scene, predicts the effects of each potential plan using MOGAN, and executes the optimal one.

in Fig 6. In the 2nd and 3rd rows of Fig 6, different tasks are assigned for the same set of objects. In the 2nd row, the model benefits the *passability* of rings onto the pole to keep the compound short. In the 3rd row, the model first benefits the *stackability* of rings and then stacks the pole to increase the height of the compound.

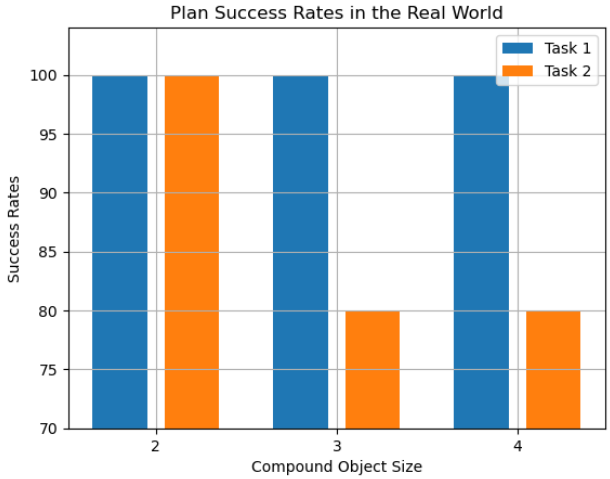

Fig. 8. Plan success rates in the real world. The goals are to build the shortest and tallest compound objects. 5 trials were conducted for each set of different sizes.

Out of 300 planning tasks, our system was able to generate 283 successful plans, as shown in Table I. The success rate was observed to slightly drop when the number of objects increases. This was an expected result, as the number of objects in the compound increase, predicting the affordance of the compound object and how it is affected from placing another object on top becomes more difficult. Additionally, as the number of objects increases, the number of predictions done during the planning increases exponentially. One erroneous prediction among all the correct predictions may cause a failure in planning. It is important to note that our MOGAN model significantly outperformed the base model in planning, as shown in Table I, showing the effectiveness of using graph structures where the features of the objects in the compound are embedded in the nodes of the GNN for modeling multi-object affordances and for the multi-object planning problems.

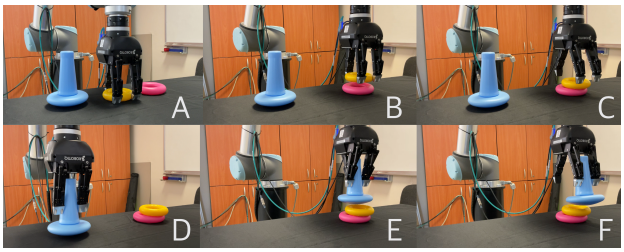

Fig. 9. A failure case is illustrated here. When the goal is to build the tallest compound, the MOGAN model chooses to place the yellow ring and the pole on top of the pink ring, respectively. However, the pole gets squeezed between the fingers of the 3-finger gripper, resulting in failure, as shown in part F.

B. Real-world Experiments

In the real-world setup, we test our system's planning capacity with the first two tasks: building the shortest and the tallest compound objects. We sampled 5 sets of objects for the compound object sizes 2, 3, and 4. A number of plan execution snapshots from sampled tasks are provided in Fig.7. Out of the 30 real-world planning tasks, 28 of the generated plans were found to be successful, as shown in Figure 8. The system is able to build desired compound objects 1) using the depth images from Realsense, 2) predicting effects with

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

TABLE IV
PREDICTION ERRORS FOR THE UNSEEN SIMULATION DATA IN DECIMETERS

Tower Size	Test Errors								
	Effect 1 (dm)			Effect 2 (dm)			Effect 3		
	MOGAN	MOGAAN	MOFFAN	MOGAN	MOGAAN	MOFFAN	MOGAN	MOGAAN	MOFFAN
1	0.036	0.037	0.035	0.000	0.000	0.000	0.013	0.010	0.011
2	**0.041**	0.042	0.050	0.001	0.002	0.001	0.019	0.0535	0.015
3	**0.023**	**0.023**	0.156	0.000	0.000	0.001	0.0521	0.204	0.033
4	**0.044**	0.073	0.067	0.000	0.001	0.000	**0.186**	0.253	0.323

the MOGAN model, 3) planning an optimal path with the tree search algorithm, and 4) executing it with the UR10 manipulator. The success rate slightly decreases as the object number in the inventory increases. Another reason for the failure is the unpredictability of the real-world systems. Since the objects we use are plastic, they exhibit slight elasticity. Therefore, the object models in the simulation do not fully capture the physical properties of real-world objects. This can lead to unexpected results during the gripper's open and close operations. In Figure 9, when the robot holds the pole, it grips it too tightly. As a result, when the gripper opens, the pole gets stuck between the fingers and does not fall.

C. Building Nonlinear Compounds

In this section, we analyze experiments involving additional stacking actions. These actions include changes in position and orientation. In this setup, the agent can choose from 3 different locations along the x-axis and 2 different orientations (upside and downside), resulting in a total of 6 different actions. Orientation information is included in the feature vector to encode the objects' orientation, thus generating node features. To connect the nodes, edges are created between them following the Algorithm 1. $checkIntersections$ function returns $True$ if the bounding boxes of the objects intersect. The $checkConnectivity$ function determines whether two objects are in contact. If either of these functions returns $True$, an edge is added between the two objects.

Algorithm 1 Edge Creation

List of object IDs id_list
$edge_list \leftarrow [\,]$
for $i = 1$ to $length(id_list) - 1$ **do**
  $a \leftarrow id_list[i]$
  $b \leftarrow id_list[i+1]$
  $intersections \leftarrow checkIntersections(a, b)$
  $connectivity \leftarrow checkConnectivity(a, b)$
  if $intersections$ or $connectivity$ **then**
    append($edge_list$, (a, b))
  end if
end for

In this setup, two additional models are designed for comparison with the proposed MOGAN model. The first is the Multi-Object Graph Attentive Affordance Network (MOGAAN), where we utilize GATConv layers instead of GCNConv layers. The second is the Multi-Object Feed Forward Network (MOFFAN), where we use linear layers for encoding.

TABLE V
COMPARISON OF PLAN SUCCESS RATES WITH MOGAAN AND MOFFAN

	Planning Success	No Solution	Failure
MOGAN	100	0	0
MOGAAN	60	40	0
MOFFAN	70	0	30

In the latter, we concatenate the object features according to the object selection order. To train and test the models with the new actions, we collected a dataset of 3000 compounds consisting of cubes and cups, using 500 of them for testing. The MOGAN model is initialized as described in Section IV, with the input sizes adjusted to match the feature sizes of the dataset, and the x-axis position conditioned in the latent space. The parameter sizes for the other models are 47618, 48290 respectively. The training parameters for this experiment are also described in Section IV.

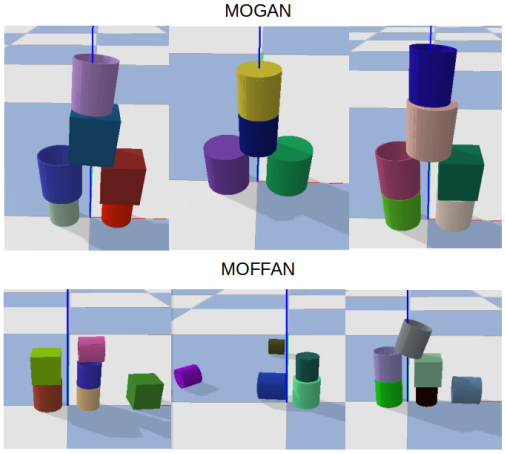

Fig. 10. Three experiments where the MOFFAN model fails are shown in comparison to the MOGAN model. The MOGAN model can reason about the lengths of the legs of a compound to successfully stack another object onto the middle, whereas the MOFFAN model cannot.

The test errors for this experiment are provided in Table IV. The errors are grouped according to the compound object sizes. The errors for predicted height differences are shown under the Effect 1 column, where the Effect 2 column corresponds to the errors for lateral differences, and the Effect 3 column corresponds to the prediction errors for collapse. The comparisons with MOGAAN and MOFFAN indicate that our model outperforms the baselines, especially as the compound size increases.

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

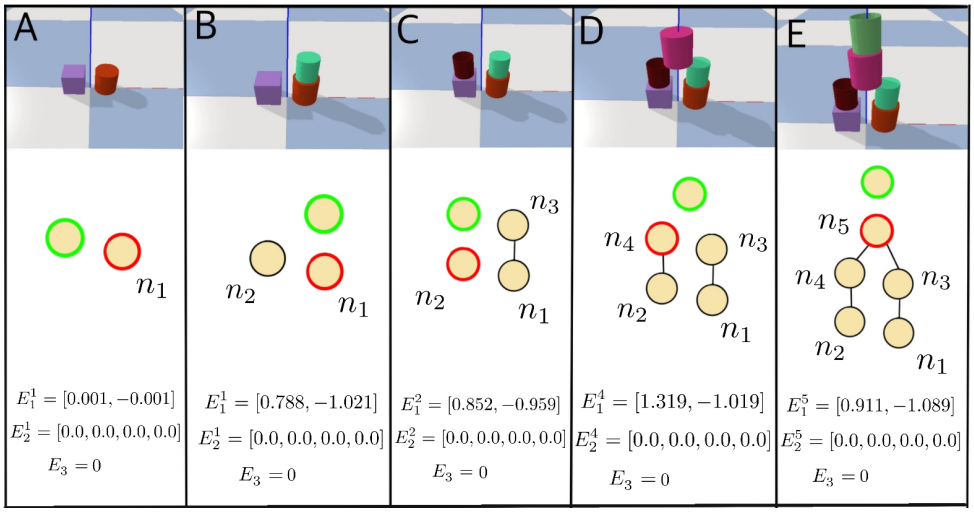

Fig. 11. An example of online graph generation and effect prediction for a new test set containing six different stacking actions is shown. The rotation of the objects is encoded in the node features, while the x-axis position is conditioned in the latent space. In the image, the effects of the green-circled object corresponding to the red-circled objects are displayed.

For this experiment, we designed a task where the goal is to build a compound shaped like a bridge. When a list of 6 objects is presented, the models are required to find the stacking order and orientations for the predefined locations that result in a bridge shape. Out of 10 planning tasks, our proposed model, MOGAN, was able to plan compounds for all the desired tasks, and the plans were executed successfully in the simulation environment. The MOGAAN model could not generate plans for 4 of the tasks. The MOFFAN model generated 10 plans, but 3 of them collapsed in the simulation environment as shown in Table V. Figure 10 shows a comparison of the plans where the MOFFAN fails. While the MOGAN model builds legs of the bridge with similar sizes to stack another object in the middle, the MOFFAN model cannot build legs with similar sizes. This demonstrates that the MOFFAN model lacks the capability to reason about the relationship between leg sizes and stability, whereas our model can reason about the affordances of the bridge legs to construct a stable bridge.

In the following part of this section, we analyze two case studies to better understand the capabilities of the MOGAN model.

1) Case Study 1: A compound building process is shown in Figure 11, along with depiction of the graph structures and predicted effects. The figure shows the effect predictions for the new objects (green circles) in relation to the objects corresponding to the red-circled nodes. In part B of the figure, a cup is stacked onto a rotated cup. The model can reason about the rotations of objects, leading to accurate effect predictions. In parts C,D, and E, it is shown that the MOGAN model can reason about the spatial relations of objects in graphs with various edge connections, leading to accurate effect predictions when a new object is to be stacked.

2) Case Study 2: In this experiment, while building a compound, a rotated cup covers a smaller cup as shown in part C of Figure 12. Note that for all objects with different orientations, depth images are taken in the initial position and features are extracted by the autoencoder. Then, the object's orientation information is appended to the feature vector. The MOGAN model is able to learn the continuous effects E_1 and E_2 for the "covering" effect, as shown in part C of the figure, where the predicted effects indicate intersecting bounding boxes, with the top and side of the red-circled object remaining inside the newly added green-circled object. Additionally, the model predicts effect E_3 as $True$ before the stacking action when the height difference between the subcompounds is too large to place a new object in the middle, demonstrating its ability to reason about spatial relations between nodes in a graph to predict the collapse effect, as shown in part D of Figure 12.

VI. Conclusion

In this research, we proposed a novel Multi-Object Graph Affordance Network, MOGAN, which models affordances of compound objects for manipulation and planning. We showed that our system was able to correctly predict the affordances of compound objects that include spheres, cups, poles, and several rings that enclose the poles. This prediction capability was effectively used to build different structures via planning structures of highly complex affordances. In the future, we plan to discover symbolic affordances of compound structures and utilize AI planners for task realization. Additionally, to enable an end-to-end and generalizable approach, we intend to use point clouds and RGB images to represent our objects and depth values to retrieve effect of actions.

References

[1] J. J. Gibson, "The theory of affordances," *Hilldale, USA*, vol. 1, no. 2, pp. 67–82, 1977.

[2] E. Rome, L. Paletta, E. Şahin, G. Dorffner, J. Hertzberg, R. Breithaupt, G. Fritz, J. Irran, F. Kintzler, C. Lörken *et al.*, "The macs project: an approach to affordance-inspired robot control," in *Towards Affordance-Based Robot Control: International Seminar, Dagstuhl Castle, Germany, June 5-9, 2006. Revised Papers*. Springer, 2008, pp. 173–210.

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

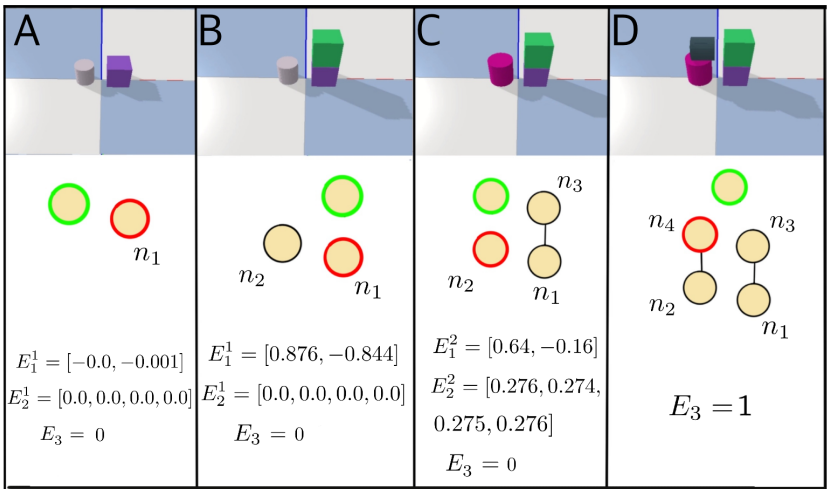

Fig. 12. An example of online graph generation and effect prediction for collision detection is provided. Since the pink cup covers the gray cup, the lengths of the compound's legs vary, preventing further stacking.

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

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

[5] 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.

[6] E. Ugur, Y. Nagai, E. Sahin, and E. Oztop, "Staged development of robot skills: Behavior formation, affordance learning and imitation with motionese," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 2, pp. 119–139, 2015.

[7] E. Ugur and J. Piater, "Emergent structuring of interdependent affordance learning tasks," in *4th International Conference on Development and Learning and on Epigenetic Robotics*. IEEE, 2014, pp. 489–494.

[8] P. Zech, S. Haller, S. R. Lakani, B. Ridge, E. Ugur, and J. Piater, "Computational models of affordance in robotics: a taxonomy and systematic classification," *Adaptive Behavior*, vol. 25, no. 5, pp. 235–271, 2017.

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[10] T. Ahmad, L. Jin, X. Zhang, S. Lai, G. Tang, and L. Lin, "Graph convolutional neural network for human action recognition: A comprehensive survey," *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 128–145, 2021.

[11] L. Wu, Y. Chen, K. Shen, X. Guo, H. Gao, S. Li, J. Pei, B. Long *et al.*, "Graph neural networks for natural language processing: A survey," *Foundations and Trends® in Machine Learning*, vol. 16, no. 2, pp. 119–328, 2023.

[12] Z. Liu, Y. Zhai, J. Li, G. Wang, Y. Miao, and H. Wang, "Graph relational reinforcement learning for mobile robot navigation in large-scale crowded environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8776–8787, 2023.

[13] Y. Zhou and J. Garcke, "Learning crowd behaviors in navigation with attention-based spatial-temporal graphs," *arXiv preprint arXiv:2401.06226*, 2024.

[14] E. Escudie and L. M. J. Saraydaryan, "Attention graph for multi-robot social navigation with deep reinforcement learning," in *AAMAS*, 2024.

[15] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende *et al.*, "Interaction networks for learning about objects, relations and physics," in *Advances in neural information processing systems*, 2016, pp. 4502–4510.

[16] M. B. Chang, T. Ullman, A. Torralba, and J. B. Tenenbaum, "A compositional object-based approach to learning physical dynamics," *arXiv preprint arXiv:1612.00341*, 2016.

[17] Y. Li, J. Wu, J.-Y. Zhu, J. B. Tenenbaum, A. Torralba, and R. Tedrake, "Propagation networks for model-based control under partial observation," in *International Conference on Robotics and Automation*, 2019, pp. 1205–1211.

[18] A. E. Tekden, A. Erdem, E. Erdem, T. Asfour, and E. Ugur, "Object and relation centric representations for push effect prediction," *arXiv preprint arXiv:2102.02100*, 2021.

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

[20] X. Yang, Z. Ji, J. Wu, and Y.-K. Lai, "Recent advances of deep robotic affordance learning: a reinforcement learning perspective," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 15, no. 3, pp. 1139–1149, 2023.

[21] H. Min, R. Luo, J. Zhu, S. Bi *et al.*, "Affordance research in developmental robotics: A survey," *IEEE TCDS*, vol. 8, no. 4, pp. 237–255, 2016.

[22] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[23] M. Ahn, D. Dwibedi, C. Finn, M. G. Arenas, K. Gopalakrishnan, K. Hausman, B. Ichter, A. Irpan, N. Joshi, R. Julian *et al.*, "Autort: Embodied foundation models for large scale orchestration of robotic agents," *arXiv preprint arXiv:2401.12963*, 2024.

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

[25] S. Qian, W. Chen, M. Bai, X. Zhou, Z. Tu, and L. E. Li, "Affordancellm: Grounding affordance from vision language models," *arXiv preprint arXiv:2401.06341*, 2024.

[26] T. Birr, C. Pohl, A. Younes, and T. Asfour, "Autogpt+ p: Affordance-based task planning with large language models," *arXiv preprint arXiv:2402.10778*, 2024.

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

[28] C. Cuttano, G. Rosi, G. Trivigno, and G. Averta, "What does clip know about peeling a banana?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2238–2247.

[29] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[30] A. Toumpa and A. G. Cohn, "Object-agnostic affordance categorization via unsupervised learning of graph embeddings," *Journal of Artificial Intelligence Research*, vol. 77, pp. 1–38, 2023.

[31] M. Leyton, "A process-grammar for shape," *Artificial Intelligence*, vol. 34, no. 2, pp. 213–247, 1988.

[32] 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*. IEEE, 2014, pp. 476–481.

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Cognitive and Developmental Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2024.3520086

[33] S. Hangl, E. Ugur, S. Szedmak, and J. Piater, "Robotic playing for hierarchical complex skill learning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2799–2804.

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

[35] S. Hart, P. Dinh, and K. Hambuchen, "The affordance template ros package for robot task programming," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6227–6234.

[36] E. Corona, A. Pumarola, G. Alenya, F. Moreno-Noguer, and G. Rogez, "Ganhand: Predicting human grasp affordances in multi-object scenes," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5031–5041.

[37] E. Ugur and J. Piater, "Emergent structuring of interdependent affordance learning tasks using intrinsic motivation and empirical feature selection," *IEEE TCDS*, vol. 9, no. 4, pp. 328–340, 2016.

[38] P. Mandikal and K. Grauman, "Learning dexterous grasping with object-centric visual affordances," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 6169–6176.

[39] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *The International Journal of Robotics Research*, vol. 41, no. 7, pp. 690–705, 2022.

[40] G. Schiavi, P. Wulkop, G. Rizzi, L. Ott, R. Siegwart, and J. J. Chung, "Learning agent-aware affordances for closed-loop interaction with articulated objects," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5916–5922.

[41] Y. Geng, B. An, H. Geng, Y. Chen, Y. Yang, and H. Dong, "Rlafford: End-to-end affordance learning for robotic manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5880–5886.

[42] S. Cheng, K. Mo, and L. Shao, "Learning to regrasp by learning to place," in *Conference on Robot Learning, 8-11 November 2021, London, UK*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 2021, pp. 277–286.

[43] Y.-C. Lin, P. Florence, A. Zeng, J. T. Barron, Y. Du, W.-C. Ma, A. Simeonov, A. R. Garcia, and P. Isola, "Mira: Mental imagery for robotic affordances," in *Conference on Robot Learning*. PMLR, 2023, pp. 1916–1927.

[44] J. Borja-Diaz, O. Mees, G. Kalweit, L. Hermann, J. Boedecker, and W. Burgard, "Affordance learning from play for sample-efficient policy learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6372–6378.

[45] O. Mees, J. Borja-Diaz, and W. Burgard, "Grounding language with visual affordances over unstructured data," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 576–11 582.

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

[47] B. Moldovan, P. Moreno, M. Van Otterlo, J. Santos-Victor, and L. De Raedt, "Learning relational affordance models for robots in multi-object manipulation tasks," in *2012 ieee international conference on robotics and automation*. IEEE, 2012, pp. 4373–4378.

[48] B. Moldovan and L. De Raedt, "Occluded object search by relational affordances," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 169–174.

[49] T. Mar, V. Tikhanoff, G. Metta, and L. Natale, "Self-supervised learning of grasp dependent tool affordances on the icub humanoid robot," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 3200–3206.

[50] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[51] A. Antunes, L. Jamone, G. Saponaro, A. Bernardino, and R. Ventura, "From human instructions to robot actions: Formulation of goals, affordances and probabilistic planning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5449–5454.

[52] G. Saponaro, L. Jamone, A. Bernardino, and G. Salvi, "Beyond the self: Using grounded affordances to interpret and describe others' actions," *IEEE TCDS*, vol. 12, no. 2, pp. 209–221, 2019.

[53] E. Ugur, E. Şahin, and E. Oztop, "Unsupervised learning of object affordances for planning in a mobile manipulation platform," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4312–4317.

[54] T. Taniguchi, E. Ugur, M. Hoffmann, L. Jamone, T. Nagai, B. Rosman, T. Matsuka, N. Iwahashi, E. Oztop, J. Piater *et al.*, "Symbol emergence in cognitive developmental systems: a survey," *IEEE transactions on Cognitive and Developmental Systems*, vol. 11, no. 4, pp. 494–516, 2018.

[55] A. R. Shahid, M. Nawaz, X. Fan, and H. Yan, "View-adaptive graph neural network for action recognition," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 15, no. 2, pp. 969–978, 2022.

[56] C. Li, S. Li, Y. Gao, X. Zhang, and W. Li, "A two-stream neural network for pose-based hand gesture recognition," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 4, pp. 1594–1603, 2021.

[57] W. Kong, M. Qiu, M. Li, X. Jin, and L. Zhu, "Causal graph convolutional neural network for emotion recognition," *IEEE TCDS*, vol. 15, no. 4, pp. 1686–1693, 2022.

[58] J. Zhu and H. Wang, "Multiscale conditional relationship graph network for referring relationships in images," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 2, pp. 752–760, 2021.

[59] A. Iriondo, E. Lazkano, and A. Ansuategi, "Affordance-based grasping point detection using graph convolutional networks for industrial bin-picking applications," *Sensors*, vol. 21, no. 3, p. 816, 2021.

[60] H. Tan, L. W. 0003, Q. Zhang, Z. Gao, N. Zheng, and G. Hua, "Object affordances graph network for action recognition." in *BMVC*, 2019, p. 145.

[61] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6541–6548.

[62] M. Kulshrestha and A. H. Qureshi, "Structural concept learning via graph attention for multi-level rearrangement planning," *Conference on Robot Learning (CoRL)*, 2023.

[63] X. Lou, Y. Yang, and C. Choi, "Learning object relations with graph neural networks for target-driven grasping in dense clutter," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 742–748.

[64] M. Wilson and T. Hermans, "Learning to manipulate object collections using grounded state representations," in *Conference on Robot Learning*. PMLR, 2020, pp. 490–502.

[65] Y. Lin, A. S. Wang, E. Undersander, and A. Rai, "Efficient and interpretable robot manipulation with graph neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2740–2747, 2022.

[66] Y. Huang, A. Conkey, and T. Hermans, "Planning for multi-object manipulation with graph neural network relational classifiers," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1822–1829.

[67] S. Tian, Y. Cai, H.-X. Yu, S. Zakharov, K. Liu, A. Gaidon, Y. Li, and J. Wu, "Multi-object manipulation via object-centric neural scattering functions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9021–9031.

[68] D. Driess, Z. Huang, Y. Li, R. Tedrake, and M. Toussaint, "Learning multi-object dynamics with compositional neural radiance fields," in *Conference on robot learning*. PMLR, 2023, pp. 1755–1768.

[69] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[70] A. E. Tekden, A. Erdem, E. Erdem, M. Imre, M. Y. Seker, and E. Ugur, "Belief regulated dual propagation nets for learning action effects on groups of articulated objects," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 556–10 562.

[71] A. E. Tekden, A. Erdem, E. Erdem, T. Asfour, and E. Ugur, "Object and relation centric representations for push effect prediction," *Robotics and Autonomous Systems*, vol. 174, p. 104632, 2024.

[72] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

Authorized licensed use limited to: UNIVERSITA TRENTO. Downloaded on January 16,2025 at 10:44:24 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.