Task Planning for Visual Room Rearrangement under Partial Observability




Karan Mirakhor* Sourav Ghosh* Dipanjan Das Brojeshwar Bhowmick

{karan.mirakhor, g.sourav10, dipanjan.da, b.bhowmick}@tcs.com

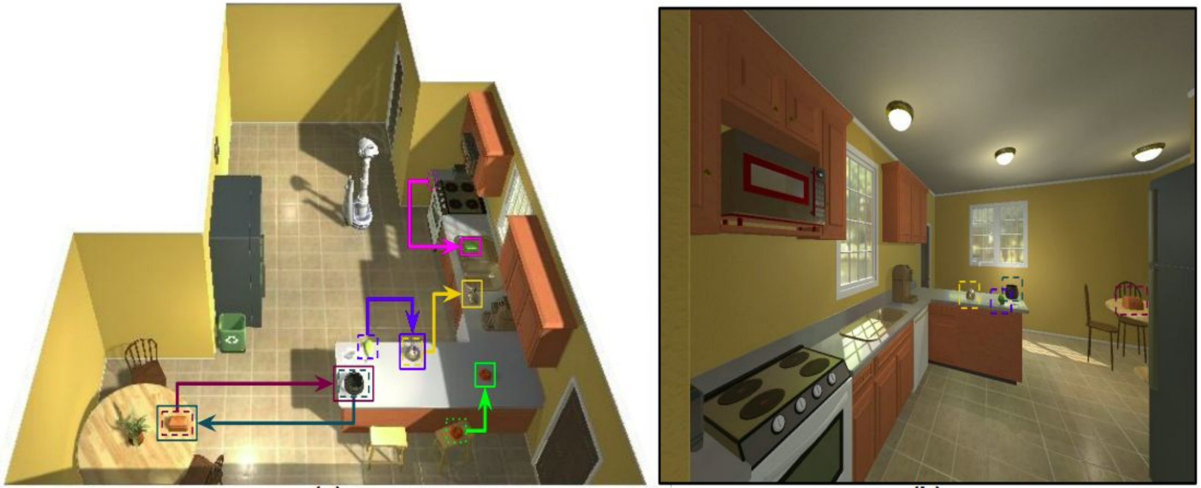
TCS Research, Kolkata, India

Paper ID : 1796

ICLR 2024

*These authors contributed equally.

Problem : *To solve the user-specified room-level object rearrangement in an efficient and time-bound manner under partial observability.*



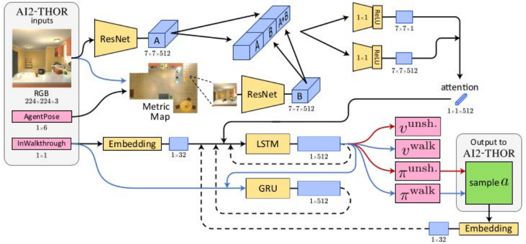
(a) (b)

a. **Rearrangement Task :**
   i. The user-specified goal state for all objects is indicated by the solid 2D bounding boxes.
   ii. To capture this goal state, the agent explores the scene using a state-of-the art exploration strategy and perception [1]
   iii. The dashed boxes show the initial positions of visible objects in the untidy current state. The dotted 2D bounding boxes represent initial positions of unseen objects in the untidy current state..

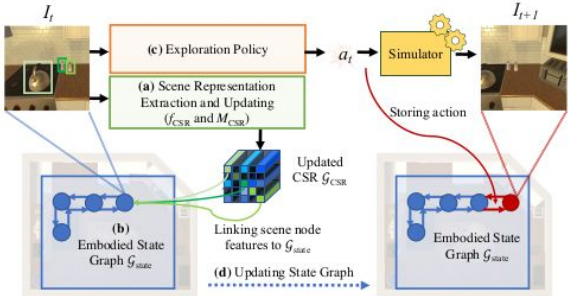
b. **Current State :** The agent's knowledge of the room scenario is limited to its egocentric perception (partial observability).
   i. Here only the kettle (yellow), lettuce (blue) and pot (dark cyan) are visible.

1. Sarch, G.; Fang, Z.; Harley, A. W.; Schydlo, P.; Tarr, M. J.; Gupta, S.; and Fragkiadaki, K. 2022. TIDEE: Tidying Up Novel Rooms using Visuo-Semantic Commonsense Priors. In European Conference on Computer Vision

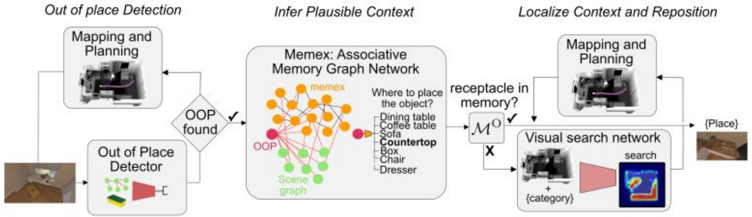
Related works on object rearrangement in a room



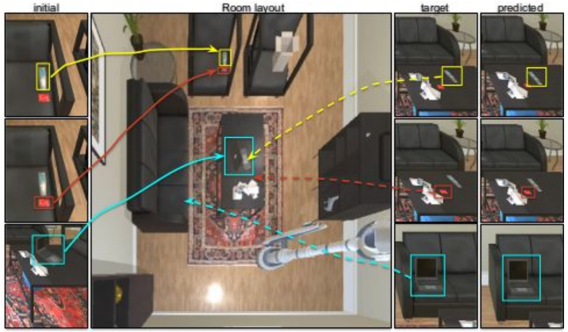
Visual Room Rearrangement
Weihs *et al.*, CVPR 2021



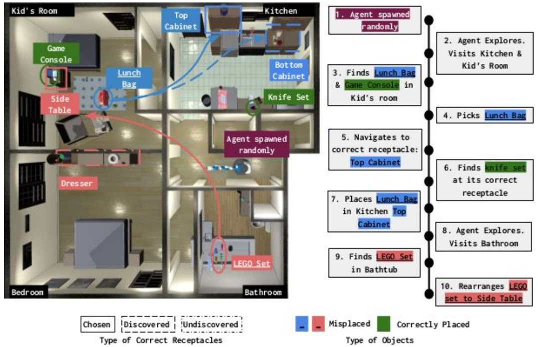
Continuous Scene Representation
Gadre *et al.*, CVPR 2022



TIDEE: Tidying Up Novel Rooms using
Visuo-Semantic Commonsense Priors
Sarch *et al.*, ECCV 2022



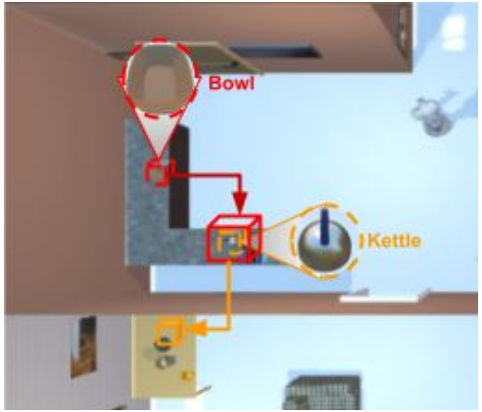
Planning Large-scale Object Rearrangement Using
Deep Reinforcement Learning
Ghosh *et al.*, IJCNN 2022



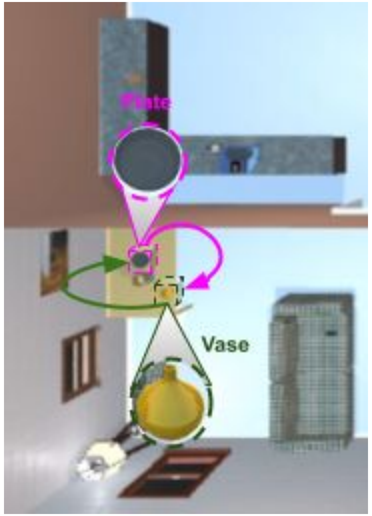
Housekeep: Tidying Virtual Households using
Commonsense Reasoning
Kant *et al.*, ECCV 2022

What is missing in the state-of-the art methods for object rearrangement?

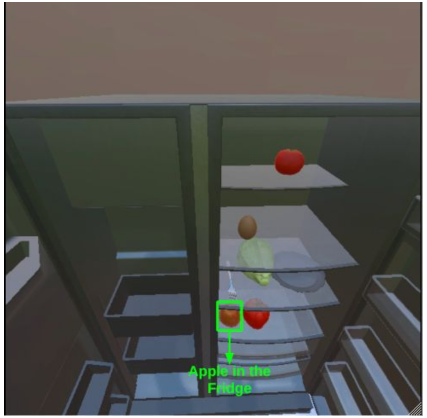
1. **Exploration in the current state :** All the existing methods perform exploration in the current state to overcome partial observability i.e. discover objects beyond the visual scope of the agent's egocentric view. This incurs huge traversal cost.
2. **Inefficient Planning :** The existing methods use greedy or other heuristic planners to plan the sequence of object rearrangement.
3. **Non-addressal of Blocked Goal/Swap Cases :** No existing methods address blocked goals or swap cases effectively.
4. **No prior work attempts to find objects placed inside closed receptacles.**



Blocked Goal Case : The goal position of one object is occupied by the current position of another object.

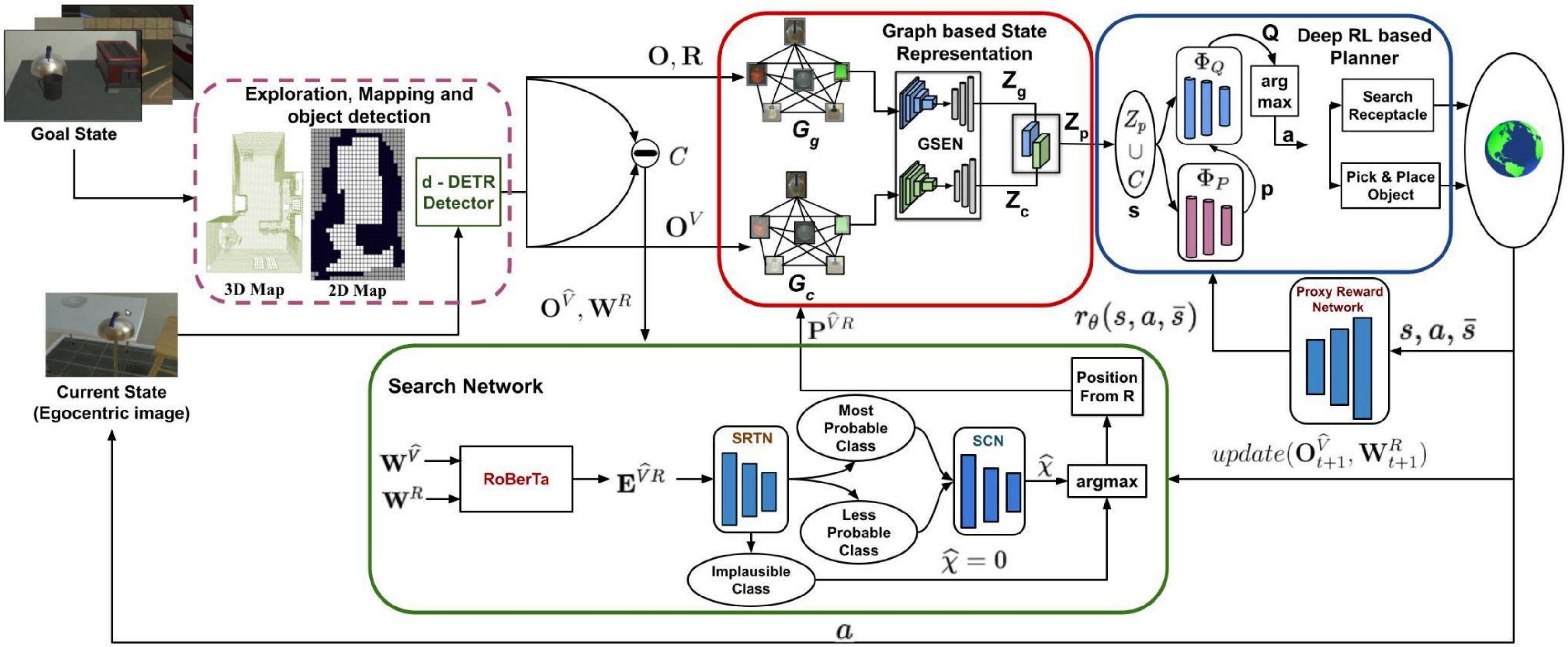


Swap Case : The goal position of one object is occupied by the current position of another object and vice versa..



Objects inside closed receptacles : An instance in which the apple is placed inside the refrigerator.

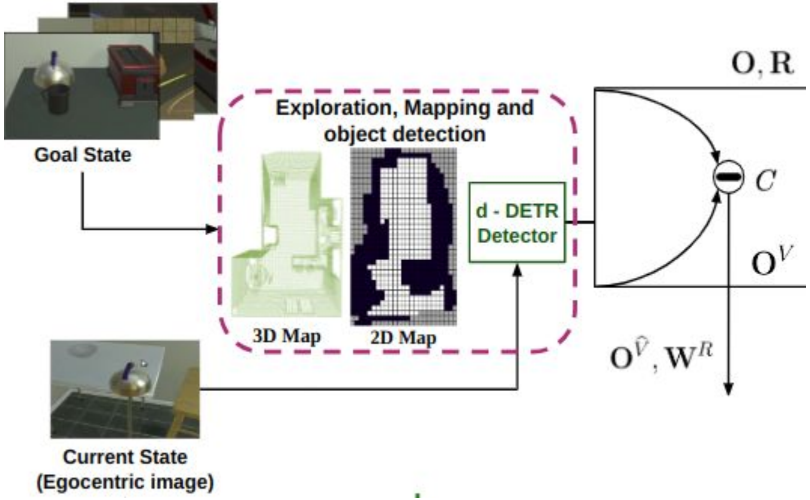
Task Planning for Visual Room Rearrangement



Contributions :

- **First end-to-end method** to address the task planning problem for single-room rearrangement from egocentric view under partial observability.
- **Novel Search Network** that leverages object-receptacle semantics using LLM to predict the most probable room-receptacle for unseen object.
- **New scalable and scene invariant Graph based state representation** containing the geometric information about the objects in the current and goal state.
- Use of **Deep RL planner trained with proxy reward** to overcome combinatorial expansion in rearrangement sequencing and, to optimize the overall traversal.
- **Introduction of a novel, sample-efficient cluster-biased sampling** for simultaneous training of the proxy reward network and the Deep RL network.
- Set of **new Evaluation criteria** to gauge the efficacy of our method in terms of the number of steps taken and the overall agent traversal.
- To address the inadequacies in existing benchmarks for evaluating single-room task planning under partial observability, we introduce the **RoPOR Dataset**.

Our Method - Exploration, Mapping and Object detection

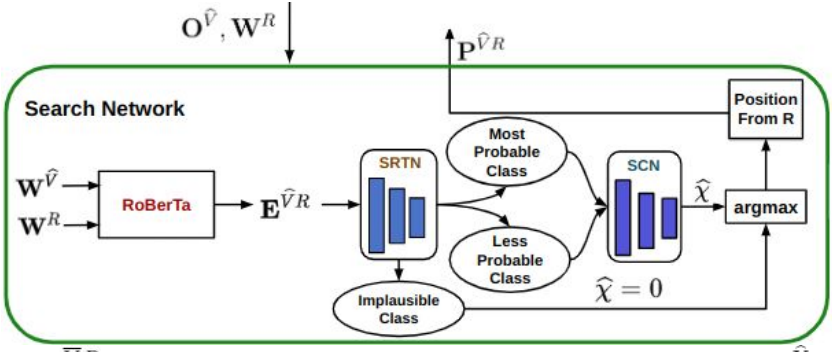


1. **Goal State :** Using the RGB-D images from the egocentric camera view along with the egomotion information, the agent generates a 3D map and a 2D occupancy map of the scene using state-of-the art exploration strategy[1].
   a. Using d-detr[2] object detector on each RGB image to obtain 2D bounding boxes and semantic labels **(L)** for objects and receptacles.
   b. The corresponding 3D centroids **(P)** are obtained using depth images, camera intrinsics and extrinsics.
   c. Using object segmentation[3] on the registered point cloud, the agent records the corners of the 3D bounding boxes **(B)** of each object in the goal state.
   d. Finally, we obtain the object list $\mathbf{O} = \{[\mathbf{L}_i, \mathbf{P}_i, \mathbf{B}_i], i = 1, 2, .., N\}$ and receptacle list $\mathbf{R} = \{[\mathbf{L}_i^R, \mathbf{P}_i^R], i = 1, 2, .., N_R\}$ in the goal state :
2. **Current State :** In this state, the agent's knowledge of the scene is limited to its egocentric view (partial observability) and thus only a set of objects are visible.
   a. Comparing the set of visible object list $\mathbf{O}^V = \{[\mathbf{L}_i^V, \mathbf{P}_i^{\hat{V}}], i = 1, 2, .., N_V\}$ in the agent's egocentric view of the current state with respect to the object list **(O)** recorded in the goal state, the agent is able to determine only the semantics of the unseen objects $\mathbf{O}^{\overline{V}} = \{[\mathbf{L}_i^{\overline{V}}], i = 1, 2, .., N_{\overline{V}}\}$
   b. In addition, comparing the positions of visible objects in the current and the goal state helps us to determine the collision cases **C**.

1. Sarch, G.; Fang, Z.; Harley, A. W.; Schydlo, P.; Tarr, M. J.; Gupta, S.; and Fragkiadaki, K. TIDEE: Tidying Up Novel Rooms using Visuo-Semantic Commonsense Priors. In European Conference on Computer Vision, 2022
2. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2021. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In 9th International Conference on Learning Representations, ICLR 2021
3. Rusu, R. B.; and Cousins, S. 2011. 3D is here: Point Cloud Library (PCL). In IEEE International Conference on Robotics and Automation (ICRA). Shanghai, China

Our Method - Search Network

- To plan efficiently, the agent needs to know the location of all objects under partial observability in the current state.
- To discover unseen objects in the untidy current state, we propose a novel Search Network that utilizes the commonsense knowledge in Large language models (LLMs) to predict the probable room-receptacles.
- Since, the current state is untidy, we search for unseen objects by prioritizing their search on/in misplaced receptacles.



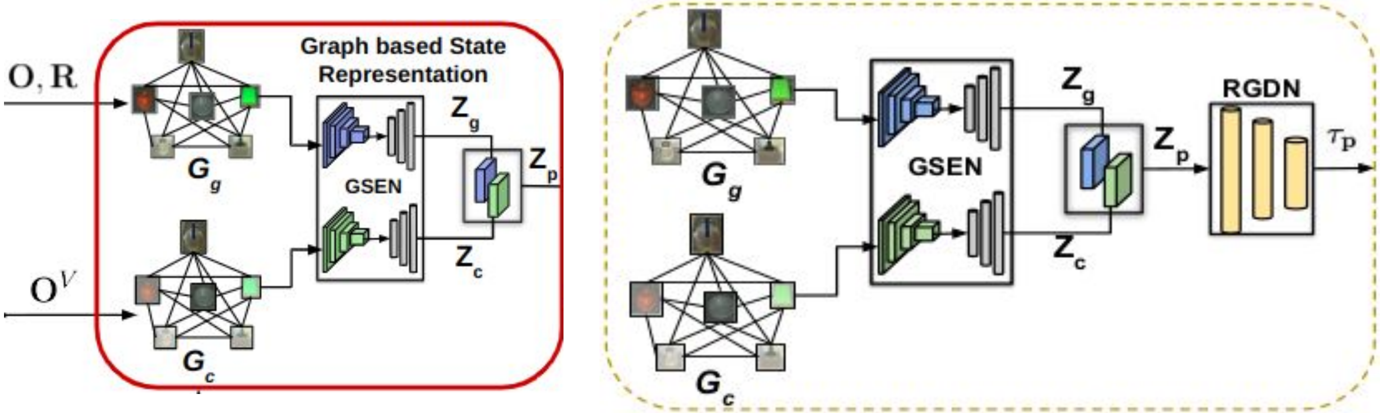
- We generate the RoBERTa embeddings $\mathbf{E}^{\hat{V}R}$ for pairwise concatenated semantic labels of unseen objects $\{\mathbf{W}^{\hat{\mathbf{V}}}\}_{i=1,2,\ldots,N_{\hat{\mathbf{V}}}}$ and room-receptacles $\{\mathbf{W}^{\mathbf{R}}\}_{i=1,2,\ldots,N_R}$
- We devise a two-step approach to enhance the accuracy and reduce the search space for finding the exact object-room-receptacle (ORR) embedding :
  - **Sorting Network (SRTN) :** MLP-based Sorting Network (SRTN) to classify the ORR embeddings.
    - We train this network using a Cross-Entropy Loss on the ground truth class labels for each ORR in the dataset.
    - Here, {i = 1 : Most Probable Class, 2 : Less Probable Class, 3 : Implausible Class} indicate the probability of finding a misplaced object at a given room-receptacle
  - **Scoring Network (SCN) :** We use a regression-based Scoring Network (SCN) to estimate the probability scores for embeddings of probable classes.
    - We train this network using Mean Square Error (MSE) Loss with respect to the ground truth probability scores from human annotation rankings.
    - The room-receptacles with the highest scores from SCN are selected as the probable room-receptacles for finding the unseen objects.
  - The predicted room-receptacles position $\{\mathbf{P}_i^{\hat{V}R}\}_{i=1,..N_{\hat{V}}}$ is taken from the goal state and used as the location for unseen objects in the current state.

Dynamically updating the search criteria and handling failures in unseen object search

- To dynamically update the search criteria, we maintain and update an object visibility list on-the go :
  - Update any object which is spotted along the agent’s path
  - Remove any room-receptacle that does not contain any unseen object.
- Search Network re-plans if the unseen object being searched currently is (a) not found at the predicte receptacle or (b) discovered along the path.

Our Method - Graph based State Representation

- There are two major challenges in representation for rearrangement planning :
  - Ensuring **Scalability** of the planner to a large number of objects.
  - **Scene Invariance** of the planner to different scenes and object configurations.
  - To address these challenges, we proposed and implemented a *Graph based State Representation for Deep RL planner*.

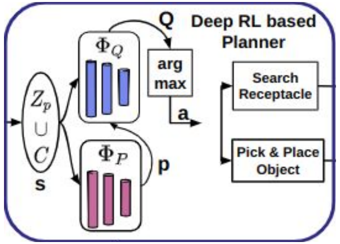


Graph based State Representation :

- The goal **(G_g)** and the current **(G_c)** state graph represent the geometric state of the room in the tidy goal state and the untidy current state..
- The nodes of these graphs contain the positions of the objects in the goal and the current state respectively.
- The nodes of this graph are fully connected through edges.
- The edge attributes include the length of the shortest collision free path $D(p_i, p_j)_{i \neq j}$ from the node position $\mathbf{P}_i$ to the connected node position $\mathbf{P}_j$
- $D(p_i, p_j)_{i \neq j}$ is computed using BFS algorithm between the 2D projections of $\mathbf{P}_i, \mathbf{P}_j$ on $\mathcal{M}^{2D}$
- For unseen objects in the current state, the source object nodes in the graph are augmented with $P^{\bar{V}R}$ from Unseen object discovery method
- We use a novel Graph Representation Network (GRN) with an encoder-decoder to generate meaningful embeddings from ***G_g*** and ***G_c*.**
- GRN consists of two major blocks, the Graph Siamese Encoder Network (GSEN) and the Residual Geodesic Distance Network (RGDN)
- GSEN uses a Graph Convolution Network to encode the graphs ***G_g*** and ***G_c*** and produce the graph embeddings ***Z_g*** and ***Z_c*** respectively.
- RGDN acts as a decoder and predicts the residual relative path length between the two graphs.
- These graph embeddings are concatenated to get the final embedding $Z_p = Z_c \cup Z_g$for RL state space.

Our Method - Deep RL based Planner

- There is are major challenges in rearrangement task planning :
  - The task planner has to plan a sequence of actions to :
    - Rearrange visible objects
    - Search for unseen objects at probable receptacles
    - Resolve blocked goal and swap cases.
  - **Combinatorial expansion** in the complexity of rearrangement sequencing with an increase in number of objects.
  - To address these challenges, we implemented a *Parameterized Deep-Q Network*.



- Our task planner sequences actions to simultaneously **(i)** rearrange visible objects, **(ii)** search for unseen objects and **(iii)** resolve blocked goal/swap cases.
- This reduces the agent's overall travel time as **(i)** it eliminates the requirement of explicit exploration to find unseen objects and **(ii)** the rearrangement of visible objects inevitably leads to the discovery of some unseen objects.
- We implement a ***Parameterized Deep-Q Network*** works with a hybrid action space $\{a = (k, p_k)\}$, where ***k*** denotes the index of the selected visible object or the probable search receptacle for unseen object and p_k signifies the location for object placement or receptacle search.
- We use a Parameter network Φ_P and the Q-network Φ_Q to generate a continuous parameter p_k and a discrete action ***k*** respectively.
- In addition, we define the Q-values as a function of the joint continuous action parameter $p = \{p_k\}_{k=1,2..,K}$ instead of updating the Q-values with its corresponding continuous parameter sample p_k.
- The modified Bellman equation is shown below :

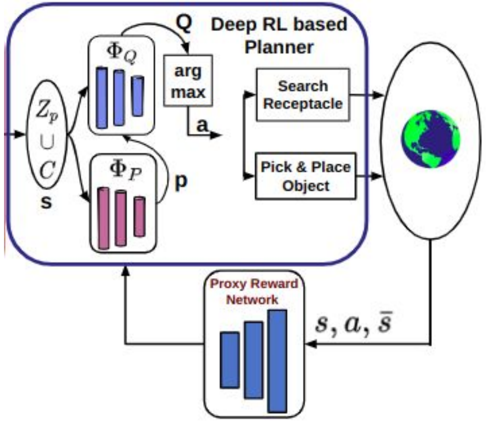
$$Q(s,k,p) = \underset{r,\bar{s}}{E}[r + \gamma \underset{k \in K}{max} Q(\bar{s}, k, \Phi_P(\bar{s}))|s,k,p]$$

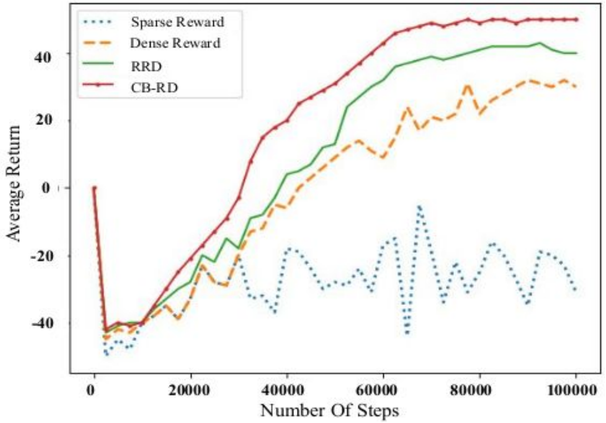
- To accomplish these objectives and obtain close to optimal solution in long-horizon problem, we employ an *Off policy* learning technique :.

$$L_P(\Phi_P) = -\sum^{R_B}\sum_{k=1}^{K} Q(s,k,\Phi_P(s);\Phi_Q) \quad L_Q(\Phi_Q) = \underset{(s,k,p,r,\bar{s})\leftarrow R_B}{E}[\frac{1}{2}(y - Q(s,k,p;\Phi_Q))^2] \quad y = r + \gamma \underset{k \in K}{max} Q(\bar{s}, \bar{k}, p(\bar{s};\Phi_P);\Phi_Q)$$

Our Method - Proxy Reward Network

- It is not sample efficient to train the Deep RL with sparse reward for Long Horizon Planning.
- Hence, we use stepwise environmental feedback based on the hierarchical dense reward.
- This reward structure provides per-step feedback, but we need episodic reward-based feedback to improve RL policy generalization.
- As this episodic reward is sparse, we use a proxy reward network to generate per-step dense Markovian reward with an episodic notion.





- Our proxy reward network is trained on the sampled experience data from the replay buffer to give our agent a notion of the overall objective of the episode.
- As seen in the graph above, random return decomposition (RRD) method is not sample efficient to train the proxy reward network.
- To this end, we propose a novel cluster-biased return reward decomposition (CB-RD) to train our proxy reward network.
- We cluster the per-step reward for the episode into 3 clusters each of size $\boldsymbol{T_j}$, where $j \in \{1, 2, 3\}$, using the c-means clustering.
- Then, we randomly sample $\boldsymbol{N_j}$ steps from each cluster $U_j = \{(s_{ij}, a_{ij}, \bar{s}_{ij})\}_{i=1}^{N_j}$, such that $\boldsymbol{N_j = N_s \times T_j / N_{ep}}$.
- Using $\{U_j\}_{j=1,2,3}$, we estimate the learned episodic reward from the proxy reward network.

$$R_{ep} = \frac{N_{ep}}{L} \sum_{i=1}^{N_{ep}} r_i \quad R_{ep,\theta} = \sum_{j=1}^{3} p_j \frac{T_j}{N_j} \sum_{i=1}^{N_j} r_\theta(s_{i,j}, a_{i,j}, \overline{s_{i,j}}) \quad L_{CBRD} = \frac{1}{M} \sum_{i=1}^{M} \left[(R_{ep_i} - R_{ep,\theta_i})^2 \right]$$

- As the graph shows, CB-RD provides effective feedback to our Deep RL method to achieve a higher average return in fewer training steps.
- We use an off-policy method with a replay buffer to simultaneously train our proxy reward network and Deep RL method.

Experiments:

1. RoPOR - Benchmark Dataset :

a. Motivation for new dataset : The existing benchmark dataset - RoomR[1] has the following limitations

- Only 5 objects in each rearrangement scenario.
- No object placement inside another receptacle in the current state.
- No blocked goal or swap cases configuration.

b. RoPOR :

- Contains single-room scenarios from iThor.
- Upto 20 objects in a single-room scenario.
- Ensures realistic scenarios using the semantic priors derived from the AMT dataset[2].
- Supports up to 108 object and receptacle categories, featuring 120 single-room scenes.
- In each single-room scene, there are 30 object configurations, with the number of objects randomly varying between 5 and 20.
- In each scene, there are configurations with object placement within receptacles in the current state, along with blocked goal and swap cases.
- The mean room dimensions along x-axis and y-axis are 3.12m and 5.80m, respectively.

1. Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In Proceedings of the IEEE/CVF Conference on Computer Vision andPattern Recognition, pages 5922–5931, 2021.
2. Yash Kant, Arun Ramachandran, Sriram Yenamandra, Igor Gilitschenski, Dhruv Batra, Andrew Szot, and Harsh Agrawal. Housekeep: Tidying virtual households using commonsense reasoning. In ECCV, 2022.

Experiments:

1. Evaluation Metrics :

a. Motivation for new metrics : The existing metrics in (Weihs et al. 2021)[1] do not highlight the efficacy of a task planner

- The metrics - Success Rate (**SR**), Fixed Strict (**FS**) and Energy Remaining (**ER)** measure how much of the rearrangement task is complete.
- Do not judge the efficiency of sequencing the order of rearrangement to reduce the number of steps taken or the agent traversal during rearrangement.

b. Our Metrics :

- **SNS : S**uccess measured by the inverse **N**umber of **S**teps evaluates the success and efficiency of rearrangement episode by combining the binary success rate and the number of steps taken by the agent to rearrange a given number of objects. A higher **SNS** indicates a more efficient and successful rearrangement episode.
- **ENR : E**fficiency in **N**umber of **R**e-plans is the ratio of the number of unseen objects in the scene initially with respect to the number of attempts to search. A higher **ENR** shows a lower number of steps required for a given number of unseen objects indicating a more efficient search to find unseen objects.
- **ATC : A**bsolute **T**raversal **C**ost metric shows the total distance traversed by the agent during the successful completion of a rearrangement episode. In an identical test configuration, a lower **ATC** indicates a more efficient rearrangement sequencing

1. Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5922–5931, 2021.

tcs Research
Inventing for impact

ICLR

Results:

1. Quantitative Result:

a. Ablation Study:

Our-GT : Our method with Ground truth perception, **Ours-RS :** Our method with random search, **Ours-GE :** Our method with greedy exploration, **Ours-DR** : Our method trained with a hierarchical Dense Reward structure

Number of Objects	Visible Objects	Unseen Objects		Swap Case	Ours-GT			Ours			Ours-RS			Ours-GE			Ours-DR		
		OOF	OPR		SNS↑	ENR↑	ATC(m)	SNS↑	ENR↑	ATC(m)	SNS↑	ENR↑	ATC(m)	SNS↑	ENR↑	ATC(m)	SNS↑	ENR↑	ATC(m)
5	5	0	0	0	**0.98**	NC	**10.57**	**0.74**	NC	**11.98**	**0.74**	NC	**11.98**	**0.74**	NC	**11.98**	**0.74**	NC	12.76
	5	0	0	2	**0.70**	NC	**12.36**	**0.53**	NC	**13.46**	**0.53**	NC	**13.46**	**0.53**	NC	**13.46**	**0.53**	NC	15.36
	3	2	0	0	**0.81**	**0.61**	**12.93**	**0.60**	**0.48**	**14.33**	0.46	0.30	22.23	0.36	0.20	26.38	**0.60**	**0.48**	16.18
	3	0	2	0	**0.79**	**0.60**	**13.39**	**0.58**	**0.47**	**14.89**	0.41	0.22	23.86	0.0	NC	NC	**0.58**	**0.47**	16.93
10	10	0	0	0	**0.97**	NC	**22.19**	**0.73**	NC	**24.51**	**0.73**	NC	**24.51**	**0.73**	NC	**24.51**	**0.73**	NC	26.37
	10	0	0	4	**0.70**	NC	**24.63**	**0.52**	NC	**27.32**	**0.52**	NC	**27.32**	**0.52**	NC	**27.32**	**0.52**	NC	29.46
	6	4	0	0	**0.84**	**0.69**	**23.78**	**0.64**	**0.53**	**25.56**	0.48	0.34	36.72	0.41	0.23	40.09	**0.64**	**0.53**	26.13
	6	0	4	0	**0.83**	**0.67**	**24.15**	**0.62**	**0.52**	**25.97**	0.43	0.25	38.47	0	NC	NC	**0.62**	**0.52**	26.59
20	20	0	0	0	**0.95**	NC	**40.05**	**0.73**	NC	**44.05**	**0.73**	NC	**44.05**	**0.73**	NC	**44.05**	**0.73**	NC	48.27
	20	0	0	8	**0.70**	NC	**45.32**	**0.52**	NC	**48.32**	**0.52**	NC	**48.32**	**0.52**	NC	**48.32**	**0.52**	NC	51.55
	12	8	0	0	**0.87**	**0.75**	**41.29**	**0.67**	**0.58**	**45.29**	0.51	0.36	52.45	0.46	0.28	56.68	**0.67**	**0.58**	47.42
	12	0	8	0	**0.87**	**0.74**	**42.13**	**0.66**	**0.57**	**45.78**	0.47	0.28	54.68	0	NC	NC	**0.66**	**0.57**	47.67

Results for Single-room Rearrangement with comparison against Baselines. Here, **OOF**. indicates objects outside agent's field of view initially and **OPR** stands for objects placed inside closed receptacles. Ours-GE does not handle OPR cases, therefore its ENR is non-computable (NC) due to division by zero. The table shows that Ours-GT and Ours outperform the other baselines in terms of the evaluation criteria,

Results:

1. Quantitative Result:

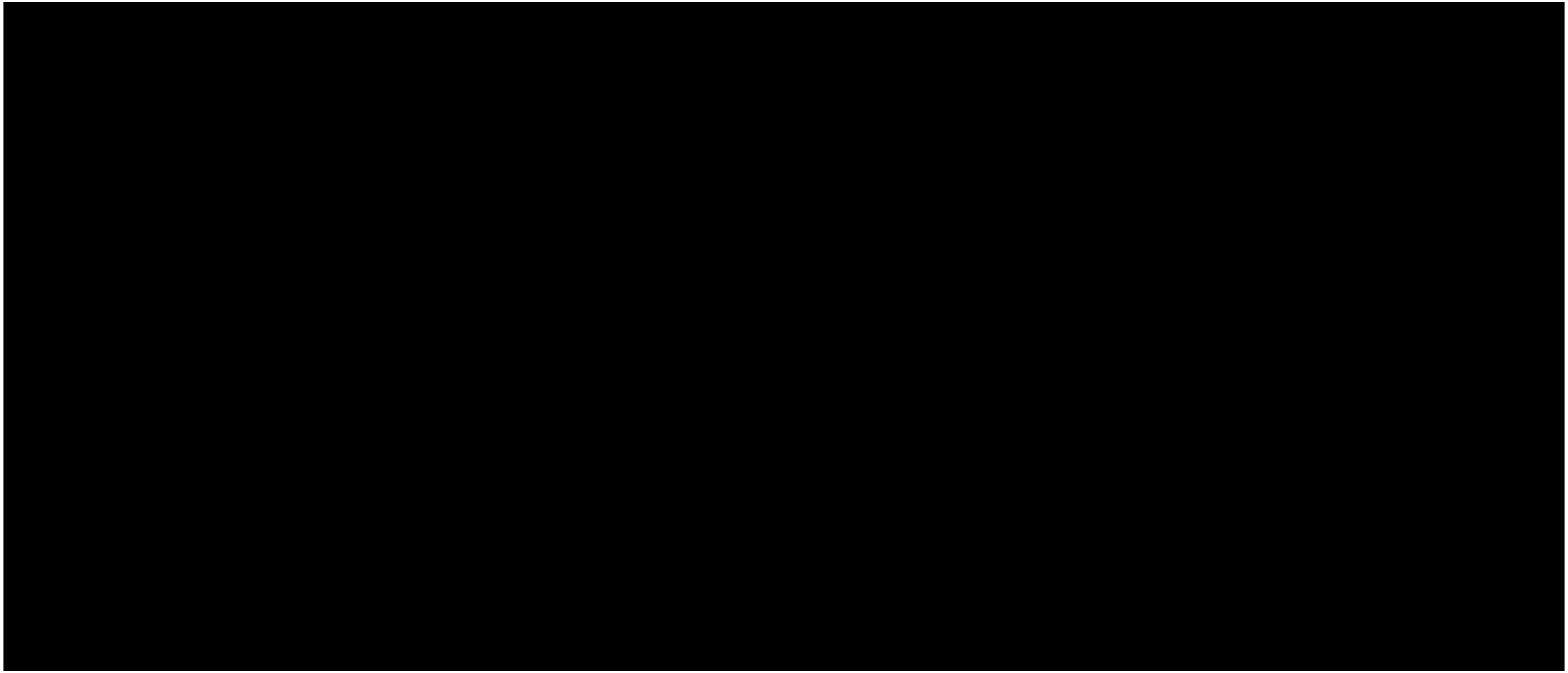
a. State-of-the art Comparison Study:

The existing methods train and show results on a single-room rearrangement task in RoomR. Therefore, for a fair comparison, we train and compare other methods against ours in a single-room setting on RoPOR - Benchmark Dataset.

Number of Objects	Visible Objects	Unseen Objects		Swap Case	Ours-GT			Ours			Weihs *et al.*			Gadre *et al.*			Sarch *et al.*			Ghosh *et al.*		
		OOF	OPR		SNS↑	ENR↑	ATC(m)	SNS↑	ENR↑	ATC(m)	SNS↑	ENR↑	ATC(m)	SNS↑	ENR↑	ATC(m)	SNS↑	ENR↑	ATC(m)	SNS↑	ENR↑	ATC(m)
5	5	0	0	0	**0.98**	NC	**10.57**	**0.74**	NC	**11.98**	0.018	NC	18.11	0.024	NC	20.15	0.058	NC	16.18	0.92	NC	13.58
	5	0	0	2	**0.70**	NC	**12.36**	**0.53**	NC	**13.46**	0	NC	NC	0	NC	NC	0	NC	NC	0.66	NC	16.73
	3	2	0	0	**0.81**	**0.61**	**12.93**	**0.60**	**0.48**	**14.33**	0.002	0.17	19.46	0.003	0.09	20.79	0.046	0.21	18.63	0	NC	NC
	3	0	2	0	**0.79**	**0.60**	**13.39**	**0.58**	**0.47**	**14.89**	0	NC	NC	0.0	NC	NC	0	NC	NC	0	NC	NC
10	10	0	0	0	**0.97**	NC	**22.19**	**0.73**	NC	**24.51**	0.002	NC	34.05	0.008	NC	36.69	0.032	NC	32.52	0.90	NC	27.98
	10	0	0	4	**0.70**	NC	**24.63**	**0.52**	NC	**27.32**	0	NC	NC	0	NC	NC	0	NC	NC	0.65	NC	30.45
	6	4	0	0	**0.84**	**0.69**	**23.78**	**0.64**	**0.53**	**25.56**	0.001	0.20	36.22	0.006	0.12	37.01	0.021	0.23	35.58	0	NC	NC
	6	0	4	0	**0.83**	**0.67**	**24.15**	**0.62**	**0.52**	**25.97**	0	NC	NC	0	NC	NC	0	NC	NC	0	NC	NC
20	20	0	0	0	**0.95**	NC	**40.05**	**0.73**	NC	**44.05**	0	NC	NC	0	NC	NC	0	NC	NC	0.88	NC	50.79
	20	0	0	8	**0.70**	NC	**45.32**	**0.52**	NC	**48.32**	0	NC	NC	0	NC	NC	0	NC	NC	0.62	NC	52.56
	12	8	0	0	**0.87**	**0.75**	**41.29**	**0.67**	**0.58**	**45.29**	0	NC	NC	0	NC	NC	0	NC	NC	0	NC	NC
	12	0	8	0	**0.87**	**0.74**	**42.13**	**0.66**	**0.57**	**45.78**	0	NC	NC	0	NC	NC	0	NC	NC	0	NC	NC

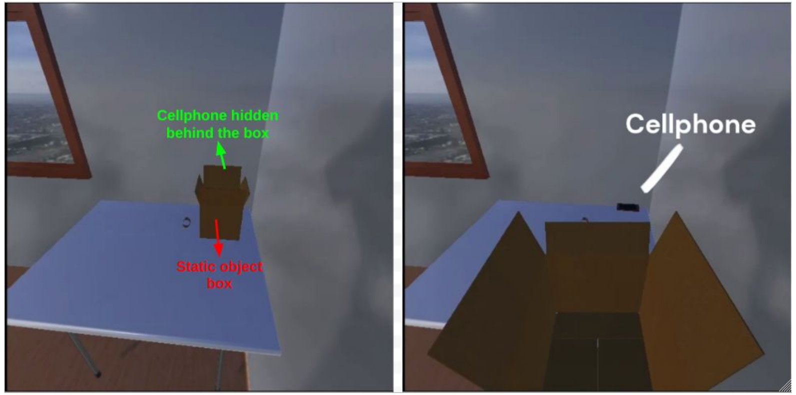
1. Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5922–5931, 2021.
2. Samir Yitzhak Gadre, Kiana Ehsani, Shuran Song, and Roozbeh Mottaghi. Continuous scene representations for embodied ai. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14829–14839, 2022.
3 Sarch, G.; Fang, Z.; Harley, A. W.; Schydlo, P.; Tarr, M. J.; Gupta, S.; and Fragkiadaki, K, TIDEE: Tidying Up Novel Rooms using Visuo-Semantic Commonsense Priors. In European Conference on Computer Vision, 2022
4. Ghosh, S.; Das, D.; Chakraborty, A.; Agarwal, M.; and Bhowmick, B. 2022. Planning Large-scale Object Rearrangement Using Deep Reinforcement Learning. In 2022 International Joint Conference on Neural Networks (IJCNN),1–8.

Qualitative Results



Limitations :

1. Our method has limitations in finding unseen objects that are concealed by other static objects - those that will not be moved during rearrangement because their current and goal locations are identical.
2. Our approach presumes the availability of flawless motion planning and manipulation.



Conclusions :

1. This paper introduces a novel task planner designed for tidying up a room while dealing with partial observability, which can be adapted to various situations and generates a sequence of actions that minimizes the agent's overall traversal and the number of steps taken during simultaneous unseen object search and rearrangement.
2. In future work, we plan to explore the deployment of our method in real-world.

Thank you