

# Mechanical Search on Shelves using Lateral Access X-RAY

Huang Huang<sup>\*1</sup>, Marcus Dominguez-Kuhne<sup>\*1,2</sup>, Vishal Satish<sup>1</sup>, Michael Danielczuk<sup>1</sup>, Kate Sanders<sup>1</sup>, Jeffrey Ichnowski<sup>1</sup>, Andrew Lee<sup>1</sup>, Anelia Angelova<sup>3</sup>, Vincent Vanhoucke<sup>3</sup>, Ken Goldberg<sup>1</sup>

**Abstract**—Efficiently finding an occluded object in a lateral access environment such as a shelf or cabinet arises in many contexts such as warehouses, retail, healthcare, shipping, and homes. While this mechanical search problem has commonly been studied in the overhead access environment, the lateral access setting introduces novel constraints both on the poses of the objects and on available grasp actions that can render pushing actions more efficient in this setting. We propose LAX-RAY (Lateral Access maXimal Reduction in support Area of occupancY distribution): a system that combines a target object occupancy distribution predictions with a mechanical search policy that sequentially pushes occluding objects to efficiently reveal the target. For scenarios with extruded polygonal objects occluding a known stationary target, we introduce two lateral access search policies that encode a history of predicted target distributions and can plan up to three actions into the future. We evaluate these policies both in 200 random shelf environments per policy using a novel First-Order Shelf Simulator (FOSS) and in 5 physical shelf environments using a Fetch robot with an embedded PrimeSense RGBD Camera and an attached blade, where they outperform baselines by up to 25% and up to 60% in physical experiments as the number of occluding objects increases. Additionally, the two-step prediction policy is the highest performing in simulation with an 87.3% average success rate, suggesting a tradeoff between future information and prediction errors. Code, videos, and supplementary material can be found at <https://sites.google.com/berkeley.edu/lax-ray>.

## I. INTRODUCTION

While researchers have explored the problem of *Mechanical Search* in unstructured clutter (in which objects have significant freedom in both position and orientation) [6, 7, 23, 28], mechanical search in semi-structured, lateral access environments such as shelves, cabinets, and closets is a less studied area despite its wide applicability. For instance, a service robot at a pharmacy or hospital may need to find supplies from a cabinet, an industrial robot may need to find kitting tools from shelves in warehouses, or a service robot in a retail store may need to search shelves for requested products from customers. Searching for an object in a lateral access environment poses new challenges not faced in unstructured environments, namely, limited action spaces, complex motion planning requirements, and a potentially limited perception view.

In previous mechanical search instantiations [7, 28, 35], the target object lies in a bin or on a tabletop and is occluded from an overhead view by other objects. The lateral access environment introduces additional constraints, specifically:

<sup>\*</sup>These authors contributed equally. <sup>1</sup>The AUTOLab at University of California, Berkeley. <sup>2</sup>The California Institute of Technology. <sup>3</sup>Robotics at Google.

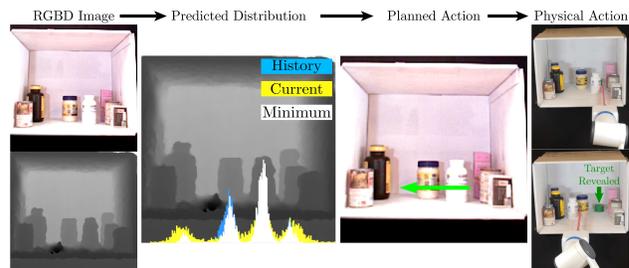


Fig. 1: Lateral-access mechanical search with LAX-RAY. The search starts with an RGBD image of the environment. LAX-RAY perception predicts a distribution over target object locations at the current step (yellow), the previous time step (blue), and the minimum of the two (white). The policy computes a push action indicated by the green arrow, and the robot executes it.

1) objects lie in stable poses due to gravity and do not rest on each other, 2) available grasps are limited due to space constraints within the environment, 3) objects must remain within the environment (i.e., cannot be relocated to an alternate staging area) during the search, and 4) only a lateral view of the environment is available. Grasping objects in structured clutter, including in lateral access environments, has been studied [10, 27, 30], with a major focus on finding an obstacle-free grasping path for a partially occluded object. However, few papers focus on the problem of searching for fully occluded objects in lateral access environments as opposed to grasping visible or partially occluded ones. In practice, collision-free grasping actions in a crowded lateral access environment can be limited due to the size of the gripper or the robot’s wrist. Additionally, keeping objects inside the lateral access environment is often preferred to removing them entirely. These constraints motivate stable pushing actions using a narrow blade end-effector such as the one shown in Figure 1 within such a partially observable environment. With such an end-effector, even more collision-free actions are feasible than in previous works that consider pushing actions, and the actions can be more efficient than iteratively grasping occluding objects while keeping the objects within the environment.

In this work, we train a neural network to predict an occupancy distribution, similar to X-RAY for the overhead access environment [6], using a novel dataset generation method that accounts for shelf depth and camera perspective effects. We propose pushing policies that address the challenges brought by the unique problem setting by encoding a history of the occupancy distribution predictions to account for object motion within the scene and looking ahead to avoid repeated actions. An overview of the resulting policy, Lateral

Access X-RAY (LAX-RAY), is shown in Figure 1 and an example push action in the physical environment is shown in Figure 2.

The contributions of this paper are:

- 1) LAX-RAY Perception: a novel dataset generation pipeline and neural network that predicts target occupancy distributions in lateral-access environments.
- 2) LAX-RAY Pushing Policies: two classes of lateral access mechanical search policies that reveal target objects in shelves based on a history of lateral occupancy prediction and multi-step lookahead, respectively.
- 3) The First Order Shelf Simulator (FOSS), a lightweight and dynamics-free, open-access framework for generating initial shelf configurations and rapidly rolling out lateral access search policies.
- 4) Experiments in simulation and on a physical robot evaluating the policies. Results from 1400 total simulated and 35 total physical trials with 2 to 9 occluding objects suggest that policies leveraging both a history of lateral occupancy distribution predictions and multi-step lookahead outperform baselines without occupancy prediction or without lookahead.

## II. RELATED WORK

While recent work in interactive perception and mechanical search [3, 7, 18, 23, 31, 36] influences our work most strongly, work on Bayesian search - the problem of searching for one or more objects located in one of  $m$  locations [20] - dates back to the 1940s. Assaf and Zamir [1] illustrated that, in many cases, this problem has an optimal greedy solution. Kress *et al.* [22], Lavis *et al.* [24], and Wen *et al.* [32] further extended this work to consider cases of false-positive target detection, moving targets, and the production of sequences of optimal actions, respectively.

Mechanical search extends the classic Bayesian search formulation by introducing physical interaction with objects in the scene, where a robot must uncover, identify, and extract a known target object among unknown distractor objects. Danielczuk *et al.* [7] formalized and explored heuristic mechanical search policies in the overhead bin picking setting and later improved the policy using a neural network-based perception system that predicts areas of the scene likely to occlude the target object [6]. Yang *et al.* [35] extend pushing and grasping work from Zeng *et al.* [37] to the target-specific problem in a tabletop block environment, also with overhead observations, using both a Bayesian exploration policy and a target-centric coordinator policy. Kurenkov *et al.* [23] and Novkovic *et al.* [28] learn nonlinear continuous pushing policies that can reveal a specified target that is partially or fully occluded, respectively. However, each of these papers considers an overhead view of the scene and assumes the objects are lying on a tabletop or in a bin. In lateral access environments, view is limited to lateral access bringing more unknown objects states.

In lateral access environments, where objects lie in their stable poses and not on top of other objects, manipulation is also constrained so that the robot does not collide with the

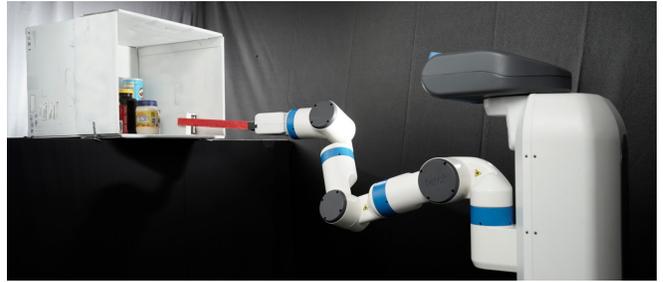


Fig. 2: The physical experiment with a Fetch robot and a shelf environment with random objects. The robot pushes an object a distance and direction calculated by LAX-RAY taking inputs of a RGBD image from the embedded PrimeSense camera.

environment. The Amazon Picking Challenge required teams to pick from or place objects into shelves [15, 38], but did not require object search within the shelf. Dogar *et al.* [12]’s connected components algorithm and POMDP solvers [25, 34] have been used to iteratively remove or rearrange objects in a lateral access environment until the target is revealed, outperforming greedy baselines by reasoning about visibility and accessibility constraints. Bejjani *et al.* [2] use continuous pushing actions generated with a learned prior and lookahead policy in a shelf environment to reach for and successfully grasp a target with up to 10 obstacles in simulation, allowing collisions with other objects. In contrast to their work, we consider 3D object geometries (as opposed to 2D) with a diverse set of target objects. Wong *et al.* [33] also use a geometric prior to inform the search, but they iteratively remove objects from the scene. Gupta *et al.* [17] present a multi-step look-ahead object search algorithm in a shelf environment using a PR2; their problem formulation is the most similar to ours, but we leverage a learned geometric prior to inform the search and use a 1D distribution over possible target poses instead of their 3D octree representation. We don’t have the grid world assumption in both simulation and physical experiments. We also don’t have grasping action available in contrast to their setting.

Given these constraints, pushing is a critical action for moving occluding objects to reveal the target without removing them from the scene entirely. Significant developments have recently been made in the study of singulating cluttered objects on a plane [4, 8, 11, 13, 19, 21, 39]. Notably, Hermans *et al.* [19] propose a pushing policy that includes a history of past actions used to determine when termination should occur. Eitel *et al.* [14] select push actions through the use of a neural network trained on physical data collected from robots interacting with cluttered scenes. The majority of the environments explored in these studies have relaxed constraints regarding object configurations and possible pushing actions compared to environments such as shelves, where a limited number of pushing actions are viable.

## III. PROBLEM STATEMENT

We consider an instance of the mechanical search problem in a lateral access environment (e.g., a shelf). In this problem, an agent searches for a target object resting in a stable



Fig. 3: Target objects spanning a variety of aspect ratios (noted below each object) used in simulation testing (left) and in physical experiments (right).

pose on the shelf, initially occluded by other objects. The geometry of the target object is known, and the agent has an RGBD camera with a fixed side-view of the shelf and known intrinsics. The agent reveals the target object by pushing objects on the shelf to the left and right using a narrow blade, shown in Figure 2, but cannot remove objects from the shelf.

We make the following assumptions:

- Exactly one instance of the search target exists in the environment.
- All objects are extruded polygons and are not stacked.
- Toppling does not occur.
- The target object is not moved during the search.

We model the problem as having states, observations, actions, and a termination condition. The state  $\mathbf{s}_t$  at time  $t$  consists of the scene object geometries and their poses within the shelf. At each time  $t$ , the camera takes an RGBD image observation of width  $w$  and height  $h$ ,  $\mathbf{y}_t \in \mathbb{R}_+^{h \times w \times 4}$ , of the scene from its fixed pose. The robot then takes a push action  $\mathbf{a}_t = (\mathbf{q}, d)$ , where  $\mathbf{q} \in \mathbb{R}^3$  is the pushing starting point in the camera frame and  $d \in \mathbb{R}$  is the pushing distance along the camera’s  $x$ -axis, which results in a new state and observation. This process continues until at least  $k\%$  of the target object is visible to the camera or no further pushing actions can be taken by the robot.

Our objective is to find a policy  $\pi$  that, given an observation  $\mathbf{y}_t$  of the underlying state  $\mathbf{s}_t$  at each time step, can minimize the number of actions taken to reveal the target object.

#### IV. METHODS

LAX-RAY combines (1) a perception pipeline predicts an occupancy distribution for the (partially) occluded target object that spans across the visible objects in the scene given the depth image and target object segmentation mask with (2) a search policy that chooses pushing actions to reveal the target based on a history of predicted occupancy distributions.

##### A. LAX-RAY Perception

Predicting an occupancy distribution in the lateral access environment differs from the overhead environment in two main ways: 1) the objects are known to be in a stable pose resting on the plane of the shelf, so target object translations are restricted to the  $xz$  plane of the camera, and 2) camera perspective effects can result in different occupancy predictions for the same occluding object depending on its depth

within the shelf. In contrast, X-RAY predictions consider flat objects that can assume any rotation and may translate within a single depth plane of the camera [6]. We address these differences by sampling target object transformations within the camera’s  $xz$  plane and rendering images for each transformation.

We generate a large dataset of synthetic depth images, target object segmentation masks, and ground truth target object occupancy distributions to train a neural network to predict the target object occupancy distribution within the image plane, as in X-RAY. However, the method of dataset generation must be extended beyond iteratively dropping objects in a heap for the lateral access environment. A scene is generated using a dataset of object meshes by first placing the target on the shelf in one of its stable poses and apply a random translation such that it does not collide with the walls of the shelf. Occluding objects are sampled randomly from a dataset of 88 Google Scanned Objects [16] models and are iteratively placed in the same manner such that they do not collide with each other, the target, or the shelf. We sample  $N \sim U(10, 20)$  occluding objects per scene. A  $256 \times 256$  depth image and target object segmentation mask is rendered from the side view looking into the shelf. Note that if the target object is fully occluded, the target segmentation mask will be blank.

We then consider 2992 target object transformations along a uniform 2D grid in the camera frame, 17 translations along the camera’s  $x$  axis, 22 translations along the camera’s  $z$  axis, and 8 rotations about the camera’s  $y$  axis. We choose this number of transformations to trade off ground-truth distribution accuracy with dataset generation time - doubling the number of rotations for a fixed number of translations results in a total variation distance of only  $6e-4$  between the two distributions as computation time increases by 74% and doubling the number of translations for a fixed number of rotations similarly yields a total variation distance of 0.002 as computation time increases by 51%. Thus, incorporating more transformations into the ground truth occupancy distribution results in a similar result at the cost of significantly increased computation time. For each of these transformations, we render a binary image of the target object by itself and a depth image of the scene with the transformed object. If the depth image of the scene with the transformed object matches the original scene depth image, we add the binary image to the ground truth occupancy distribution. The final ground truth occupancy distribution is a normalized sum of the binary images that correspond to target transformations resulting in the same scene depth image (i.e., all possible pixels that could contain any part of the target object, with values corresponding to their relative likelihood). We render these images and compute the corresponding distribution across 5 camera positions uniformly sampled from a range of the view sphere for each scene to mitigate uncertainty of the camera position in physical experiments.

We repeat this process for 6,000 unique scenes and generate a total of 30,000 depth image, target object mask, and ground-truth occupancy distribution training tuples. Due the

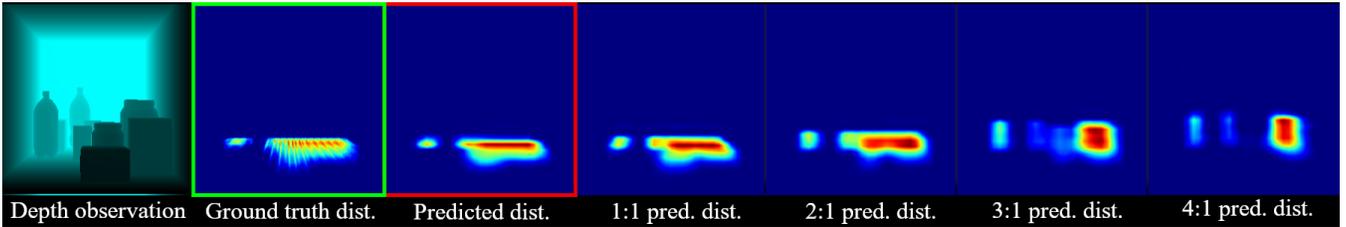


Fig. 4: LAX-RAY occupancy distribution predictions on a simulated image from the test set for a target object with 1:2 aspect ratio in a fully occluded case. The fourth to sixth images show the network predictions for aspect ratios from 1:1 to 4:1. The large variation in the predictions implies the critical influence of the target aspect ratios on the occupancy distribution.

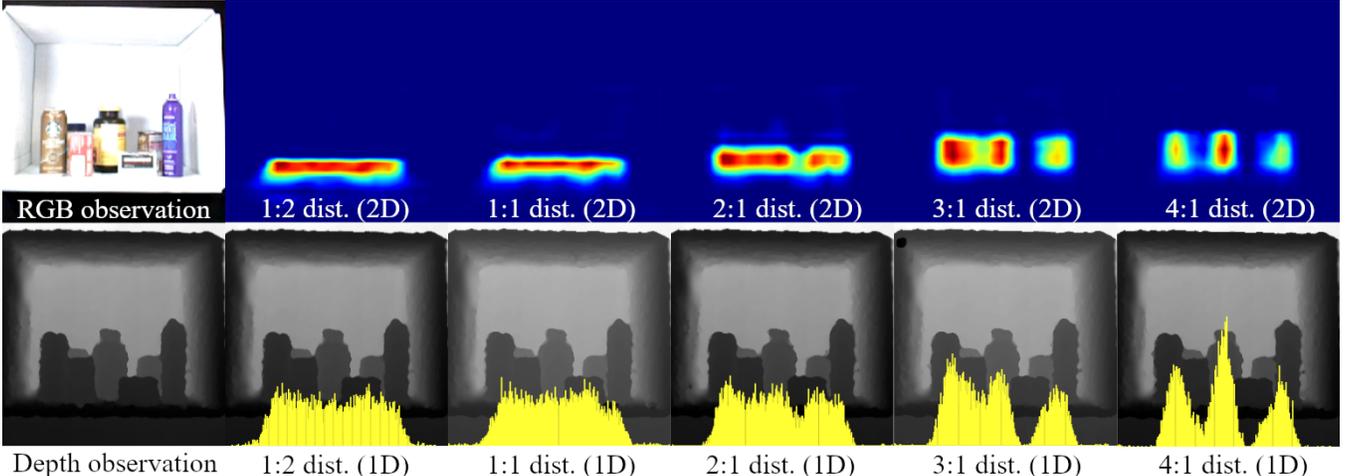


Fig. 5: Validation of the pre-trained model on physical experiment environment for 5 different target object with aspect ratios from 1:2 to 4:1 in a fully occluded case. The top row shows 2D occupancy distribution from the pre-trained model and the second row shows the corresponding 1D occupancy distribution along  $x$  axis in camera frame overlaid with the depth observation. A significant difference for predictions of each aspect ratio is shown validating the accuracy of the pre-trained model for real applications.

aspect ratio’s dominating effect on the occupancy distribution, we used five cuboid target objects with varying aspect ratios from 1:2 to 4:1 and apply the prediction from the aspect ratio closest to the target object at test time. About 50% of these images include a fully occluded target object. We also render 10,000 images using occluding object models sampled from 44 object models unseen in training from the Google Scanned Objects dataset [16] with realistic target objects, shown in Figure 3, to create a test set.

We split the data into training and validation sets with a 4:1 ratio. We train a fully convolutional network (FCN) with a ResNet backbone with 33 million parameters on the dataset using stochastic gradient descent with a momentum 0.99 for 60000 iterations with batch size 32 and learning rate  $10^{-5}$ . Training takes approximately 5 hours on a Tesla V100 GPU.

### B. LAX-RAY Search Policies

We propose two classes of policies that leverage the LAX-RAY perception output over several timesteps and iteratively attempt to reduce either the support or the entropy of the history of predicted target occupancy distributions.

In all policies, we first generate a set of candidate pushes by considering possible push starting locations  $\mathbf{q} = (q_x, q_y, q_z)$  and associated distances  $d$  based on the other objects in the scene. In addition to the output provided by the perception pipeline, all policies receive a segmentation image of the scene as input. In practice, we use ground-

truth segmentation masks in simulated experiments and a gradient-based edge detector segmentation method in physical experiments, although any state-of-the-art segmentation method could be used.

For each object segmentation mask, up to two possible push candidate  $q_x$  values are generated using the edges of the mask, depending on available space to both insert the blade and push the object. We choose the blade height  $q_y$  to be a fixed value just above the support surface such that objects will not topple as they slide, and we choose the blade depth  $q_z$  to be at the midpoint of the object depth, assuming that the object depth is linearly proportionally to its width. Note that both of these choices might be further improved by introducing shape completion [29] and explicitly estimating object centers of mass. The horizontal pushing distance  $d$  given a starting point  $\mathbf{q}$  is the distance the object can be pushed until it collides either with another object or the shelf and its sign is given by the relative position of  $\mathbf{q}$  to the object to be pushed. This collision avoidance pushing actions allows us to maintain the stationary target object position assumption by avoiding pushing in an unknown area.

Then, we make use of both the current predicted distribution  $P_t(x)$  and an encoding of previous predicted distributions  $P'_{t-1}(x)$  at each timestep. Our first insight is that we can reduce the 3D or 2D distributions over possible target object poses from previous work [6, 17, 29] to a 1D distribution in the shelf setting, since all objects lie on the shelf’s

support surface. Specifically, we collapse the 2D prediction  $p_t(x, y)$  generated by LAX-RAY perception by summing over the  $y$ -axis of the image:  $P_t(x) = \sum_{y=0}^{h-1} p_t(x, y)$ , as seen in Figure 5. Our second insight, consistent with [17], is that an encoding of previous observations is necessary to avoid repeated actions, since objects cannot be removed from the shelf. In our policies, we encode the history of previous observations via the minimum of the current 1D predicted distribution and the previous encoding:  $P'_t(x) = \min\{P_t(x), P'_{t-1}(x)\}$ . For the first timestep, we set  $P'_t(x) = P_t(x)$ . Our third insight, similar to [6], is that minimizing the *entropy* of the 1D distribution over target object poses across multiple timesteps should reveal the target object, where the entropy is defined as:

$$c_t = - \sum_{x=0}^{w-1} P'_t(x) \log(P'_t(x)),$$

Given these insights, we propose two classes of policies:

**Distribution Area Reduction (DAR)** computes a score based on the width of the push object’s segmentation mask in the image, weighted by  $P'_t(x)$  at those pixels:  $\sum_{x=0}^{w-1} I_o(x) P'_t(x)$ , where  $I_o(x) \in \{0, 1\}$  indicates whether an object  $o$ ’s segmentation mask is nonzero at that  $x$  value in the image (for any  $y$  value). It then considers the difference in this score before and after each action is taken, and selects the action that reduces the score the most. The post-action score is computed by translating the object segmentation mask in the image according to the action and recomputing based on the same  $P'_t(x)$ . If the target object is partially revealed, the policy does not take an action that would cover the target more. This policy is maximizing the occupancy information gain by minimizing the overlap of the object with the occupancy distribution.

**Distribution Entropy Reduction over  $n$  steps (DER- $n$ )** estimates the 1D distribution  $\hat{P}'_{t+n}$  after taking  $n$  actions and chooses the action it predicts will produce the smallest entropy value  $\hat{c}(t+n)$  at time step  $t+n$  (assuming optimal actions are taken given the current information), for  $n \geq 1$ . First, the observation  $\mathbf{y}_{t+1}$  is predicted by translating the pixels belonging to the object to be pushed across the image. Pixels behind the moving object are assumed to be of maximum depth unless a partially observed object lies behind them. This assumption to decrease the number of predicted actions which increases exponentially with prediction steps. Then, the occupancy distribution  $\hat{P}'_{t+1}$  is predicted for the estimated observation and the process is repeated for  $n$  steps.

## V. FIRST ORDER SHELF SIMULATOR (FOSS)

To evaluate the proposed policies, we develop a First Order Shelf Simulator (FOSS), a Python-based simulator that uses `trimesh` [9] and `pyrender` to model object pushing actions within a parametrizable shelf environment without explicitly modeling contact dynamics such as friction. In contrast to commonly-used simulators such as `pybullet` [5], `Isaac Sim`, or others that use physics-based simulation [26],

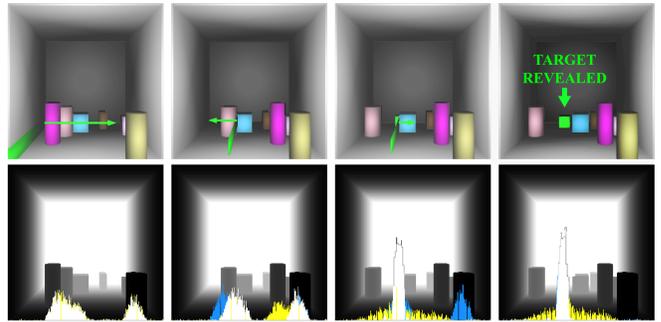


Fig. 6: The DER-1 policy reveals a hidden object with 6 occluding objects. First row shows color images at each step with a green blade, and a green arrow denoting push direction and distance. The occupancy distribution at the bottom of each depth image (second row) includes: predicted distribution from previous time step (**blue**), predicted distribution at current time step (**yellow**), and minimum of the two (**white**). Target is revealed in last image, where the occupancy distribution has a very dense probability.

FOSS only performs collision checking between objects - all actions are taken by translating objects along a line parallel to the shelf support surface. The lack of dynamic simulation allows FOSS to be more efficient than other simulators for pushing for use in large-scale policy rollouts while maintaining a feasible scene configuration at all times.

FOSS represents objects as extruded polygons (cuboids and cylinder approximations). Cuboids uniformly range from 0.02 to 0.10 m for the square base and are 0.1 m tall. Cylinders uniformly range from 0.02 to 0.05 m for the base radius and are between 0.1 and 0.2 m tall. A green target object cube with sides of size 0.07 m and aspect ratio of 1:1 is used.

FOSS checks collisions between targets and shelf wall. If an object collides with another object during the push, both objects move together for the remainder of the push in the push direction. When an object collides with the shelf wall, the push action stops. Notice the pushing action given by the policy will avoid potential collision between objects.

Although FOSS allows for pushes in 2D, we focus on 1D lateral pushes in this paper. For simulation runs we place objects randomly on the shelf while checking for collisions, as described in Section IV-A, to ensure object separation. Furthermore, to guarantee possible initial pushes, objects are initialized with at least the blade thickness away from the wall. FOSS can produce both RGBD images and ground truth segmentation masks at each timestep.

## VI. EXPERIMENTS

We evaluate both the perception system and the proposed policies both in simulation and in a physical shelf environment (W:50cm, H:40cm, D:56cm).

### A. Perception Experiments

We benchmark the model in simulation using an intersection-over-union metric defined as the sum of positive pixels in both the ground truth and predicted distributions divided by the sum of total positive pixels in either distribution. We binarize the prediction into positive and negative

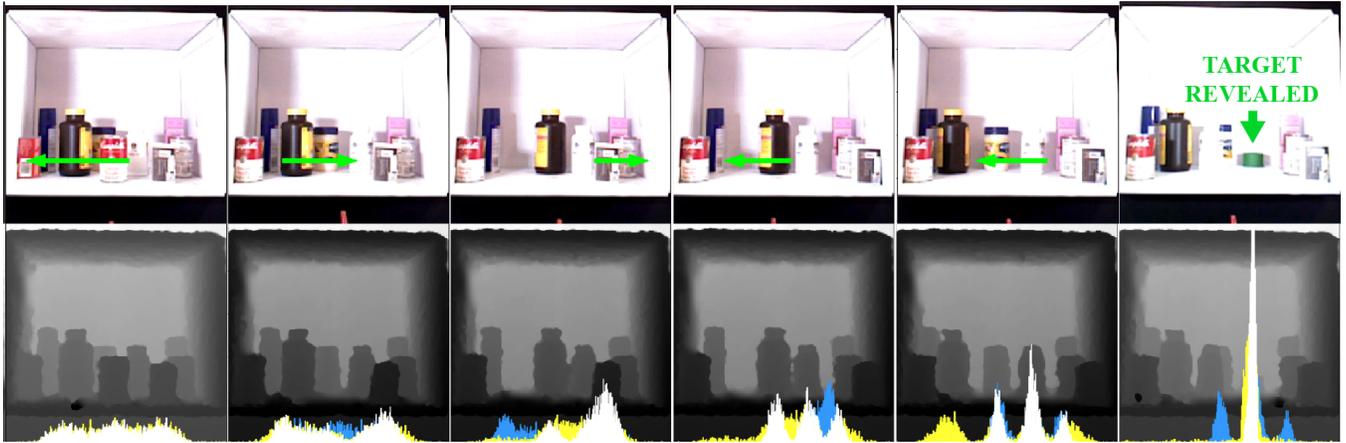


Fig. 7: The DER-1 policy reveals the target object among 9 occluding objects in a physical experiment. The plotted occupancy distributions at the bottom of each depth image including three parts as in Figure 6. **Top:** RGB image with green push action arrow denoting the pushing direction and distance. **Bottom:** depth image with occupancy distributions.

pixel predictions by thresholding at a normalized value of 0.1 and measuring error to the ground truth distribution. Pixels with values greater than 0.1 within 0.2 of ground truth are labeled as true positives, and values less than 0.1 within 0.2 of ground truth are true negatives [6]. On images in simulation, the model yields a validation IoU of 0.79 and a test IoU of 0.53. Example images from the test set are shown in Figure 4 in the case where the target object is fully occluded. The predicted distributions vary significantly across target object aspect ratios, indicating that the predicted distributions is correctly sensitive to the aspect ratio.

We also benchmark the model on images from the physical setup, using green target objects with similar sizes and aspect ratios to those used in simulation, including cuboids and cylinders, as shown in Figure 3. We put common household objects on a white shelf and use a PrimeSense RGBD camera embedded on a Fetch robot to acquire color and depth images. We segment the target object if it is visible using HSV color thresholding to detect the green target object. Figure 5 shows the prediction results of the pre-trained model on a shelf with randomly arranged objects in a fully occluded case for five target objects with varying aspect ratios. Qualitatively, the model transfers well to the real images and the predicted distribution again varies significantly with different aspect ratios.

### B. Search Policy Simulation Experiments

We use FOSS to rapidly test LAX-RAY across many scenarios and evaluate the insights noted in Section IV. We evaluate whether minimizing entropy over multiple future time steps can outperform a greedy policy by comparing DAR and DER-n. In addition, we evaluate whether 1) LAX-RAY outperforms iteratively searching, 2) the predicted LAX-RAY occupancy distribution can help the policy reveal the target more efficiently, 3) a history encoding of past observations aids shelf exploration. To this end, we introduce three baseline policies:

**Scan** first pushes all objects to one side of the shelf, and then, when no pushes in that direction can be found, pushes

TABLE I: Simulation results for 7 policies over 200 simulated rollouts each. 50 trials were executed for 2, 4, 6, and 8 occluding objects each, and for each we show the **Success Rate** followed by the **Average of Steps** and **Standard Deviation** of steps. As DER-n assume no unseen objects are behind the pushed object, DER-n policies have higher prediction errors as n increases. Addressing this is an important topic for future research.

No.	Results	SCAN	GUPTA-2013	M-DAR	DAR	DER-1	DER-2	DER-3
2	Succ.	98%	97%	90%	98%	98%	97%	94%
	Step Avg	1.53	1.34	1.37	1.36	1.36	1.80	1.99
	Step Std	0.95	0.56	0.72	0.77	0.67	0.93	1.19
4	Succ.	82%	88%	76%	96%	92%	92%	89%
	Step Avg	3.24	2.27	2.13	2.33	2.37	2.57	3.21
	Step Std	2.42	1.48	1.52	1.78	1.79	1.93	2.02
6	Succ.	62%	71%	58%	87%	79%	89%	85%
	Step Avg	3.06	2.99	2.58	2.79	3.04	3.32	3.82
	Step Std	2.56	2.21	2.00	2.06	2.30	2.14	2.44
8	Succ.	58%	46%	53%	66%	63%	71%	66%
	Step Avg	3.59	3.61	2.44	3.40	3.13	3.94	4.08
	Step Std	2.30	2.75	1.68	2.47	2.25	2.65	2.54
Avg	Succ.	75.0%	75.5%	69.3%	86.8%	83.0%	87.3%	83.5%
	Step Avg	2.86	2.56	2.13	2.47	2.48	2.91	3.28

TABLE II: Physical experiment results. The first column shows the number of occluding objects. The remaining columns show **task success (top)** with Y denoting success and N denoting fail and the number of **actions to succeed (bottom number)**. The last row shows the average of the 5 experiments.

No.	Results	SCAN	GUPTA-2013	M-DAR	DAR	DER-1	DER-2	DER-3
2	Success	Y	Y	N	Y	Y	Y	Y
	Steps	3	3	10	3	3	1	1
4	Success	Y	Y	Y	Y	Y	Y	Y
	Steps	4	1	1	1	2	1	1
6	Success	N	Y	Y	Y	Y	Y	Y
	Steps	3	3	4	4	4	6	6
8	Success	N	N	N	N	Y	Y	Y
	Steps	4	4	6	6	3	4	3
9	Success	Y	N	N	Y	Y	Y	Y
	Steps	3	9	3	5	5	3	5
Avg	Success	60%	60%	40%	80%	100%	100%	100%
	Steps	3.4	4	4.8	3.8	3.4	3	3.2

the objects to the other side of the shelf. The initial direction is set as the direction that will push objects towards the less-occluded half of the shelf based on the depth observation at  $t = 0$ . Pushes are executed in the order of how far the pushed object moves.

**Gupta-2013** is an analogue of the policy in [17] without grasping and lookahead, but with a history encoding. [17] maximize a 3D information gain; here, we maximize a 2D information gain in width considering the gravity constraint brought by the lateral access environment. This policy can

also be thought of an analogue of the “largest” policy with additional history encoding in the overhead setting [6, 7]. It can be interpreted as forming an occupancy distribution that informed by the target object shape, only by the occluding object shapes, which is an ablation of DAR policy without target object geometric prior.

**Markov-DAR (M-DAR)** is an ablation of the DAR policy without encoding a history of occupancy distributions. At each time step, this policy plans an action based on the current observation. This policy addresses the importance of remembering history in the environment where objects are not removed and an explored area may be occluded again the future.

We generated 50 random scenes, as described in Section V, with 2, 4, 6, and 8 occluding objects, giving us 200 scenes in total per policy. We tested 3 prediction steps of DER- $n$  for  $n \in \{1, 2, 3\}$  on each scene. A rollout is considered successful if at least 90% of the target object is revealed within 10 actions. The policies’ performance in simulation is summarized in Table I with an example successful DER-1 policy rollout shown in Figure 6.

Table I shows Scan performs better than Gupta-2013 and M-DAR in general but much worse than DAR and DER in most cases suggesting the limitation of iteratively searching in the lateral access environment setting. DAR policies perform better than the Gupta-2013 policy, especially as the number of objects increases. This discrepancy suggests that taking the target object’s shape into account for the occupancy distribution prediction aids policy efficiency. A better performance of DAR compared to M-DAR implies the importance of history encoding avoiding repeated actions. The results validate the efficiency and novelty of LAX-RAY by integrating the target geometric prior through perception pipeline with history encoding and looking ahead. All policies’ performances drop when the number of occluding objects increases since more steps are needed to reveal the target. For higher numbers of occluding objects, DER-2 outperforms both DAR and DER-3. This result suggests both that minimizing the distribution’s entropy over several timesteps can outperform a policy that greedily tries to maximally reduce the distribution at the current timestep and that a tradeoff emerges between prediction accuracy and prediction lookahead. The prediction of the next step assumes no unseen objects are behind the pushed object, which can deviate from reality. DER-1 does not perform as well as DER-2 because not enough future information is revealed while DER-3 performs worse because prediction error accumulates. When there are fewer objects, prediction errors dominate this tradeoff, which could explain the better performance of DAR in this setting.

### C. LAX-RAY Physical Experiments

To evaluate the policies in a physical lateral access environment with the perception pipeline, we placed household kitchen and bathroom objects on a shelf and executed the insertion and pushing actions using a Fetch robot with a blade attached to the gripper. An embedded PrimeSense

camera is used for taking RGBD observations. Figure 2 shows the setup. At each time step, we execute the pushing action planned by the policy in 5 steps starting and ending at the same fixed position. By dividing the pushing action into those 5 steps, we avoid the occlusion of the robot arm to the camera and any potential collisions. We tested 5 different layouts for each policy with 2 occluding objects, 4 occluding objects, 6 occluding objects, 8 occluding objects and 9 occluding objects respectively. The target object with aspect ratio of 1:2 from Fig. 3 is used and is fully occluded in all layouts.

The performance of each policy is summarized in Table II. The results suggest consistency with the simulation results in that the M-DAR baseline policy has the lowest success rate. And in physical experiments, DER-2 shows the best performance with lowest average number of steps. DAR and DER- $n$  outperforms all the baseline policies showing the validation of LAX-RAY in practical environment with perception noise and diverse objects. Figure 7 shows an action series for experiment 5 with DER-1. In step 4, where the middle white bottle has the highest sum of occupancy distribution, the policy pushes the brown bottle away firstly to make pushing space for the pushing the white bottle which is occluding the target.

## VII. CONCLUSION AND FUTURE WORK

This paper considers the Mechanical Search problem in the lateral access environment with pushing actions, a setting with many practical applications. We introduce additional challenges that distinguish our work from previous work with overhead views or where grasping is less constrained and motivates a novel target occupancy distribution prediction pipeline as well as pushing actions such that objects remain confined to the shelf. The evaluation results of LAX-RAY both in simulation and on the physical system suggest history encodings, learned target occupancy distributions, and multi-step lookahead are all key parts of an efficient mechanical search policy. While our current policy does not explicitly consider pushing multiple objects, stacked objects, or choosing pushes that may rotate objects in the scene, we look to address these challenges in future work. While traditional grasping using a gripper is limited in this environment, using pneumatically-activated suction cups to lift and pull occluding objects from the shelf may also create more available actions.

### ACKNOWLEDGEMENTS

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab. This research was supported in part by: NSF National Robotics Initiative Award 1734633 and by a Focused Research Award from Google Cloud. The authors were supported in part by donations from Google and Toyota Research Institute, the National Science Foundation Graduate Research Fellowship Program under Grant No. 1752814, and by equipment grants from PhotoNeo and NVidia. We thank our colleague Daniel Seita who provided helpful feedback and suggestions.

### REFERENCES

- [1] D. Assaf and S. Zamir, “Optimal sequential search: A bayesian approach,” *The Annals of Statistics*, pp. 1213–1221, 1985.

- [2] W. Bejjani, W. C. Agboh, M. R. Dogar, and M. Leonetti, "Occlusion-aware search for object retrieval in clutter," *arXiv preprint arXiv:2011.03334*, 2020.
- [3] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action," *IEEE Trans. Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017.
- [4] L. Chang, J. R. Smith, and D. Fox, "Interactive singulation of objects from a pile," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2012, pp. 3875–3882.
- [5] E. Coumans and Y. Bai, *Pybullet, a python module for physics simulation for games, robotics and machine learning*, <http://pybullet.org>, 2016–2019.
- [6] M. Danielczuk, A. Angelova, V. Vanhoucke, and K. Goldberg, "X-ray: Mechanical search for an occluded object by minimizing support of learned occupancy distributions," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020.
- [7] M. Danielczuk, A. Kurenkov, A. Balakrishna, M. Matl, D. Wang, R. Martin-Martin, A. Garg, S. Savarese, and K. Goldberg, "Mechanical search: Multi-step retrieval of a target object occluded by clutter," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2019, pp. 1614–1621.
- [8] M. Danielczuk, J. Mahler, C. Correa, and K. Goldberg, "Linear push policies to increase grasp access for robot bin picking," in *Proc. IEEE Conf. on Automation Science and Engineering (CASE)*, IEEE, 2018, pp. 1249–1256.
- [9] Dawson-Haggerty et al., *Trimesh*, version 3.2.0, Dec. 8, 2019.
- [10] M. Dogar, K. Hsiao, M. Ciocarlie, and S. Srinivasa, "Physics-based grasp planning through clutter," in *Proc. Robotics: Science and Systems (RSS)*, 2012.
- [11] M. Dogar and S. Srinivasa, "A framework for push-grasping in clutter," *Proc. Robotics: Science and Systems (RSS)*, vol. 1, 2011.
- [12] M. R. Dogar, M. C. Koval, A. Tallavajhula, and S. S. Srinivasa, "Object search by manipulation," *Autonomous Robots*, vol. 36, no. 1, pp. 153–167, 2014.
- [13] Z. Dong, S. Krishnan, S. Dolasia, A. Balakrishna, M. Danielczuk, and K. Goldberg, "Automating planar object singulation by linear pushing with single-point and multi-point contacts," in *Proc. IEEE Conf. on Automation Science and Engineering (CASE)*, IEEE, 2019, pp. 1429–1436.
- [14] A. Eitel, N. Hauff, and W. Burgard, "Learning to singulate objects using a push proposal network," in *Robotics Research*, Springer, 2020, pp. 405–419.
- [15] C. Eppner, S. Höfer, R. Jonschkowski, R. Martin-Martin, A. Sieverling, V. Wall, and O. Brock, "Lessons from the amazon picking challenge: Four aspects of building robotic systems.," in *Proc. Robotics: Science and Systems (RSS)*, 2016.
- [16] *Google scanned objects dataset*, <https://app.ignitionrobotics.org/GoogleResearch/fuel/collections/Google%20Scanned%20Objects>.
- [17] M. Gupta, T. Rühr, M. Beetz, and G. S. Sukhatme, "Interactive environment exploration in clutter," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2013, pp. 5265–5272.
- [18] M. Gupta and G. S. Sukhatme, "Using manipulation primitives for brick sorting in clutter," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2012, pp. 3883–3889.
- [19] T. Hermans, J. M. Rehg, and A. Bobick, "Guided pushing for object singulation," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2012, pp. 4783–4790.
- [20] B. O. Koopman, "Search and screening, operations evaluation group report 56," *Center for Naval Analysis, Alexandria, Virginia*, 1946.
- [21] M. Kopicki, J. Wyatt, and R. Stolkin, "Prediction learning in robotic pushing manipulation," in *2009 International Conference on Advanced Robotics*, IEEE, 2009, pp. 1–6.
- [22] M. Kress, K. Y. Lin, and R. Szechtman, "Optimal discrete search with imperfect specificity," *Mathematical methods of operations research*, vol. 68, no. 3, pp. 539–549, 2008.
- [23] A. Kurenkov, J. Taglic, R. Kulkarni, M. Dominguez-Kuhne, R. Martín-Martín, A. Garg, and S. Savarese, "Visuomotor mechanical search: Learning to retrieve target objects in clutter," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020.
- [24] B. Lavis, T. Furukawa, and H. F. D. Whyte, "Dynamic space re-configuration for bayesian search and tracking with moving targets," *Autonomous Robots*, vol. 24, no. 4, pp. 387–399, 2008.
- [25] J. K. Li, D. Hsu, and W. S. Lee, "Act to see and see to act: Pomdp planning for objects search in clutter," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 5701–5707.
- [26] C. Mitash, A. Boularias, and K. Bekris, "Physics-based scene-level reasoning for object pose estimation in clutter," *Int. Journal of Robotics Research (IJRR)*, p. 0278364919846551, 2019.
- [27] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-dof grasping for target-driven object manipulation in clutter," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2020, pp. 6232–6238.
- [28] T. Novkovic, R. Pautrat, F. Furrer, M. Breyer, R. Siegwart, and J. Nieto, "Object finding in cluttered scenes using interactive perception," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2020, pp. 8338–8344.
- [29] A. Price, L. Jin, and D. Berenson, "Inferring occluded geometry improves performance when retrieving an object from dense clutter," in *Int. S. Robotics Research (ISRR)*, 2019.
- [30] A. Saxena, L. Wong, M. Quigley, and A. Y. Ng, "A vision-based system for grasping novel objects in cluttered environments," in *Robotics research*, Springer, 2010, pp. 337–348.
- [31] H. Van Hoof, O. Kroemer, and J. Peters, "Probabilistic segmentation and targeted exploration of objects in cluttered environments," *IEEE Trans. Robotics*, vol. 30, no. 5, pp. 1198–1209, 2014.
- [32] Z. Wen, B. Kveton, B. Eriksson, and S. Bhamidipati, "Sequential bayesian search," in *Proc. Int. Conf. on Machine Learning*, 2013, pp. 226–234.
- [33] L. L. Wong, L. P. Kaelbling, and T. Lozano-Pérez, "Manipulation-based active search for occluded objects," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2013, pp. 2814–2819.
- [34] Y. Xiao, S. Katt, A. ten Pas, S. Chen, and C. Amato, "Online planning for target object search in clutter under partial observability," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2019, pp. 8241–8247.
- [35] Y. Yang, H. Liang, and C. Choi, "A deep learning approach to grasping the invisible," *IEEE Robotics & Automation Letters*, vol. 5, no. 2, pp. 2232–2239, 2020.
- [36] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee, *Transporter networks: Rearranging the visual world for robotic manipulation*, 2020. arXiv: 2010.14406 [cs.RO].
- [37] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 4238–4245.
- [38] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2017, pp. 1386–1383.
- [39] C. Zito, R. Stolkin, M. Kopicki, and J. L. Wyatt, "Two-level rrt planning for robotic push manipulation," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2012, pp. 678–685.