Robotic Learning the Sequence of Packing Irregular Objects from Human Demonstrations

André Santos\textsuperscript{1}, Atabak Dehban\textsuperscript{1} and José Santos-Victor\textsuperscript{1}

Abstract—We address the unsolved task of robotic bin packing with irregular objects, such as groceries, where the underlying constraints on object placement and manipulation, and the diverse objects’ physical properties make preprogrammed strategies unfeasible. Our approach is to learn directly from expert demonstrations in order to extract implicit task knowledge and strategies to achieve an efficient space usage, safe object positioning and to generate human-like behaviors that enhance human-robot trust.

We collect and make available a novel and diverse dataset, BoxED, of box packing demonstrations by humans in virtual reality. In total, 263 boxes were packed with supermarket-like objects by 43 participants, yielding 4644 object manipulations.

We use the BoxED dataset to learn a Markov chain to predict the object packing sequence for a given set of objects and compare it with human performance. Our experimental results show that the model surpasses human performance by generating sequence predictions that humans classify as human-like more frequently than human-generated sequences.

I. INTRODUCTION

The ceaseless digitalisation of the modern world has steadily transformed many aspects of our daily lives, including the grocery shopping experience. In recent years it too has begun the shift from the physical retail spaces into the online environment, with companies such as Ocado\textsuperscript{1}, a dedicated online grocery retailer, reporting 2.5 billion GBP of revenue in 2021. Behind this trend lays the complex logistic challenge of packing all orders in shipping containers.

Concurrently, robots have also steadily been entering our workplaces and homes. The so-called “cobots”\textsuperscript{2} have attracted a considerable amount of research and development attention, with the introduction of Amazon’s new Astro robot\textsuperscript{3} and Temi robot\textsuperscript{4}. Besides the critical need to ensure a safe operation while interacting with humans, they also must act as human-like as possible, such that their actions can be understood and anticipated by their human owners, providing users with an improved interaction experience.

The task that motivated our work lies at the intersection of these two rapidly expanding fields: a robot that packs groceries alongside and learns from a human collaborator. There exists already appropriate hardware solutions, envisaged for robot-human collaboration, that feature special adaptations such as flexible joints. Furthermore, the grasp inference literature contains many algorithms with impressive accuracy\textsuperscript{5}.

However, there is very little literature on how to pack heterogeneous objects such as those found in a supermarket, and there are even less publicly available data on this task. Existing methods that address packing objects inside a container consider essentially simple shape objects (such as cuboids and spheres) and often deploy search-based algorithms to determine the best placement pose, requiring multiple hours to find a solution\textsuperscript{6},\textsuperscript{7}. Methods that predict a packing sequence given a set of objects (for instance, the objects in an online order) generally consider only cuboid objects\textsuperscript{8},\textsuperscript{9}.

Our contributions are twofold:

1) We learn a model to predict the packing sequence, based on Markov chains, that extracts implicit task knowledge directly from humans. The generated sequences lead to safe and efficient positioning of the objects and are classified by experts as human-like sequences.

2) We introduce BoxED: Box packing with Everyday items Dataset\textsuperscript{1} the first publicly available collection of human experts packing groceries into a box. BoxED was collected in Virtual Reality (VR), as shown in Fig.\textsuperscript{1} and captures many parameters of this task, including 6-DOF pick-and-place grasp poses, object trajectories, packing sequence and more. This dataset enables learning models for multiple aspects of this task from humans, that generate human-like behaviors.

Section\textsuperscript{11} addresses related research, Section\textsuperscript{11} describes

\begin{verbatim}
https://vislab.isr.tecnico.ulisboa.pt/datasets_and_resources/BoxED
\end{verbatim}
the dataset, Section IV introduces our packing sequence prediction model, and Section V presents final remarks.

II. RELATED WORK

A. Learning from Demonstrations

Learning from Demonstrations (LfD) consists of learning how to execute new tasks and their constraints by observing an expert performing them [10]. This paradigm has attracted increasing attention from the robotics community because it circumvents predefined behaviors and introduces more flexibility in robotic learning. These methods require tracking the human’s actions which is not a trivial task due to rapid movements, occlusions and observation noise [11], [12].

One common approach for extracting knowledge from demonstrations is policy learning, where the goal is to learn a policy that maps the input space into the output space (usually the state and action spaces, respectively) and that can accurately mimic the demonstrations. Some approaches bypass explicit state characterization by defining as input raw data from the system with deep learning methods [13], [14]. Another approach that is widely used is based on Markov Decision Processes. For instance, Munzer T. et al. [15] used a variation of Markov Decision Processes along with first order logic in a system for a collaborative robot that learns tasks and human preferences before and during execution.

One difficulty in LfD is how to encode the demonstration in a useful format for training models. In some works, the expert wears a motion capture suit [16] or the demonstrations occur in a simulated environment [17], while others use complex pose estimators [18].

B. Bin Packing

This field addresses the task of packing a set of objects into a bin with maximum space efficiency. In general, there are two categories of methods: those based on search strategies and those that use unsupervised learning methods such as reinforcement learning (RL). The first cluster has the advantage of fast deployment as there is no learning phase, however most have slow execution times (ranging from dozens of seconds to a few hours [6], [7]). Conversely, methods based on RL or Deep RL (DRL) [19], [20] require complex training schemes but have faster execution times.

Our approach aims to overcome a big limitation in both categories of methods, as they often consider only cuboid objects (otherwise the search and action spaces grow drastically) and ignore properties as fragility.

C. Sequence Prediction

There are multiple types of sequence prediction, for instance sequence-to-sequence and sequence classification, however, for our task the most relevant is next value prediction. This type of sequence prediction focuses on predicting the next element of a sequence and encompasses problems such as time series forecasting and product recommendation.

Algorithms that address this class of problems range widely in their approach, from explicit association rules [21] and pattern mining algorithms [22], to deep learning based approaches which learn implicit representations. These algorithms search for sub-sequences that appear often in the data, and thus contain important sequential or associative information to predict the next element.

D. Related Datasets

Perhaps most similar to our work is the dataset proposed by Song S. et al. [23], consisting of videos of eight participants performing pick-and-place tasks in cluttered environments. The videos are annotated with the gripper trajectory (before the grasp and during the object manipulation), grasp pose, picking order and object mask. Although more encompassing than most robotic grasping datasets, this dataset is not related to bin packing and as such does not include packing order, placement inside a container nor object pose since pose estimation in a real environment is not trivial.

III. THE BOXED DATASET

A. The Virtual Environment

We chose to collect the BoxED dataset in VR as it simplifies much of the complexity associated with tracking object and gripper pose seen on [23]. As such, a virtual environment was created in Unity [24], illustrated in Figure 2 that consists of a circular area where the user is free to move and a table in its center. The task takes place on the table: the user should pack the available objects into a box [2]. We use 24 different objects, most chosen from the YCB dataset [25] and some obtained from public object model platforms.

![Image](https://example.com/image1.png)

Fig. 2. Virtual environment and gripper created in Unity for data collection.

The participants interact with the virtual world via the physical controller. The pose of the controller is mimicked by the virtual gripper and a pressure-sensing button controls the closing and opening of the virtual gripper’s fingers. The objects are configured as rigid bodies, which means that they are affected by gravity, friction and can collide with other objects. Furthermore, we implemented a haptic feedback on the physical controller to signal collisions between the gripper and objects to the participants. Upon grasping an object, it is rigidly attached to the gripper so that their relative pose is maintained throughout the manipulation.

A video of the first author completing the task can be found here [https://youtu.be/TUD-eCD5I8](https://youtu.be/TUD-eCD5I8)
B. Data Collecting Experiments

Before each box packing task, i.e., a scene, the participants were asked to behave as realistically as possible. Particular emphasis was given on the objects’ fragilities since this property is not present in the virtual environment but must be considered when stacking objects on top of each other. They were also instructed to pack the objects in an orderly manner starting from one side of the container, for reasons that are clarified in Section IV-A.

In each scene the objects selected correspond to a random subset of the entire object collection with replacement, generated such that the total sum of each object’s bounding box volume is between 70% to 90% of the container’s volume. This is to ensure that the task is not trivial, forcing the human expert to carefully reflect on how to place the objects. At the beginning of the scene the objects are spread out on the table with no overlaps to ensure that the participant can see and grasp all objects.

The experiment consists of 4 scenes (i.e., the expert packs 4 different boxes). For each object manipulation we record the 6-DOF pick-up grasp pose, the 6-DOF placement pose inside the box and the object’s trajectory from the former to the later sampled at 20 Hz and shown in Fig. 3. Additionally we also save the sequence in which the objects were packed, the pose of the VR headset throughout the experiment, and a top-down view of the initial layout of the objects.

C. Unpacking the Collected Data

The experiments were conducted using a HTC Vive Pro headset, totaling 75 experiments and 43 different participants (some participants volunteered more than once). In total the dataset contains 263 scenes, or in other words, 263 demonstrations of boxes being packed. This translates to 4644 grasp poses and object placements, which on average corresponds to 194 grasp poses and placements per object.

The histogram of each scene’s duration, presented in Fig. 4(a), shows that it approximates a gamma distribution with an average duration of 1 minute and 17 seconds, and a standard deviation of 36 seconds. Furthermore, a plot of each scene’s duration as a function of the corresponding number of objects in it is presented in Fig. 4(b). The pink line in this figure was obtained with a standard least-squares linear regression and indicates that, on average, each object adds 4 seconds to the task’s duration. This result meets our expectations, since the difficulty of the task naturally increases with the number of objects that need to be packed.

Analyzing the placement poses is not trivial since they are six-dimensional variables. Thus, for better visualization, we analyze the position and orientation separately, and only the two objects shown in Fig. 5(a).

To detect frequent patterns in the positioning of different objects inside the box, Fig. 5 shows a plot of the placement positions for two different objects, in a top-down view. In other words, this plots ignores the height of the placement and considers only the horizontal coordinates inside the box.

The strawberry is placed uniformly throughout the box, however the average placement is closer to the side of the box closest to the participant. This occurs because the strawberry is one of the most fragile items and so participants tended to pack it only after all other objects, confirming that participants considered the objects’ fragilities during the task. In contrast, the bleach cleaner has a very uneven distribution. It is clear that it is generally placed along the edges of the box and further away from the participant. This is likely
a strategy that most participants intuitively adopted due to two factors. Firstly, to optimize the available space inside the box, and secondly because large objects would become an obstacle for the task if they were placed closer to the participant since they would obstruct subsequent placements.

A similar strategy can be used to analyze the orientation, except in this case, we visualize independently each of the three angles that compose it - rotation around each axis. This analysis is presented in Fig. 6.

Fig. 6. Placement orientations inside the box of two objects, where each line segment corresponds to one placement. The objects are (a) strawberry and (b) bleach cleanser.

The visualization presented in Fig. 6 reveals, once again, distinctive patterns for each object. For instance, both have near uniform distribution around the Z axis since they can be safely placed in any orientation around it. On the other hand, the bleach cleanser presents a very skewed distribution around the Y axis (which in Unity points up). This is a consequence of its symmetric shape which leads participants to frequently place it aligned with the edges of the box.

The above two examples reemphasize the vast implicit knowledge humans use to pack a box with groceries and motivates our approach of learning these patterns from the data rather than relying on hand-designed rules or constraints.

IV. PACKING SEQUENCE PREDICTION

In this section we present both the approach we used to learn the packing sequence model from human demonstrations, as well as how to use the model to predict packing sequences for a certain set of objects.

A. Learning the Packing Sequence Prediction Model

Many methods exploit Markov chains as the core of their strategies to learn from demonstrations. The popularity of this model stems from its simplicity and from not requiring large amounts of data to train. This latter fact is important since our dataset contains less than 300 object packing sequences, which is not ideal for deep learning. Generally in these approaches, the state encodes relevant information about the target environment and agent at each instant, and the agent’s actions cause state transitions.

To deploy such a model to predict the packing sequence we consider the following simplifying assumptions:

1) The choice of what object to pack next depends only on the last object.

2) The aforementioned choice does not depend on the pose of the last object.

Furthermore, we can ignore the restrictions imposed by what objects are still available to pack for now (we show later how we address this when sampling the Markov chain). Consequently, the state is defined as the object’s name and transitioning to a new state corresponds to packing another object. To the reader this may seem as an over-simplification of the problem. In fact, when deciding what object to pack next, a human will likely take into account the last few objects that where packed (not just the last one), their disposition inside the box, and the objects that remain unpacked. Yet we show in this section that surprisingly accurate results can be obtained with our simplified formulation.

To formalize the model, let \( O = \{ obj_A, obj_B, \ldots \} \) be the set of all objects. We define \( S = \{ <\text{start}>, O \} \) as the set of all possible states, where at each instant the state indicates what object was packed last, and the state \( <\text{start}> \) indicates that no object was packed inside the box yet. Each transition probability, from an object \( obj_A \) to \( obj_B \), is represented by the conditional \( P(obj_B | obj_A) \) and indicates the likelihood of placing \( obj_B \) after placing \( obj_A \).

To materialize an actual model, we need to estimate the transition probabilities. Recall from the previous section that participants were asked to pack the box orderly, such that if an object "A" is placed after object "B", then "A" is likely stacked on top of "B", or at least is next to it. This request enables the extraction of meaningful patterns from the sequences without having to analyze each object’s placement pose.

However this is not true for all pairs of objects in the sequences, and so we need to extract only the meaningful patterns in the dataset. As we assumed that choosing the next object depends only on the last, we are only interested in extracting patterns with length of 2. In other words, we will only extract ordered pairs with the format \( obj_A \rightarrow obj_B \) from the sequences. Inspired by pattern mining algorithms [22], we extracted the pairs of objects that appeared in the sequences with frequency above an adjustable threshold (0.032). In other words, the pairs of objects whose support (probability of appearing in a sequence in the dataset) is smaller than the threshold are discarded.

The adjustable threshold controls how sensitive the model is to the transitions in the data. A higher threshold retains only the most frequent transitions whereas a lower threshold retains more transitions and produces a Markov chain that is more densely connected, at the cost of including transitions that may be irrelevant. This threshold was adjusted such that it was small enough to include at least one transition for each state in \( S \) and large enough to minimize loops in the chain, in order to maintain a hierarchical structure.
Once the transition probabilities were normalized such that the sum of the transition probabilities from each state is 1, the resulting Markov chain is presented in Fig. 7.

![Markov chain diagram]

Fig. 7. A Markov chain modeling the object packing sequence. Each node corresponds to a state and the numbers on the transition arrows are the corresponding probabilities.

We see that larger and more robust objects are placed first, followed by smaller and more fragile objects. For instance, the topmost object is the cracker box, which is the largest object in the dataset and was usually placed first inside the box by the participants. On the other hand, the bread and the toothbrush are the bottom-most objects since they are the most fragile and smallest, respectively. Furthermore, all transitions correspond to objects that could be stacked, one on top of the other, safely and stably, or correspond to pairs of objects that were frequently placed together, side-by-side.

B. Predicting a Packing Sequence with a Modified Beam Search

Regardless of how well the Markov model captures the humans’ packing strategies, it is of no use without an adequate sampling mechanism. When sampling from large stochastic models, beam search [26] generally provides better sampling than naive approaches such as greedy search. However, we introduce modifications to adjust it to models such as ours.

Firstly, recall that in each scene the set of objects that need to be packed is a subset of all objects, and consequently some objects in the Markov chain may be unavailable. In this case, standard beam search cannot be applied since the search must be restricted to only the objects that remain unpacked. This not only restricts what child nodes can be considered during search, but if, for a set of objects that were not yet packed, a node has no possible child nodes, it may still have a valid transition to a grandchild node through an unavailable child. By grandchild node we denote all nodes reachable from the current node with two or more transitions. This can make the search process very slow because for each node the algorithm would need to search all of its child and grandchild nodes to find valid transitions. The key observation to our solution is that the Markov chain is static - it never changes. Because of this we can build and store a table of all possible transitions for each object before executing the algorithm.

For each state in \( S \) (the set of all states), we store all possible transitions in levels. The first level contains the transitions to direct child nodes – those that require only one transition. The second level contains the transitions to the grandchild nodes that require two transitions, and so forth. This is essentially a more detailed (and redundant) description of our Markov chain. The original chain contains only the child nodes for each state – the minimum amount of information needed to store the chain. This new representation contains all transitions for all states. This increases the algorithm’s memory complexity but makes it faster, as the search has instant access to all valid transitions.

Secondly, during the search process we consider that a sequence is complete when it reaches a leaf node – a node that has no more valid transitions. But longer sequences will generally have smaller probabilities since we are multiplying probabilities (always smaller than 1). If, at the end of an iteration, the algorithm needs to select between two sequences - one is shorter and complete, the other is longer and still incomplete - which one should be discarded? If the search considers only the likelihood of a sequence then it will inevitably favor shorter sequences that have higher probabilities. This effect, in conjunction with the first change we made, would encourage the algorithm to find sequences that reach a leaf node with the smallest length possible.

Thus, we must introduce a preference for longer sequences in detriment of complete sequences. In practice this can be accomplished with two additions:

1) During the search process, choose sequences that can still expand to more nodes in detriment of sequences that have already reached a leaf node.
2) Enforce that transitions to a new node are as small as possible. In other words, start by searching for a valid state in the first level of our new representation. If none are found, proceed to the second level, and so forth.

The pseudo-code for the final algorithm is shown in Algorithm 1.

C. Testing the Model

Now that an appropriate sampling mechanism has been implemented, this section focuses on describing how to test the sampled sequences. The main challenge lies in the fact that, for a given set of objects, there are numerous valid packing sequences. Depending on how the objects are placed...
inside the box, two distinct sequences can lead to an equally efficient packing. This dependency on the object placement makes evaluating a sequence prediction very difficult without actually placing the objects inside the box. Additionally, current methods to predict a placement pose for an object in a packing task are limited.

As such, drawing some inspiration from the Turing test in which a human evaluator must distinguish between human or computer generated text responses, we develop a new virtual environment to test the sequence prediction model. This new environment is visually identical to the environment used to collect BoxED but the participant no longer chooses the order in which the objects are packed. Instead, the sequence is indicated by highlighting what object should be placed next. Some of the sequences are a replay of the real sequences from BoxED, whereas others are generated by the computer. At the end of each scene, the experimenter asks the participant if the sequence was a real sequence executed by a human or a computer generated sequence. The participants answer “human generated” if they felt that the sequence was logical and human-like, or “computer generated” otherwise.

There are three types of computer generated sequences:

1) Random: given the subset of objects on the table, a packing sequence is randomly generated.
2) Beam-N: sample a sequence prediction using the modified beam search algorithm. In this case, no limit is imposed on the length of a sequence prediction.
3) Beam-3: the same as Beam-N except a maximum sequence length of 3 is imposed. This method was introduced because we noticed that Beam-N produces long sequences that often pack fragile objects too soon in the packing process.

Each of the four types of sequence has its purpose. The real sequences verify that the participants can detect sequences executed by other humans. This tests whether the personal packing preferences are sufficiently strong to influence the final evaluation, or if there are general packing strategies that most humans adhere to. The random sequences test if participants distinguish between logical and random sequences. The results from evaluating this type of sequences and from the real sequences will measure if humans are good evaluators for this task and validate the experimental design. Finally, the results from the Beam-N and Beam-3 sequences will indicate if our model was able to “trick” participants into believing that the sequence was generated by another human. Note that in these two cases, a new sequence is sampled from the model whenever the participant packs all the objects in the previous sequence or when the participant places an object at the top of a stack of objects, near the top of the box. The intuition is that the next object will be placed next to this stack (and not on top of it), so a new sequence should be sampled from the top of the Markov chain.

D. Experimental Results

In total this second experiment involved 29 participants for a total of 96 evaluations, shown in Table I. Since Beam-N was replaced by Beam-3 during the experiment, both have fewer samples than the other two sequence types.

| Sequence Type | Human generated | Computer generated | Number of samples |
|---------------|-----------------|--------------------|-------------------|
| Random        | 7%              | 93%                | 30                |
| Real          | 79%             | 21%                | 33                |
| Beam-N        | 50%             | 50%                | 16                |
| Beam-3        | 88%             | 12%                | 17                |
Regarding the real sequences, the result is positive with 79% of the evaluations correctly identifying these as human generated, and thus we can infer that participants are generally capable of identifying a sequence produced by another human. However, the percentage of incorrect answers is larger than expected. We suggest that there are two main factors contributing to this error. Firstly, the individual differences of how to best pack objects in a box which may lead individuals to dislike or deem as illogical another person’s packing sequence. The second factor is more arbitrary and is related to the fact that the participant may place a few of the initial objects in positions that become an obstacle or that are incompatible with the rest of the sequence. Nevertheless, the result is positive as the majority of the participants correctly identified the sequence type.

Likewise, the vast majority correctly identified that the randomly generated sequences were illogical, proving that there are significant differences between real and random sequences. Along with the previous result, this validates the experimental design - humans are good evaluators of this task and can recognize packing strategies in sequences produced by other humans. This implies that there are, in fact, general rules or strategies that most humans apply to this task.

Unexpectedly, 50% of participants correctly detected that the sequences generated by our model with Beam-N sampling were computer generated. Such a high percentage clearly indicates that sequences sampled with Beam-N are not similar to the real sequences. This result led to the development and introduction of the variation Beam-3.

This modification clearly had a positive influence on the performance of our model. The success rate at “tricking” participants increased from 50% with Beam-N to a surprising 88% with Beam-3, even higher than the percentage of real sequences that were classified as human generated. One possible explanation for this result is that our model learned more general principles for the task that the majority of humans adhere to, instead of learning specific principles that only a few humans consider, thus pleasing more participants.

To validate these intuitive conclusions we conduct Boschloo’s exact test [27] (instead of the standard two proportions Z-test due to limited sample size) to test whether the proportion of individuals that was “tricked” by the Beam-3 algorithm, \( p_1 \), is greater than the proportion of individuals “tricked” by Beam-N, \( p_2 \), at a significance level of \( \alpha = 0.05 \). As such, our null and one-tailed alternative hypothesis are

\[
\begin{align*}
H_0 : p_1 &= p_2 \\
H_1 : p_1 > p_2
\end{align*}
\]

From this test we obtain a p-value of 0.0099, and hence, at this significance level, we reject the null hypothesis and prove the difference between these two proportions. In other words, Beam-3 is an improvement over Beam-N. However, this only guarantees that there is an effect, and not that it is significant. The effect size measures importance or significance on a normalized scale, regardless of the quantities in the study. There are multiple measures of this effect size, but we consider Cohen’s \( \text{H} \) [28], as is standard in statistical analysis. Our results produce an \( \text{H} \) value of approximately 0.86, which indicates a large effect according to Cohen’s interpretation [28]. Finally, these results are validated with a post-hoc power analysis, reporting a power of 79%, which is a satisfactory power value according to [28].

V. CONCLUSIONS

We proposed a new method for learning the packing sequence of irregular objects from human demonstrations. The solutions generated by our model have been perceived as human-like by subjects. By learning implicit task constraints directly from expert demonstrations we showed that harvesting human knowledge can not only circumvent labor and data intensive approaches but also generate human-like behaviors.

Besides tackling this largely unsolved problem, we created BoxED, a new dataset of demonstrations of human experts performing the box packing task. Using a portion of our new dataset our model surpassed human performance at predicting object packing sequences that are classified by experts as human-like.

Future work may include the first supervised learning approach to predicting a placement pose for objects in the bin packing task, and the introduction of a level of abstraction from the objects to eliminate explicit knowledge of each object’s physical properties.

REFERENCES

[1] Ocado Group. [Online]. Available: https://www.ocadogroup.com/ (accessed Aug. 8, 2022).
[2] J. E. Colgate and M. A. Peshkin, “Robots,” U.S. Patent 5952796, Sep. 14, 1999.
[3] Amazon Inc., “Introducing Amazon Astro, household robot for home monitoring.” [Online]. Available: https://www.amazon.com/Introducing-Amazon-Astro/dp/B078NSDFSB (accessed Jan. 10, 2022).
[4] temi USA inc., “temi - The personal robot.” [Online]. Available: https://www.robotemi.com/ (accessed Jan. 10, 2022).
[5] K. Kleeberger, R. Bornmann, W. Kraus, and M. F. Huber, “A survey on learning-based robotic grasping,” Current Robotics Reports, 2020.
[6] F. Wang and K. Hauser, “Robot packing with known items and nondeterministic arrival order,” IEEE Trans. Autom. Sci, vol. 18, no. 4, pp. 1901–1915, 2021.
[7] F. Wang and K. Hauser, “ Stable bin packing of non-convex 3d objects with a robot manipulator,” in Proc. 2019 IEEE Int. Conf. Robot. Autom. (ICRA), 2019, pp. 8698–8704.
[8] R. Hu, J. Xu, B. Chen, M. Gong, H. Zhang, and H. Huang, “TapNet: Transport-and-pack using reinforcement learning,” ACM Trans. Graph., vol. 39, no. 6, nov 2020.
[9] Y. Jiang, Z. Cao, and J. Zhang, “Solving 3d bin packing problem via multimodal deep reinforcement learning,” in Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2021, p. 1548–1550.
[10] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, “Recent advances in robot learning from demonstration,” Annual Review of Control, Robotics, and Autonomous Systems, vol. 3, no. 1, pp. 297–330, 2020.
[11] F. Codella, M. Müller, A. López, V. Kolton, and A. Dosovitskiy, “End-to-end driving via conditional imitation learning,” in Proc. 2018 IEEE Int. Conf. Robot. Autom. (ICRA), pp. 4693–4700.
[12] H. Oh, H. Sasaki, B. Michael, and T. Matsubara, “Bayesian disturbance injection: Robust imitation learning of flexible policies,” in Proc. 2021 IEEE Int. Conf. Robot. Autom. (ICRA), pp. 8629–8635.
[13] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, “Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration,” in Proc. 2018 IEEE Int. Conf. Robot. Autom. (ICRA), pp. 3758–3765.

[14] F. Esposito, C. Pek, M. C. Welle, and D. Kragic, “Learning task constraints in visual-action planning from demonstrations,” in Proc. 2021 IEEE Int. Conf. Hum. Interact. Commun. (RO-MAN), pp. 131–138.

[15] T. Munzer, M. Toussaint, and M. Lopes, “Efficient behavior learning in human—robot collaboration,” Auton. Robots, vol. 42, no. 5, p. 1103–1115, June 2018.

[16] A. Ghadirzadeh, X. Chen, W. Yin, Z. Yi, M. Björkman, and D. Kragic, “Human-centered collaborative robots with deep reinforcement learning,” IEEE Robot. Autom. Lett. (RA-L), vol. 6, no. 2, pp. 566–571, Apr. 2021.

[17] S. Dasari and A. K. Gupta, “Transformers for one-shot visual imitation,” in Proc. 2020 Conf. Robot Learn., Nov.

[18] P. Wang, F. Manhardt, L. Minciullo, L. Garattoni, S. Meier, N. Navab, and B. Busam, “Demograsp: Few-shot learning for robotic grasping with human demonstration,” in Proc. 2021 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Sep., pp. 5733–5740.

[19] H. Zhao, Y. Yu, and K. Xu, “Learning efficient online 3d bin packing on packing configuration trees,” in Proc. 2022 Int. Conf. Learn. Representations.

[20] R. Verma, A. Singhal, H. Khadilkar, A. Basumutary, S. Nayak, H. V. Singh, S. Kumar, and R. Sinha, “A generalized reinforcement learning algorithm for online 3d bin-packing,” CoRR, vol. abs/2007.00463, 2020.

[21] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules in large databases,” in Proc. 20th Int. Conf. Very Large Data Bases, ser. VLDB ’94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, p. 487–499.

[22] M. J. Zaki, “Spade: An efficient algorithm for mining frequent sequences,” in Machine Learning, 2001, pp. 31–60.

[23] S. Song, A. Zeng, J. Lee, and T. Funkhouser, “Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations,” IEEE Robot. Autom. Lett., vol. 5, pp. 4978–4985, 2020.

[24] Unity Technologies. [Online]. Available: https://unity.com/ (accessed May, 12 2022).

[25] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, “Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set,” IEEE Robot. Automat. Mag., vol. 22, no. 3, pp. 36–52, 2015.

[26] Carnegie-Mellon University.Computer Science Dept., “Speech understanding systems: summary of results of the five-year research effort at Carnegie-Mellon University.” Computer Science Department, 1977.

[27] R. D. Boschloo, “Raised conditional level of significance for the 2 × 2-table when testing the equality of two probabilities,” Statistica Neerlandica, vol. 24, no. 1, pp. 1–9, 1970.

[28] J. Cohen, Statistical power analysis for the behavioral sciences. Routledge, 1988.