

Semantic Mechanical Search with Large Vision and Language Models

Satvik Sharma^{1*}, Huang Huang^{1*}, Kaushik Shivakumar¹
 Lawrence Yunliang Chen¹, Ryan Hoque¹, Brian Ichter², Ken Goldberg¹

Abstract: Moving objects to find a fully-occluded target object, known as *mechanical search*, is a challenging problem in robotics. As objects are often organized semantically, we conjecture that semantic information about object relationships can facilitate mechanical search and reduce search time. Large pretrained vision and language models (VLMs and LLMs) have shown promise in generalizing to uncommon objects and previously unseen real-world environments. In this work, we propose a novel framework called Semantic Mechanical Search (SMS). SMS conducts scene understanding and generates a semantic occupancy distribution explicitly using LLMs. Compared to methods that rely on visual similarities offered by CLIP embeddings, SMS leverages the deep reasoning capabilities of LLMs. Unlike prior work that uses VLMs and LLMs as end-to-end planners, which may not integrate well with specialized geometric planners, SMS can serve as a plug-in semantic module for downstream manipulation or navigation policies. For mechanical search in closed-world settings such as shelves, we compare with a geometric-based planner and show that SMS improves mechanical search performance by 24% across the pharmacy, kitchen, and office domains in simulation and 47.1% in physical experiments. For open-world real environments, SMS can produce better semantic distributions compared to CLIP-based methods, with the potential to be integrated with downstream navigation policies to improve object navigation tasks. Code, data, videos, and the appendix are available [here](#).

Keywords: Vision and Language Models, Mechanical Search, Object Search

1 Introduction

Mechanical search, where a robot manipulates objects and/or navigates to find a fully occluded target object [1, 2], is a challenging robotics problem. Prior work has shown success in revealing the desired object by manipulating the occluding objects [3, 4, 5], obtaining new observations after rotating the camera [6], or navigating to new locations [7, 8]. However, generalization to unseen environments remains challenging due to the numerous long-tail objects present in the real world.

Environments are often organized semantically, for example, toothpaste is often stored in a home bathroom near toothbrushes. In this paper, we explore how LLMs can provide such semantic relationships to facilitate mechanical search. Large vision and language models (VLMs and LLMs) show promise for such relationships as they are pretrained on internet-scale data which empirically captures knowledge of semantics. A large body of prior work has shown that these models can provide good visual representations [9, 10, 11, 12, 13], ground language instructions [14, 15, 16, 17, 18, 19, 20], and serve as planners out of the box [21, 22, 23, 24, 25, 26]. CLIP [27] is a commonly-used interface to associate vision and language, and many works [7, 8, 28, 29] use it to build semantic scene representations and show improved performance on object query and navigation tasks. However, while informative, the dot product of CLIP text and image embeddings lacks deep reasoning

*Equal Contribution

¹AUTOLab at the University of California, Berkeley

²Google Brain

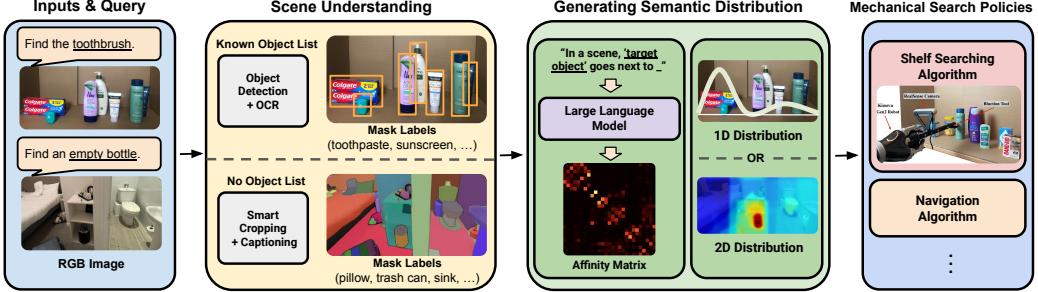


Figure 1: **Overview of Semantic Mechanical Search (SMS).** SMS accepts as input a scene image and a desired target object. It applies an object detection, or segmentation algorithm combined with captioning as necessary when object lists are unavailable. SMS then uses an LLM to compute affinities between detected objects to the target object, and it uses these affinities to output a semantic occupancy distribution which can be used for downstream mechanical search policies.

capabilities and sometimes behaves as a bag-of-words [30]. As such, CLIP is most useful for localizing objects that are already visible somewhere in the scene or map [13, 8], a property that many open-vocabulary object detectors build on [31, 32, 33]. When the target object is fully occluded, CLIP alone may not provide enough clues about potential target object locations.

LLMs demonstrate advanced reasoning and planning capabilities [34]. Many prior works [25, 35, 7] use VLMs and LLMs as end-to-end planners for both perception and planning. While such paradigms benefit from the semantic reasoning abilities of LLMs, they do not handle additional information that cannot be easily expressed through language and may not integrate well with other domain-specific policies. For example, for mechanical search on shelves, the geometric properties of objects provide valuable cues for identifying potential target object positions, and various algorithms have been proposed for handling uncertainty and planning ahead [3, 4, 5]. Likewise, for object navigation, prior research has explored exploration and navigation strategies that are independent of semantic understanding [36, 37, 38, 39]. As such, decoupling semantic reasoning and geometric planning may allow flexible integration with task-specific modules for various downstream settings.

We propose Semantic Mechanical Search (SMS), which generates an explicit intermediate representation, a *semantic occupancy distribution*, as a plug-in semantic module for existing mechanical search algorithms. This distinguishes it from prior work where VLMs and LLMs serve as end-to-end planners doing both semantic reasoning and action planning. With the goal of adding semantic reasoning to existing search policies, we study two questions: (1) Can a semantic distribution facilitate mechanical search? (2) What is the best way to generate this semantic distribution? For the second question, we hypothesize that translating image features into language features (with VLMs) first and then extracting semantic distributions from only language features (with LLMs) can outperform VLM-only methods that most current works use. We show that, rather than burdening VLMs (e.g. CLIP) with both object detection and reasoning, decoupling these two tasks leads to better results as the LLM language feature space is better at capturing semantic relations. For the first question, we show SMS can be easily integrated with a geometric shelf searching algorithm [3] to improve performance for closed-world environments such as pharmacy shelves with known object lists. In closed-world settings where object lists are available, SMS uses an open vocabulary object detection model [40] refined with Optical Character Recognition (OCR) to identify objects. In open-world settings where object lists are unavailable, SMS combines segmentation [41] and image captioning [42] to generate object mask descriptions.

This paper makes three contributions:

1. SMS, a novel framework that uses pretrained VLMs and LLMs to synthesize semantic occupancy distributions that can be easily integrated to enhance mechanical search policies;
2. A way of using LLMs to augment the reasoning capabilities of VLMs for generating better semantic distributions, with evaluations of semantic distribution quality in both closed-world and open-world settings.

- Closed-world experiments for mechanical search on shelves showing that SMS improves a geometric-based planner by 24% across the pharmacy, kitchen, and office domains in simulation and 47% in real, and a preliminary study of SMS in open-world settings.

2 Related Work and Preliminaries

2.1 Mechanical Search

Mechanical search [1, 2] refers to a broad class of robotics problems on searching for occluded and out-of-view objects via manipulation and navigation. In the former case, bin [1] and shelf environments [43, 44, 45, 46, 47] are widely studied, where intelligent estimation and manipulation planning based on possible locations of the hidden target object significantly affects the search efficiency. Many prior work uses only geometric priors [1, 4, 5, 3, 48]. A number of authors have also explored using semantic context object information [49]. Kollar and Roy [50] obtain co-occurrence statistics from web-based ontologies and Wong et al. [51] extend the approaches to occluded target objects. Kurenkov et al. [2] propose a hierarchical model to integrate semantic and geometric information and learn in simulation. However, they manually craft semantic categories, which are also sparse and can not accurately and scalably reflect real-world distributions. Instead, we harness large pretrained models to extract open-vocabulary semantic information zero-shot.

There are many types of navigation tasks, such as point goals [52, 53, 54], image goals [55, 56], and object goals [57, 58]. Finding out-of-view objects is an object goal navigation task, and the problem is also known as active visual search [59, 60]. Classical geometry-based methods typically first build a map [61, 62] and then perform planning [63, 64]. Learning-based methods typically use reinforcement learning trained in simulation [65, 57, 66, 53, 67, 68, 69, 70, 71], through YouTube videos [58], or by querying the Internet [72] to learn semantics and efficient exploration strategies. Recently, many works have explored using LLMs and VLMs out-of-the-box for semantic scene understanding [73] and zero-shot object navigation, which this work belongs to. The most common strategy is to use CLIP features [27] obtained from pretrained open-vocab detectors [74, 75] as in VL-Maps [7] and OpenScene [76] or from region proposals models as in CLIP-Fields [77], ConceptFusion [78], and NLMaps-SayCan [79] and fuse them into 3D point clouds or implicit representations [30]. The constructed representations can then be used for open-vocabulary target queries to locate the object and perform navigation. Gadre et al. [8] propose a family of methods to adapt CLIP and open-vocabulary models to localize target objects. Through a systematic comparison, they find OWL-ViT detector [80] works best, followed by patchifying images to obtain separate CLIP embeddings and compute similarity with text embeddings. In this work, instead of using the similarity of CLIP embeddings to construct relevancy maps [81], we use the LLM feature space to explicitly reason about the object’s semantic relationships and show that SMS outperforms these two methods.

2.2 Natural Language in Robotics

Grounding natural language instructions is a widely-studied problem in robot navigation [82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93], human-robot interaction [94, 95], and is increasingly studied in the manipulation literature [96, 97, 98, 99]. While classical methods commonly rely on semantic parsing and factor graphs [87, 91, 94], end-to-end learning and leveraging pretrained models are now the most popular paradigms thanks to advances in deep learning and LLMs. Examples include language-conditioned imitation learning [100, 19, 16, 18, 20, 101], language-conditioned reinforcement learning [102, 103, 14], and online correction of robot policies through language feedback [104, 105]. In particular, pretrained image encoders and open-vocabulary object detectors have enabled generalization to novel object queries at test time [25, 24, 13]. In this work, we also take in novel object targets specified using natural language, but since the target objects are not visible in the scene, the robot instead needs to detect and localize other objects and reason about their relationships. This is particularly challenging in an open-world environment when the set of possible objects is unknown, making object detectors significantly less accurate. Our method shares similarity with HOLM [6], which uses an LLM to hallucinate nearby objects in partially observable scenes based on semantics computed from affinity scores. However, it relies on an object list and only considers camera adjustment actions in simulation. We relax this assumption of accessing object

lists [7, 76, 6], propose a pipeline for generating object labels without access to any object lists, and generate semantic distributions for open-world environments.

Many studies have also used LLMs as a planner by letting them break down tasks through step-by-step reasoning [22, 23, 24, 25, 35, 106] or directly write code [21, 26]. While these end-to-end planning paradigms benefit from the deep reasoning abilities of LLMs, it's not straightforward to incorporate additional non-language information and integration with domain-specific policies. The latter is particularly valuable when the task is more complex and a flexible generalist LLM can benefit from specialized searching and planning algorithms developed by the robotics community. We propose decoupling semantic reasoning and geometric planning; rather than directly output primitive instructions from image observations, SMS uses LLMs' semantic reasoning from its feature space into a semantic distribution that specialized planning and manipulation policies can use.

2.3 Occupancy Distribution

An occupancy distribution indicates the probability of each pixel in the image containing the target object's amodal segmentation mask [1]. Prior works [1, 3, 4, 5] have utilized geometric information to generate spatial occupancy distributions by considering object geometries and camera perspective (e.g., tall target objects cannot be occluded by short objects and objects in the center of an image occlude more areas) to facilitate the search. Huang et al. [4] propose the LAX-RAY system, which uses a neural network to predict the spatial occupancy distribution. A greedy policy called Distribution Area Reduction (DAR) uses this distribution to greedily reduce the overlap between objects and the distribution the most. SMS generates the occupancy distribution using semantic information, which can then be combined with the LAX-RAY spatial distribution for downstream search.

3 Problem Statement

We consider a partially observable environment that contains a target \mathcal{O}_T and N other objects $\{\mathcal{O}_1, \dots, \mathcal{O}_N\}$. We assume the scenes in the environment are semantically organized, meaning that the starting state of the environment is sampled proportionally to their approximate likelihood of occurrence in the real world. With this assumption of semantically organized scenes, the target object location probability is proportional to object pair affinities. States $s_t \in \mathcal{S}$ consist of the full geometries, poses, and names of the objects in the scene at timestep t , and observations $y_t \in \mathcal{Y} = \mathcal{R}^{H \times W \times 3}$ are RGB images from a robot-mounted RGB camera at timestep t . Given the name of the target object and the observation y_t , the goal is to generate a useful dense occupancy distribution that encodes semantic affinities (with respect to the target object).

4 Semantic Mechanical Search

We propose SMS, a framework using VLMs and LLMs to create a dense semantic distribution between a scene and the target object to be used for downstream tasks. Fig. 1 visualizes the pipeline. SMS first uses VLMs to perform scene understanding by creating mask-label pairs to densely describe all image portions. It then uses an LLM to generate affinity scores between the labels and the target object. We spatially ground these affinities using the labels' corresponding masks. In this way, we densely represent the affinities between a target object and all parts of a scene using an LLM. SMS can be applied to two common situations: 1) a closed world where all objects in the scene are a subset of a known list and 2) an open world where some objects in the scene are previously unseen.

4.1 Scene Understanding

The goal of scene understanding is to generate mask-label pairs that characterize the scene.

Object Detection + OCR When an object list is available, we use an open vocabulary object detection model, specifically ViLD [40], to obtain object segmentation masks and labels from an RGB image. We also find using Keras Optical Character Recognition (OCR) [107] can improve the quality of the ViLD object detection, by increasing the mean average precision (mAP) on 100 pharmacy scene images from 2.4 to 28.9 (Table 5). Details are in the Appendix.

Crop Generation + Image Captioning When an object list is not available, many open vocabulary detectors such as ViLD cannot be used. We instead create image crops and use a VLM for crop

captioning, specifically BLIP-2 [42], to convert object crops to their text descriptions. We ask for the dominant objects in each crop for less noisy captions. We generate crops that are both object-centric (using Segment-Anything (SAM) [41]) for better object boundaries in the semantic distribution and general multiscale, overlapping crops that help encode large-scale semantic information.

4.2 Creating the Semantic Distribution

We consider two ways to use a language model to generate affinity scores for the semantic distribution. **(1) SMS-LLM:** We iterate over all the mask-label pairs and query the LLM with a specific prompt: “*I see the following in a room: {label}. This is likely to be the closest object to {target object}*”. This prompt directly represents the probability of the target object given we see the label. Since object labels are contained within the prompt, we do not need to normalize to account for the prior. A similar prompt with the label and the target object switched would also provide affinity scores between objects but would then have to be normalized to account for that object’s prior. The affinity score for the target object with each label is the completion probability for the tokens that represent the target object. We generate a semantic distribution from these affinity scores and detected objects. The semantic distribution models the probability of the target object occupying each location, which we approximate to be proportional to the affinity score between the target and the object closest to that location. To account for noise, we apply spatial smoothing using a Gaussian kernel with std σ . More details are in the Appendix. **(2) SMS-E:** An alternative method we explore is to use a language embedding model (e.g. OpenAI Embedding Model [108]) to get embeddings for all labels and the target object, and obtain an affinity score between each label and the target object through the dot product between these vectors.

When there are no object lists, the Crop Generation + Image Captioning pipeline described in Section 4.1 can contain many incorrect or hallucinated labels, making the distribution noisy. To mitigate this, we use CLIP to verify the captions and not for any semantic reasoning. Specifically, we compute the CLIP dot products between the image crops and the generated labels and weight the affinity scores by these relevance scores. To produce the final semantic distribution, each pixel receives the average of the weighted affinity scores of all the masks it belongs to. We find that averaging across multiple overlapping masks also helps reduce noises in the absence of object lists.

4.3 Combining with Mechanical Search Policies

Closed-World Environments We consider semantically organized shelves with objects from a known list. For mechanical search on shelves, the robot needs to manipulate objects in the shelves to reveal the occluded target object using pushing and pick-and-place actions. The goal is to minimize the number of actions taken to reveal the target object. Additional details are in the appendix. We use SMS as a plug-in semantic module for an existing search algorithm, LAX-RAY [4], by multiplying the semantic occupancy distribution with a learned spatial distribution that LAX-RAY generates based on geometry. We then use the DAR policy [4] to perform mechanical search. Since the search in cluttered environments requires manipulating other objects, once the search begins, the shelf may become semantically disorganized. As such, at each step in a rollout, SMS computes the semantic distribution using the object locations where each object was first discovered.

Open-World Environments We consider large room spaces, with semantic diversity (rooms of an office, home, aisles in a grocery store, etc.). We do not perform any manipulation in this setting and explore a downstream heuristic navigation policy that terminates when the object is within view. Given a starting position, the policy moves a fixed distance towards the highest affinity region in the image. Afterward, it takes four new images by rotating in place. We first select the desired view direction amongst the four by choosing the one that has the highest 90-percentile affinity score to ensure we are more robust to outlier affinities that may result from not having an object list. Then, after selecting the view, we again select the highest affinity point and move to that location.

5 Experiments

We investigate two questions: with a given downstream search policy, (1) can a semantic distribution improve search performance? and (2) what is the best way to generate a semantic distribution?

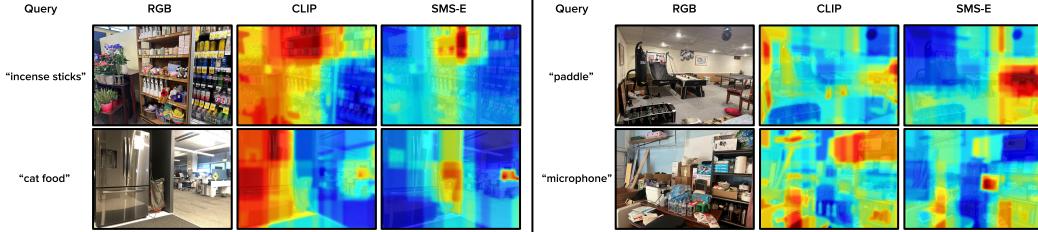


Figure 2: **Generating semantic distributions in open-world environments.** Four examples from the evaluation dataset with the 2D probability distributions generated for SMS-E and CLIP. These heatmaps are red for high-probability regions of finding the target object and blue for low probability. **Top Left:** An example of a grocery store, where the target object is “incense sticks.” CLIP highlights both near the candles and near the flowers as they are somewhat visually similar to sticks, while SMS-E only highlights the candles. **Bottom Left:** An example of an office kitchen, where the target object is “cat food.” CLIP gets distracted by the refrigerator and only slightly highlights the cat sign. **Top Right:** An example of a house, where the target object is “paddle.” CLIP incorrectly highlights the wooden panels along the walls, while SMS-E highlights the ping pong table. **Bottom Right:** For the target word “microphone,” SMS-E highlights the box with the speaker but CLIP struggles as the objects are not visually similar.

5.1 Evaluation of Semantic Distributions Quality

We investigate the second question first to obtain a semantic distribution for the downstream policy. We evaluate semantic distribution generation both in closed-world and open-world environments. In close-world environments, we evaluate the affinity matrix quality where the semantic distribution is generated, for the given object list. In open-world environments, we evaluate the semantic distribution quality on a dataset of real-life scenes.

5.1.1 Closed-World Environments

Experimental Setup: With an object list, the semantic distributions are directly generated from semantic affinity matrices, with rows and columns as objects from the list and the entries representing the affinities scores between objects calculated in Section 4.2. We use an object list of 27 objects in the pharmacy domain (list in Appendix). We directly compare the affinity matrix quality. We approximate a ground truth affinity matrix with Google Taxonomy as in Section C.1. The Google taxonomy provides semantic information for evaluation purposes to avoid human bias. Since it has limited categories, it cannot be used directly for objects that do not appear in the taxonomy.

Results: We compare the quality of the affinity matrix for the given object list generated by SMS and a CLIP-based baseline proposed by CoW [8], which uses the dot products of CLIP text embedding as the affinity scores. We compute the reduction of Jensen-Shannon Distance (JSD) [109] between the generated affinity matrix and the ground truth affinity matrix compared to the JSD between a uniform matrix and the ground truth. This quantifies the benefit SMS provides over a uniform distribution. From Table 1, we see that SMS significantly outperforms the CLIP-based method, while the SMS-LLM variant slightly outperforms the SMS-E. This suggests the reasoning capability of LLM models is more valuable for capturing semantics than CLIP embeddings. We compare other methods in Table 6 in the Appendix.

Metric	CLIP	SMS-E	SMS-LLM
$\Delta\% \text{ JSD} \uparrow$	20.0%	33.8%	44.6%

Table 1: **Closed-world semantics evaluation.** Percentage improvement of the generated semantic distributions in the pharmacy domain compared to a uniform prior, measured based on the Jensen-Shannon Distance (JSD) from the ground truth distribution.

Method	OWL-ViT	CLIP	SMS-LLM	SMS-E
IoU	0.138 ± 0.031	0.221 ± 0.034	0.345 ± 0.039	0.391 ± 0.039

Table 2: **Open-world semantics evaluation.** IoU results for different methods. The object detector-based method, OWL-ViT, performs poorly because even though the target objects are semantically related, many have very little visual similarity. CLIP performs worse than SMS because SMS is getting semantic similarity in a language-only latent space which can capture more nuance than the visual-language embedding space.

5.1.2 Open-World Environments

Experimental Setup: For the open-world environment, we evaluate the semantic distribution generation on a static image dataset consisting of 30 real scenes taken from 4 houses, 4 office buildings, and 3 local grocery stores. We sampled 90 objects across the three domains and chose those scenes

based on our accessibility to those places. In all scenes, the objects’ numbers and placements are set by their management. All scenes and the target object list are in Appendix F. Since these scenes are large, we are interested in quantifying the accuracy of the semantic distribution along both the x - and y -axes. We annotate the ground truth search area based on the real scene and use Intersection over Union (IoU) to quantitatively evaluate the accuracy of each method.

Results: We evaluate the following VLM-only baselines: CLIP and OWL-ViT, the two best-performing methods found by Gadre et al. [8]. For CLIP, it uses the same crop-label pairs as SMS to generate a semantic distribution as described in Section 5.1 but with further augmentations (jittering and horizontal flipping) on those crops for better performance [73]. We threshold this distribution and create a mask to calculate IoU with the ground truth. OWL-ViT gives bounding boxes for its labels and we directly use them to calculate the IoU. We find that OWL-ViT performs better if the best bounding box is selected rather than weighting all bounding boxes by their score and thresholding that distribution. Table 2 shows the results. SMS generates semantic distributions within 35 to 45 seconds. We see that SMS outperforms the VLM-only methods including both CLIP and OWL-ViT. More examples are in the Appendix. We hypothesize that this is because CLIP focuses more on the visual appearance of the objects rather than semantic relations. This would be less of a problem for searching visible objects but is not ideal for searching objects that are outside the field of view or occluded. For example, CLIP would associate incense sticks with sticks used for gardening while LLMs would associate the incense sticks with the candles. In addition, CLIP has a “bag of words” behaviour [30], causing it to incorrectly relate “cat food” with a fridge instead of a cat sign. In contrast, LLMs have better semantic reasoning as shown in Figure 10, where “cat food” highlights the cat sign as the highest region but also highlights the gray bag because cat food could be occluded inside of a bag. Since LLMs are trained on large corpora of human language, we hypothesize that they effectively encode the semantics of both common and rare objects and are also capable of semantic reasoning (e.g. cat food can be inside the bag) beyond just creating class categories and thus are better suited for searching fully-occluded objects. SMS-E slightly outperforms SMS-LLM as they are both bottlenecked by the quality of labels from BLIP-2.

We also conduct an ablation study for each module of SMS on semantic distribution quality with results in Table 11 in the Appendix, indicating the effectiveness of cropping with SAM and CLIP weighting, and the impact of image captioning model choice.

5.2 Semantic Distribution Effect on Mechanical Search Performance

Given a semantic distribution, we investigate the first question by conducting simulation and real experiments in close-world environments to evaluate search performance improvement brought by the semantic distribution. We combine the semantic distribution with an existing mechanical search policy LAX-RAY as in Section 4.3.

Experimental Setup: For simulation, we consider a pharmacy, a kitchen, and an office domain. In addition to the 27 objects for the pharmacy domain, we consider 24 and 40 representative objects for the kitchen and office domains from the Google Product Taxonomy [110]. We generate semantically organized scenes with the procedure detailed in Appendix E.1 and examples in Figure 7. The simulation and real experiments take place within a $0.8\text{ m} \times 0.35\text{ m} \times 0.57\text{ m}$ shelf environment.

5.2.1 Simulation Experiments

We run extensive experiments using the First Order Shelf Simulator (FOSS) from [3]. In simulation experiments, we assume perfect object detection but consider geometry for occlusion. For each domain, we generate semantically organized scenes (details in Appendix E.1) with various numbers of objects $N=12, 15, 18, 21$ with 200 scenes for each. Termination occurs when the target object becomes visible or reaches maximum action number $2N$.

For each scene, we evaluate whether SMS improves the performance of LAX-RAY [3], which only uses geometric models. We consider both SMS-E and SMS-LLM for augmenting the geometric distribution from LAX-RAY. We report two metrics: **Success rates:** The ratio of trials where the target object is found within the maximum action limit to the total number of trials. **Number of actions:** The mean and standard error of the number of actions required to reveal the target object.

	Pharmacy Domain			Kitchen Domain			Office Domain		
	Successes	# Actions	$\Delta\%$	Successes	# Actions	$\Delta\%$	Successes	# Actions	$\Delta\%$
LAX-RAY	576/741	5.56 ± 0.20	N/A	703/770	3.32 ± 0.14	N/A	575/753	4.14 ± 0.19	N/A
SMS-E	591/741	4.18 ± 0.17	24.8	725/770	2.43 ± 0.10	26.8	580/753	4.10 ± 0.18	0.9
SMS-LLM	606/741	3.76 ± 0.14	32.4	710/770	2.42 ± 0.10	27.1	598/753	3.63 ± 0.16	12.3

Table 3: Simulation experiment results for three domains averaged over 12, 15, 18, 21 number of objects, also reported with $\Delta\%$, the percentage reduction in the number of actions compared to LAX-RAY.

We report results for all numbers of objects N in the Appendix and the results averaged across all values of N in Table 3. In all domains, SMS-LLM and SMS-E improve LAX-RAY performances with higher success rates and fewer search actions. In the pharmacy and office domain, SMS-LLM outperforms SMS-E, while in the kitchen domain, they perform comparably. For the office experiments, the performance improvement is relatively small. We hypothesize that this is due to a majority of the office environment consisting of generic office supplies that do not have a clear semantic categorization, making semantic prior less effective. Overall, the results suggest that SMS-LLM can serve as a semantic plug-in module and improve LAX-RAY performance in semantically arranged environments by 32.4%, 27.1%, and 12.3% in the pharmacy, kitchen, and office domains respectively while improving success rates. SMS-LLM outperforms SMS-E, indicating the quality of the affinity matrix is directly correlated with the task performance.

We also show a strong positive correlation between object detection accuracy and task performance with results and details in Appendix E.3, indicating the benefits of SMS using OCR. In addition, we show SMS are effective on different downstream policies by using SMS as the plug-in module for Distribution Entropy Reduction (DER) from [3]. The details are in Appendix E.3.1.

5.2.2 Physical Experiments

Method	# Actions	$\Delta\%$	Method	# Actions	$\Delta\%$
LAX-RAY	4.25 ± 0.64	N/A	SMS-LLM	2.25 ± 0.46	47.1

Table 4: Physical experiment results (12 trials each). We report the average number of actions taken to reveal the target object as well as the percentage reduction in the number of actions over the spatial neural network.

We conduct experiments on a physical pharmacy shelf. We use the Kinova Gen2 robot with a 3D-printed blade and suction tool [4] (see Figure 1). An Intel RealSense depth camera mounted on the tool provides RGBD observations. We use 3 scenes each of $N = 7, 8, 9$, and 10 objects for a total of 12 scenes and a threshold visibility of 50% for determining success. More details are in the Appendix. As simulation results from Table 3 suggest SMS-LLM outperforms SMS-E, we evaluate SMS-LLM in physical experiments. An identical set of 12 semantically arranged scenes (starting configurations) is used for each method.

Results are shown in Table 4. We observe that SMS significantly accelerates mechanical search on shelves, reducing the average number of actions by 47.1%. In physical experiments, the noises in the depth images result in worse spatial distribution than in simulation, making the semantic distribution more critical in identifying where a target object may lie. We also conduct a preliminary navigation experiment in open-world environments as in Section 4.3 with details in Appendix E.5.

6 Limitations and Future Work

We present Semantic Mechanical Search (SMS), an algorithm for semantic distribution generation for a fully-occluded target object. SMS facilitates mechanical search in closed-world environments and improves semantic distribution quality for open-world environments. SMS has several limitations, which open up possibilities for future work: (1) We only evaluate one open-world navigation task with a heuristic navigation policy without large-scale evaluations of other open-world environments; (2) While SMS, which operates in the LLM feature space, generates better semantic distributions than CLIP-based method, we have not compared to other VLMs such as GPT-4 [111] or LLaVa [112] due to their inaccessibility to obtain affinity scores. VLMs with strong reasoning abilities, such as GPT-4V [113], have the potential to directly generate high-quality semantic distributions. Further applying GPT-4V in object search would be an interesting future direction. We conduct an initial exploration in Section A. (3) SMS for closed-world relies on creating an offline affinity matrix which can take a few minutes with large object lists, while SMS for open-world takes

35 to 45 seconds for each semantic distribution (4) The performance of SMS is sensitive to the quality of each module in the framework.

Acknowledgments

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab. The authors are supported in part by donations from Toyota Research Institute, Bosch, Google, Siemens, and Autodesk. L. Y. Chen is supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE 2146752. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors. We thank our colleagues who helped with the project and provided helpful feedback and suggestions, in particular Chung Min Kim and Alishba Imran.

References

- [1] 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 *2019 International Conference on Robotics and Automation (ICRA)*, 2019.
- [2] A. Kurenkov, R. Martín-Martín, J. Ichnowski, K. Goldberg, and S. Savarese. Semantic and geometric modeling with neural message passing in 3d scene graphs for hierarchical mechanical search. *International Conference on Robotics and Automation (ICRA)*, 2021.
- [3] H. Huang, M. Dominguez-Kuhne, V. Satish, M. Danielczuk, K. Sanders, J. Ichnowski, A. Lee, A. Angelova, V. Vanhoucke, and K. Goldberg. Mechanical search on shelves using lateral access x-ray. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2045–2052, 2021.
- [4] H. Huang, M. Danielczuk, C. M. Kim, L. Fu, Z. Tam, J. Ichnowski, A. Angelova, B. Ichter, and K. Goldberg. Mechanical search on shelves using a novel “blunction” tool. *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.
- [5] H. Huang, L. Fu, M. Danielczuk, C. M. Kim, Z. Tam, J. Ichnowski, A. Angelova, B. Ichter, and K. Goldberg. Mechanical search on shelves with efficient stacking and destacking of objects. *International Symposium on Robotics Research (ISRR)*, 2022.
- [6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. Holm: Hallucinating objects with language models for referring expression recognition in partially-observed scenes. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.
- [7] C. Huang, O. Mees, A. Zeng, and W. Burgard. Visual language maps for robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, London, UK, 2023.
- [8] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song. Cows on pasture: Baselines and benchmarks for language-driven zero-shot object navigation, 2022.
- [9] S. Parisi, A. Rajeswaran, S. Purushwalkam, and A. Gupta. The unsurprising effectiveness of pre-trained vision models for control. In *International Conference on Machine Learning*, pages 17359–17371. PMLR, 2022.
- [10] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

- [11] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell. Real-world robot learning with masked visual pre-training. In *Conference on Robot Learning*, pages 416–426. PMLR, 2023.
- [12] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.
- [13] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, B. Zitkovich, F. Xia, C. Finn, et al. Open-world object manipulation using pre-trained vision-language models. *arXiv preprint arXiv:2303.00905*, 2023.
- [14] S. Nair, E. Mitchell, K. Chen, B. Ichter, S. Savarese, and C. Finn. Learning language-conditioned robot behavior from offline data and crowd-sourced annotation. In *Conference on Robot Learning (CoRL)*, 2021.
- [15] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.
- [16] M. Shridhar, L. Manuelli, and D. Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. *Conference on Robot Learning (CoRL)*, 2022.
- [17] C. Lynch and P. Sermanet. Language conditioned imitation learning over unstructured data. *arXiv preprint arXiv:2005.07648*, 2020.
- [18] C. Lynch and P. Sermanet. Language conditioned imitation learning over unstructured data. *Robotics: Science and Systems (RSS)*, 2021.
- [19] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. K. Baruch, T. Armstrong, and P. R. Florence. Interactive language: Talking to robots in real time. *ArXiv preprint arXiv:2210.06407*, 2022.
- [20] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan. Vima: General robot manipulation with multimodal prompts. *arXiv preprint arXiv:2210.03094*, 2022.
- [21] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. R. Florence, and A. Zeng. Code as policies: Language model programs for embodied control. *ArXiv preprint arXiv:2209.07753*, 2022.
- [22] B. I. et al. Do as i can, not as i say: Grounding language in robotic affordances. In *6th Annual Conference on Robot Learning*, 2022.
- [23] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022.
- [24] A. Zeng, A. Wong, S. Welker, K. Choromanski, F. Tombari, A. Purohit, M. Ryoo, V. Sindhwani, J. Lee, V. Vanhoucke, et al. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022.
- [25] W. H. et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.
- [26] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg. Progprompt: Generating situated robot task plans using large language models. *arXiv preprint arXiv:2209.11302*, 2022.

- [27] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.
- [28] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler. Open-vocabulary queryable scene representations for real world planning, 2023.
- [29] D. Shah, B. Osiński, brian ichter, and S. Levine. LM-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *6th Annual Conference on Robot Learning*, 2022. URL <https://openreview.net/forum?id=UW5A3SweAH>.
- [30] J. Kerr, C. M. Kim, K. Goldberg, A. Kanazawa, and M. Tancik. Lerp: Language embedded radiance fields. *arXiv preprint arXiv:2303.09553*, 2023.
- [31] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui. Open-vocabulary object detection via vision and language knowledge distillation, 2022.
- [32] F. Liang, B. Wu, X. Dai, K. Li, Y. Zhao, H. Zhang, P. Zhang, P. Vajda, and D. Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip, 2023.
- [33] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra. Detecting twenty-thousand classes using image-level supervision, 2022.
- [34] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. Chi, Q. Le, and D. Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.
- [35] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence. Palm-e: An embodied multimodal language model. In *arXiv preprint arXiv:2303.03378*, 2023.
- [36] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, pages 146–151. IEEE, 1997.
- [37] J. Maja. Integration of representation into goal-driven behavior-based robots. *IEEE transactions on robotics and automation*, 8(3):304–312, 1992.
- [38] S. Thrun and A. Büken. Integrating grid-based and topological maps for mobile robot navigation. In *Proceedings of the national conference on artificial intelligence*, pages 944–951, 1996.
- [39] B. Yamauchi and R. Beer. Spatial learning for navigation in dynamic environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(3):496–505, 1996.
- [40] X. Gu, T.-Y. Lin, W. Kuo, and Y. Cui. Open-vocabulary object detection via vision and language knowledge distillation. *International Conference on Learning Representations (ICLR)*, 2021.
- [41] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv:2304.02643*, 2023.
- [42] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv*, abs/2301.12597, 2023.
- [43] M. Gupta, T. Rühr, M. Beetz, and G. S. Sukhatme. Interactive environment exploration in clutter. pages 5265–5272, 2013.

- [44] M. R. Dogar, M. C. Koval, A. Tallavajhula, and S. S. Srinivasa. Object search by manipulation. *Autonomous Robots*, 36(1):153–167, 2014.
- [45] J. K. Li, D. Hsu, and W. S. Lee. Act to see and see to act: Pomdp planning for objects search in clutter. pages 5701–5707, 2016.
- [46] Y. Xiao, S. Katt, A. ten Pas, S. Chen, and C. Amato. Online planning for target object search in clutter under partial observability. pages 8241–8247, 2019.
- [47] W. Bejjani, W. C. Agboh, M. R. Dogar, and M. Leonetti. Occlusion-aware search for object retrieval in clutter. pages 4678–4685, 2021.
- [48] L. Y. Chen, H. Huang, M. Danielczuk, J. Ichnowski, and K. Goldberg. Optimal shelf arrangement to minimize robot retrieval time. *IEEE International Conference on Automation Science and Engineering (CASE)*, 2022.
- [49] L. E. Wixson and D. H. Ballard. Using intermediate objects to improve the efficiency of visual search. *International Journal of Computer Vision*, 12(2-3):209–230, 1994.
- [50] T. Kollar and N. Roy. Utilizing object-object and object-scene context when planning to find things. In *2009 IEEE International Conference on Robotics and Automation*, pages 2168–2173. IEEE, 2009.
- [51] L. L. Wong, L. P. Kaelbling, and T. Lozano-Pérez. Manipulation-based active search for occluded objects. In *2013 IEEE International Conference on Robotics and Automation*, pages 2814–2819. IEEE, 2013.
- [52] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020.
- [53] P. Chattopadhyay, J. Hoffman, R. Mottaghi, and A. Kembhavi. Robustnav: Towards benchmarking robustness in embodied navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15700, 2021.
- [54] D. Gordon, A. Kadian, D. Parikh, J. Hoffman, and D. Batra. Splitnet: Sim2sim and task2task transfer for embodied visual navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1022–1031, 2019.
- [55] L. Mezghan, S. Sukhbaatar, T. Lavril, O. Maksymets, D. Batra, P. Bojanowski, and K. Alahari. Memory-augmented reinforcement learning for image-goal navigation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3316–3323. IEEE, 2022.
- [56] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017.
- [57] Z. Al-Halah, S. K. Ramakrishnan, and K. Grauman. Zero experience required: Plug & play modular transfer learning for semantic visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17031–17041, 2022.
- [58] M. Chang, A. Gupta, and S. Gupta. Semantic visual navigation by watching youtube videos. *Advances in Neural Information Processing Systems*, 33:4283–4294, 2020.
- [59] J. K. Tsotsos. On the relative complexity of active vs. passive visual search. *International journal of computer vision*, 7(2):127–141, 1992.
- [60] A. Aydemir, A. Pronobis, M. Göbelbecker, and P. Jensfelt. Active visual object search in unknown environments using uncertain semantics. *IEEE Transactions on Robotics*, 29(4):986–1002, 2013.

- [61] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
- [62] H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mobile robot navigation and mapping. *The International Journal of Robotics Research*, 18(7):650–668, 1999.
- [63] B. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and autonomous systems*, 8(1-2):47–63, 1991.
- [64] B. H. Wilcox. Robotic vehicles for planetary exploration. *Applied Intelligence*, 2:181–193, 1992.
- [65] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018.
- [66] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33:4247–4258, 2020.
- [67] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.
- [68] Y. Liang, B. Chen, and S. Song. Sscnav: Confidence-aware semantic scene completion for visual semantic navigation. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 13194–13200. IEEE, 2021.
- [69] S. Wani, S. Patel, U. Jain, A. Chang, and M. Savva. Multion: Benchmarking semantic map memory using multi-object navigation. *Advances in Neural Information Processing Systems*, 33:9700–9712, 2020.
- [70] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6750–6759, 2019.
- [71] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018.
- [72] M. Samadi, T. Kollar, and M. Veloso. Using the web to interactively learn to find objects. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 2074–2080, 2012.
- [73] H. Ha and S. Song. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. In *6th Annual Conference on Robot Learning*, 2022.
- [74] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin. Open-vocabulary image segmentation. *arXiv preprint arXiv:2112.12143*, 2021.
- [75] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022.
- [76] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. *arXiv preprint arXiv:2211.15654*, 2022.
- [77] N. M. M. Shafiullah, C. Paxton, L. Pinto, S. Chintala, and A. Szlam. Clip-fields: Weakly supervised semantic fields for robotic memory. *arXiv preprint arXiv:2210.05663*, 2022.

- [78] K. M. Jatavallabhula, A. Kuwajerwala, Q. Gu, M. Osama, T. Chen, S. Li, G. Iyer, S. Saryazdi, N. Keetha, A. Tewari, et al. Conceptfusion: Open-set multimodal 3d mapping. *arXiv preprint arXiv:2302.07241*, 2023.
- [79] B. Chen, F. Xia, B. Ichter, K. Rao, K. Gopalakrishnan, M. S. Ryoo, A. Stone, and D. Kappler. Open-vocabulary queryable scene representations for real world planning. *arXiv preprint arXiv:2209.09874*, 2022.
- [80] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, et al. Simple open-vocabulary object detection with vision transformers. *arXiv preprint arXiv:2205.06230*, 2022.
- [81] H. Chefer, S. Gur, and L. Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 397–406, October 2021.
- [82] D. Chen and R. Mooney. Learning to interpret natural language navigation instructions from observations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 859–865, 2011.
- [83] F. Duvallet, T. Kollar, and A. Stentz. Imitation learning for natural language direction following through unknown environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 1047–1053. IEEE, 2013.
- [84] F. Duvallet, M. R. Walter, T. Howard, S. Hemachandra, J. Oh, S. Teller, N. Roy, and A. Stentz. Inferring maps and behaviors from natural language instructions. In *Experimental Robotics: The 14th International Symposium on Experimental Robotics*, pages 373–388. Springer, 2016.
- [85] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell. Speaker-follower models for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 31, 2018.
- [86] S. Hemachandra, F. Duvallet, T. M. Howard, N. Roy, A. Stentz, and M. R. Walter. Learning models for following natural language directions in unknown environments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5608–5615. IEEE, 2015.
- [87] T. M. Howard, S. Tellex, and N. Roy. A natural language planner interface for mobile manipulators. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6652–6659. IEEE, 2014.
- [88] T. Kollar, S. Tellex, D. Roy, and N. Roy. Toward understanding natural language directions. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 259–266. IEEE, 2010.
- [89] M. MacMahon, B. Stankiewicz, and B. Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. *Def*, 2(6):4, 2006.
- [90] C. Matuszek, D. Fox, and K. Koscher. Following directions using statistical machine translation. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 251–258. IEEE, 2010.
- [91] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to parse natural language commands to a robot control system. In *Experimental robotics: the 13th international symposium on experimental robotics*, pages 403–415. Springer, 2013.
- [92] H. Mei, M. Bansal, and M. Walter. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

- [93] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 1507–1514, 2011.
- [94] J. Thomason, S. Zhang, R. J. Mooney, and P. Stone. Learning to interpret natural language commands through human-robot dialog. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.
- [95] M. Shridhar and D. Hsu. Interactive visual grounding of referring expressions for human-robot interaction. *arXiv preprint arXiv:1806.03831*, 2018.
- [96] D. K. Misra, J. Sung, K. Lee, and A. Saxena. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. *The International Journal of Robotics Research*, 35(1-3):281–300, 2016.
- [97] R. Paul, J. Arkin, D. Aksaray, N. Roy, and T. M. Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot platforms. *The International Journal of Robotics Research*, 37(10):1269–1299, 2018.
- [98] S. Patki, E. Fahnestock, T. M. Howard, and M. R. Walter. Language-guided semantic mapping and mobile manipulation in partially observable environments. In *Conference on Robot Learning*, pages 1201–1210. PMLR, 2020.
- [99] O. Mees and W. Burgard. Composing pick-and-place tasks by grounding language, 2021.
- [100] M. Shridhar, L. Manuelli, and D. Fox. Cliport: What and where pathways for robotic manipulation. *Conference on Robot Learning (CoRL)*, 2021.
- [101] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.
- [102] D. K. Misra, J. Langford, and Y. Artzi. Mapping instructions and visual observations to actions with reinforcement learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.
- [103] Y. Jiang, S. S. Gu, K. P. Murphy, and C. Finn. Language as an abstraction for hierarchical deep reinforcement learning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.
- [104] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox. Correcting robot plans with natural language feedback. *Robotics: Science and Systems (RSS)*, 2022.
- [105] Y. Cui, S. Karamcheti, R. Palletti, N. Shivakumar, P. Liang, and D. Sadigh. No, to the right: Online language corrections for robotic manipulation via shared autonomy. *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2023.
- [106] W. Huang, F. Xia, D. Shah, D. Driess, A. Zeng, Y. Lu, P. Florence, I. Mordatch, S. Levine, K. Hausman, et al. Grounded decoding: Guiding text generation with grounded models for robot control. *arXiv preprint arXiv:2303.00855*, 2023.
- [107] Keras ocr. <https://support.google.com/merchants/answer/6324436?hl=en>.
- [108] A. N. et al. Text and code embeddings by contrastive pre-training. *arXiv preprint arXiv:2201.10005*, 2022.

- [109] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi:10.1109/18.61115.
- [110] G. M. Center. Google product category. <https://support.google.com/merchants/answer/6324436?hl=en>. Accessed: 2023-01-31.
- [111] OpenAI. Gpt-4 technical report, 2023.
- [112] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning, 2023.
- [113] OpenAI. Gpt-4v(ision) system card, 2023.
- [114] S. Karaoglu, J. Gemert, and T. Gevers. Object reading: Text recognition for object recognition. volume 7585, 10 2012. ISBN 978-3-642-33884-7. doi:10.1007/978-3-642-33885-4_46.
- [115] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv preprint arXiv:1810.04805*, 2019.
- [116] S. Z. et al. Opt: Open pre-trained transformer language models. *ArXiv preprint 2205.01068*, 2022.

Appendix

A Preliminary Comparisons to GPT-4V

With the recent development of GPT-4V, we conduct a preliminary exploration to see if VLMs with strong reasoning abilities can create an explicit semantic distribution over an image of the scene. We use the following prompt with an image of the scene to extract a location within the image that should correspond to the highest activation. Since explicitly creating heatmaps is currently nontrivial, we ask GPT-4V to identify bounding boxes as it is an easier task. We use the prompt:

In this image, where are the couple most likely places in the image I would find TARGET_OBJECT? List the places in decreasing order of likelihood and explain why this place was chosen (for example considering objects in that place). Explicitly write one bounding box (written as a tuple) per place and code with opencv2 to place the bounding boxes on the image. Fit the bounding boxes to the object. The image has a width of WIDTH pixels and a height of HEIGHT pixels.

where we substitute TARGET_OBJECT, WIDTH, HEIGHT for the target object, width of the image, and height of the image respectively. We design the prompt so the model has to explain its reasoning and write code, both of which have been shown to increase model performance [21].

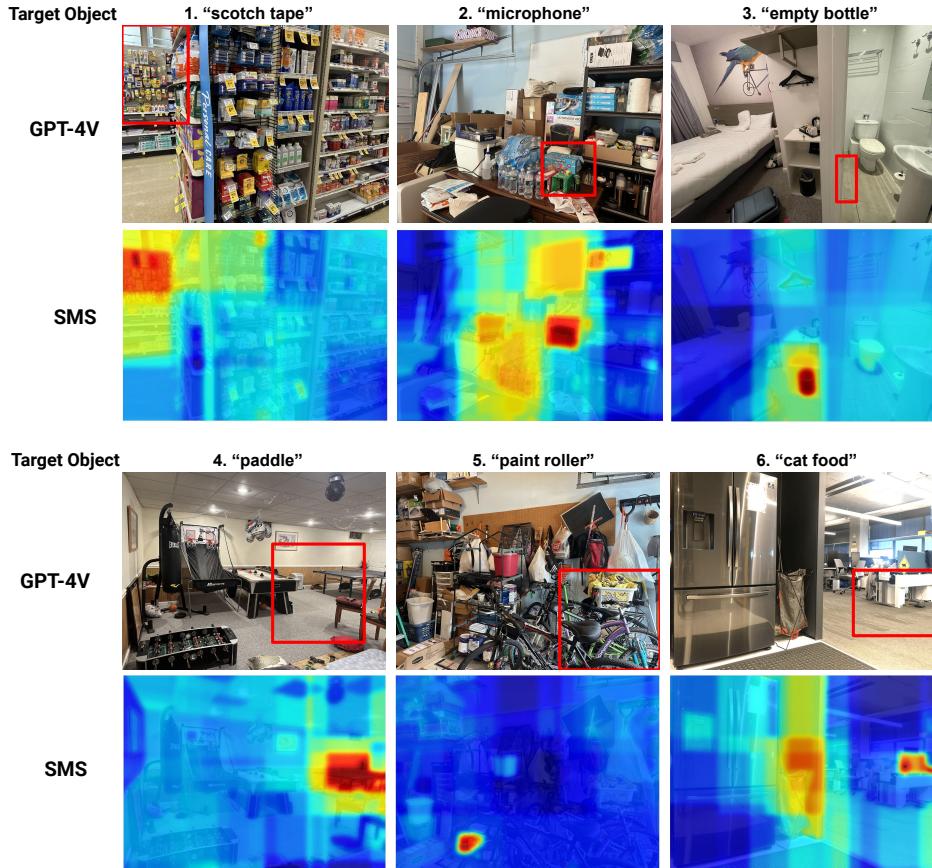


Figure 3: There are six examples, each example has the target object and the corresponding SMS semantic distribution and the GPT-4V comparison. The GPT-4V depicts a red bounding box, which is a visualization of bounding box (specified by the tuple) of the most likely place to find the target object, extracted from the corresponding response in Figure 4.

<p>1.  In the given image, considering the types of products displayed, the most likely places to find scotch tape would be in sections with office or stationary supplies. Upon examining the image, here are the potential places where scotch tape might be located, in decreasing order of likelihood:</p> <ol style="list-style-type: none"> 1. The section with the yellow and blue packets: This area seems to have a variety of small items that might be typical of an office supplies section, which is where one might expect to find scotch tape. * Bounding Box: (0, 0, 1000, 1500) 2. The section near the bottom left: Although it's a bit hard to discern the exact items, this section has some boxed items that could be stationary or office related. * Bounding Box: (0, 1800, 1000, 3024) 3. The section in the top right: This looks like it has some health and beauty related products, but occasionally, stores might place tape and other small stationary items near checkout counters or in miscellaneous sections. * Bounding Box: (2500, 0, 4032, 1000) 	<p>2.  Based on the contents of the image, the most likely places in decreasing order of likelihood where one might find a microphone are:</p> <ol style="list-style-type: none"> 1. Box labeled "PATHFINDER 280": It looks like some sort of electronics packaging, which could contain a microphone or related equipment. * Bounding box: (2200, 1800, 3000, 2600) 2. Box with visible electronics next to the "PATHFINDER 280": This seems like an electronics device packaging, and often microphones or their related accessories come in similar packages. * Bounding box: (1800, 1800, 2200, 2600) 3. Shelf with various items: There are miscellaneous items on the shelf, and there could potentially be a microphone among them. * Bounding box: (2700, 500, 4032, 2800)
<p>3.  Based on the image provided and typical behavior of humans, the most likely places in decreasing order of likelihood where one might find an empty bottle are:</p> <ol style="list-style-type: none"> 1. Trash Can: People often dispose of empty bottles in trash cans. * Bounding box: (2500, 2000, 2800, 2700) 2. Bedside Table: It's not uncommon for someone to have a drink by their bedside and leave the bottle there. * Bounding box: (0, 1500, 700, 2800) 3. Bathroom Counter: Sometimes, people leave empty bottles of toiletries or other products near the sink. * Bounding box: (3000, 500, 4032, 1800) 	<p>4.  Based on the contents of the image, the most likely places in decreasing order of likelihood where one might find a paddle are:</p> <ol style="list-style-type: none"> 1. Near the Ping Pong Table: Given the presence of the ping pong table, it's most probable to find paddles near it. * Bounding box: (2300, 1100, 3700, 2600) 2. Under the Ping Pong Table: Sometimes paddles are placed or fallen under the table. * Bounding box: (2300, 1800, 3700, 2900) 3. On the Shelf Below the Basketball Game: The shelf beneath the basketball game could potentially store equipment related to other games, including ping pong paddles. * Bounding box: (800, 1500, 1700, 2000)
<p>5.  In the provided image, the places where you might expect to find a paint roller would typically be near painting supplies or areas that seem to be set aside for maintenance tools. Based on the contents of the image, here are the potential places where a paint roller might be located, in decreasing order of likelihood:</p> <ol style="list-style-type: none"> 1. Near the paint cans: This is the most obvious location because paint rollers are commonly stored near paint cans. This is seen at the bottom right of the image. * Bounding Box: (2500, 1500, 4032, 3024) 2. On the shelf above the bicycles: This area seems to have miscellaneous items. Although it's cluttered, it might be a place where a paint roller could be kept. * Bounding Box: (0, 1000, 2000, 2000) 3. Near the storage drawers on the left: The drawers seem like a place where tools and related items might be kept. * Bounding Box: (0, 1500, 1200, 2500) 	<p>6.  Based on the image provided, the areas to find cat food would typically be near signs or objects related to cats or areas that look like storage places for pet supplies. Here are the potential places where cat food might be located, in decreasing order of likelihood:</p> <ol style="list-style-type: none"> 1. Near the "Caution: Cat Crossing" sign: The presence of a cat crossing sign indicates the possibility of a cat's presence in the vicinity. The desk or area near the sign might have cat food. * Bounding Box: (2800, 1500, 4032, 2500) 2. Refrigerator: Cat food, especially the wet type, can be stored in a fridge to preserve freshness. It's possible there could be cat food inside. * Bounding Box: (0, 1200, 2500) 3. The black bag on the left: This bag looks like a possible storage place, and it might contain cat food or other pet supplies. * Bounding Box: (800, 1500, 1300, 2500)

Figure 4: These six GPT-4V responses specify the bounding boxes that are visualized in Figure 3. The prompt used for each example is an image of the scene and what is stated in Section A. These examples show that GPT-4V is good at object identification in the image and semantic reasoning to know near which objects in the image the target object would be. However, these bounding boxes do not reliably encompass the objects that GPT-4V references in the responses, indicating questionable object localization.

We compare six examples where Figure 4 contains the responses of GPT-4V for each example and Figure 3 contains the visualization of the highest likelihood bounding box mentioned in the corresponding response and a comparison with SMS. We note that GPT-4V is very good at identifying objects in the image and semantically reasoning where the target object should be with respect to the objects it has identified in the image. Going through the examples, in example 1, it was able to correctly identify the stationary materials in the image and correspond the scotch tape to associated with that region. In example 2, it was able to perform OCR and identify the 'PATHFINDER 280' box and correctly reason that the microphone would be near that electronic packaging. However, the bounding box is not accurate as it only partially includes the 'PATHFINDER 280' box. In example 3, GPT-4V correctly identifies a trash can in the scene and that is the most likely location for an empty bottle, but fails to place an accurate bounding box around the trash can. A similar story happens in example 4 where GPT-4V identifies the ping pong table and reasons the paddle should be near the table but isn't able to place a tight bounding box around the table. In example 5, GPT-4V is able to reason the paint roller would be near the paint cans but the bounding box is predominantly encompassing the bicycles. Lastly for example 6, the bounding box primarily contains a desk and the floor rather than the cat sign.

This initial exploration suggests that GPT-4V is able to correctly reason about the objects in the image to determine what highly correlates to the target object, but GPT-4V is not able to reliably identify those regions in the image. The comparisons to SMS indicate that SMS is more reliable for creating explicit semantic distributions. Since the high performing closed-source VLMs (e.g. GPT-4V) can only be interacted through their language output, there is no current nontrivial method to extract accurate distributions from these models as the token probabilities are not available and the weights are not available to fine-tune the model for object localization. Future work could explore further prompt engineering, iterative adjustments with chain-of-thought prompting, and semantic distribution generation with diffusion.

B Scene Understanding

B.1 Object detection + OCR

Because ViLD is a general-purpose detector, it cannot easily distinguish between objects belonging to the same domain (e.g., Advil versus Ibuprofen). Because of this, we use OCR with Keras OCR[107] to improve the quality of the object detections. While OCR has been used in prior work to aid object detection [114], we use text embedding combined with OCR for better performance. For each object, we concatenate the text observed on it and compute the text embedding using OpenAI Embeddings. We compute the dot product between the embeddings of the concatenated text and every class label. We normalize this probability vector by subtracting the minimum value and then adjusting the vector with some temperature. We finally multiply this by the object detection probability vector.

Let C_i denote the class label of object \mathcal{O}_i (e.g., “Tylenol” as opposed to the broader category “medication”); I_i represent the general shape, size, and color-related features of \mathcal{O}_i ; and T_i be the detected text on \mathcal{O}_i . Recall that all objects belong to some class C_i . We calculate

$$\begin{aligned} & P(C_i | I_i, T_i) \\ &= \frac{P(I_i, T_i | C_i) \cdot P(C_i)}{P(I_i, T_i)} \\ &= \frac{P(T_i | I_i, C_i) \cdot P(I_i | C_i) \cdot P(C_i)}{P(I_i, T_i)} \\ &= \frac{P(T_i | C_i) \cdot P(I_i | C_i) \cdot P(C_i)}{P(I_i, T_i)} \\ &= \frac{P(C_i | T_i) P(T_i)}{P(C_i)} \cdot \frac{P(C_i | I_i) P(I_i)}{P(C_i)} \cdot \frac{P(C_i)}{P(I_i, T_i)} \\ &\propto P(C_i | T_i) P(C_i | I_i) \end{aligned}$$

as T_i is independent of I_i when conditioned on C_i , and $P(C_i)$ is uniform. This illustrates that the multiplication of the OCR probabilities and the object detection probabilities can give us a refined estimate of the category probabilities.

We test object detection performance on scenes generated through isolated perception experiments. We take RGB images of 100 scenes of the Pharmacy domain using a high-resolution camera and study the effect of having OCR. Results for this experiment are in Table 5. As is standard in the computer vision literature, we report mAP (mean Average Precision) averaged over intersection-over-union (IOU) thresholds from 0.50 to 0.95 with a step size of 0.05, as well as top- k classification accuracy (i.e., if the ground truth label appears in the k labels with the highest probabilities). The results show that OCR leads to a significant improvement across all metrics, with mAP improving by a factor of 12 and top-1 accuracy improving by a factor of 3.

Table 5: Object Detection Refinement Results. We study the effect of OCR and report the mean average precision (mAP) of the predicted bounding boxes and top-K accuracy of the predicted labels.

Method	mAP (\uparrow)	Top-K Accuracy % (\uparrow)		
		k=1	k=3	k=5
ViLD	2.4	14.7	32.3	41.6
ViLD + OCR	28.9	45.0	62.0	69.5

C Creating the Semantic Distribution

C.1 Affinity Matrix Generation

We compare two ways to generate affinity matrices. First, we generate the affinity matrices using large language models using the following procedure: 1) We replace both the {obj} and the {target}

object} in the prompt: "I see the following in a room: {obj}. This is likely to be the closest object to {target object}." 2) We find the log probability of the target object instead of the {obj} since it will be represented by the same number of tokens regardless of the {obj} in this prompt and use this instead for representing affinity values. Second, we use an embedding model to encode {obs} and {target object} and use the cosine similarity to find the affinity values. We normalize when appropriate. For the pharmacy domain, in Table 6, we generate affinity matrices with different LLMs and embedding models and then compare the quality of affinity matrices quantitatively by comparing them to the open-source Google Product Taxonomy [110] as the "ground truth" matrix. Visualizations of the affinity matrices generated by the ground truth, the best embedding model (OpenAI Embeddings), and the best LLM are shown in Figure 5 for the pharmacy domain. In the pharmacy domain, we have the following 6 categories and items from the taxonomy:

1. Supplements: vitamins, fish oil, omega-3, calcium, probiotics, protein powder, COQ10, anthocyanin
2. Hair Care: shampoo, conditioner
3. Oral Care: toothpaste, toothbrush, dental floss
4. Cosmetics: face wash, sunscreen, lotion, hand cream, body wash
5. Medication: aspirin, tylenol, ibuprofen, advil, pain relief
6. Outliers: shaving cream, eye drops, deodorant, band-aid

For the ground-truth matrix, all elements in a category are given uniform affinities to each other, and each row is normalized to sum to 1.0 probability. Note that each item in the "outliers" category (e.g., eye drops) does not belong to any of the other 5 categories and is treated as its own category. We use the Google taxonomy to categorize the objects within each category. With the categories listed in order along both axes of the matrix, the ground truth affinity matrix has a block-diagonal structure with a uniform block for each category (Figure 5A). We evaluate the following LLMs and embedding models off-the-shelf, without finetuning: BERT [115], CLIP [27], embeddings from the OpenAI API [108], OPT-13B [116], and PaLM. JSD measures the similarity between two probability distributions, so we measure the similarity between each row of the affinity matrix and the corresponding row of the ground truth. Then, we average across the rows to get the average distance from each object's probability distribution to that object's ground truth. We observe that the choice of LLM has a significant impact on the affinity matrix (Table 6), and that the LLMs can approximately recover the block diagonal structure of the ground truth matrix (Figure 5). PaLM attains the highest accuracy, with a 44.6% improvement over a uniform affinity matrix.

Table 6: Affinity matrix results. We report the average Jensen-Shannon Distance (JSD) between each row of the affinity matrix and the ground truth matrix, as well as the percentage improvement over the uniform JSD (i.e., (uniform JSD - method JSD) / uniform JSD).

Method	JSD (↓)	% Improvement (↑)
Uniform	0.65	N/A
BERT Embedding	0.64	1.5
CLIP Embedding	0.52	20.0
OpenAI Embedding	0.43	33.8
OPT-13B	0.38	41.5
PaLM	0.36	44.6

C.2 Offline Semantic Distribution Generation with Object List

We now generate a semantic distribution based on the affinity matrix and detected objects. The semantic occupancy distribution models the probability that the target object occupies a given location, given the classes of observed objects in the scene, i.e. $P(L_T = l \mid L_{1\dots n} = l_{1\dots n}, C_{1\dots n} = c_{1\dots n})$, where L_T is the location of the target object, $L_{1\dots n}$ are the positions of the *visible* objects, and $C_{1\dots n}$ are the inferred classes of the visible objects. We abbreviate this quantity as $P(L_T = l \mid L, C)$.

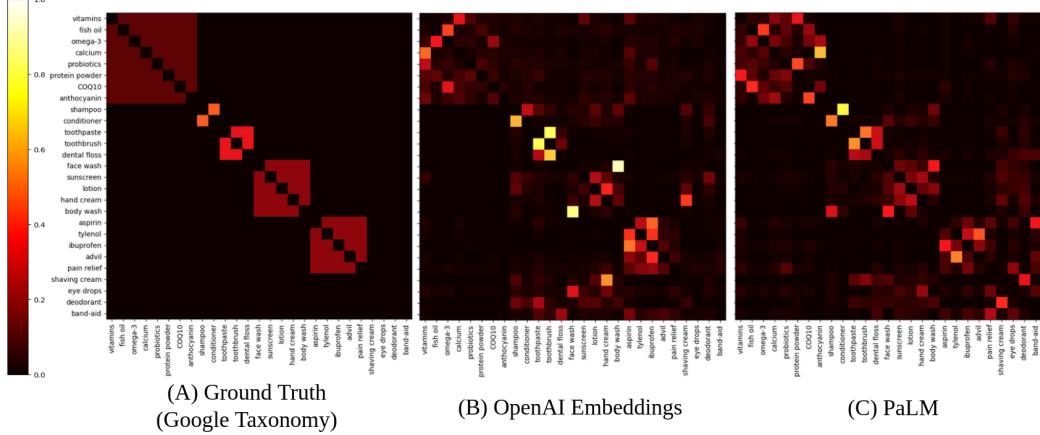


Figure 5: Three different affinity matrices for the pharmacy domain. The left shows the affinity matrix generated from the Google Taxonomy. The center shows the affinity matrix generated by the OpenAI Embeddings model and the right shows the matrix generated by PaLM.

We interpret affinity values M_{ij} to be the probability of object j being the closest to object i in expectation across scenes. However, given the current scene, there may be more or less space that is nearest to a particular object, so we interpret these affinity values as being normalized per unit area. Thus, formally, given $N(l)$ representing the index of the object closest to location $l = (x_l, y_l)$, $P(L_T = l | L, C) \propto M_{target, N(l)}$.

In simulation experiments for constrained environments, $N(\cdot)$ is computed using the 3D coordinates of the visible objects obtained from the depth image. We compute the 2D semantic occupancy distribution (in the horizontal plane of the shelf) and reduce it to 1D by summing along camera rays. In physical experiments, to avoid errors due to noisy depth readings we compute the distribution directly in 2D, using pixel distance for $N(\cdot)$ instead of world coordinates.

D Closed-World Downstream Mechanical Search Policies

D.1 Problem Statement

We consider the problem of robotic mechanical search for a target object \mathcal{O}_T in a cluttered, semantically organized shelf containing the target and N other rigid objects $\{\mathcal{O}_1, \dots, \mathcal{O}_N\}$ of cuboidal shapes in stable poses. We build on the problem statement and assumptions in Huang et al. [4]. We model the setup as a finite-horizon Partially Observable Markov Decision Process (POMDP). States $s_t \in \mathcal{S}$ consist of the full geometries and poses of the objects in the shelf at timestep t and observations $y_t \in \mathcal{Y} = \mathcal{R}^{H \times W \times 4}$ are RGBD images from a robot-mounted depth camera at timestep t . Actions $a_t \in \mathcal{A} = \mathcal{A}_p \cup \mathcal{A}_s$ are either *pushing* or *suction* actions, where the former are horizontal linear translations of an object along the shelf and the latter pick up an object with a suction gripper and translate it to an empty location on the shelf with no other objects in front of it. We make the following assumptions:

- The dimensions of the shelf are known.
- Each dimension of each object is between size $S_{\min} = 5$ cm and size $S_{\max} = 25$ cm.
- The shelf is semantically organized.
- The names of all objects in the shelf are a subset of a known list of object names.
- Actions cannot inadvertently topple objects or move multiple objects simultaneously.

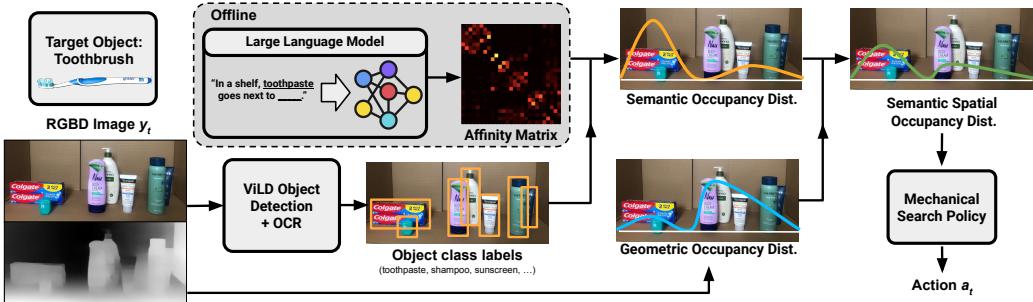


Figure 6: Overview of SMS for Constrained Environments with Downstream Mechanical Search Policy. SMS first receives a scene image and a desired target object. Since the object list is known, it then applies object detection and OCR to identify objects within the scene. SMS then uses a large language model to compute affinities between detected objects to the target object, and it uses these affinities to output a semantic occupancy distribution of the appropriate dimension for the downstream problem. This distribution indicates the likelihood of the physical presence of objects which is used to determine the next action by the downstream mechanical search policy.

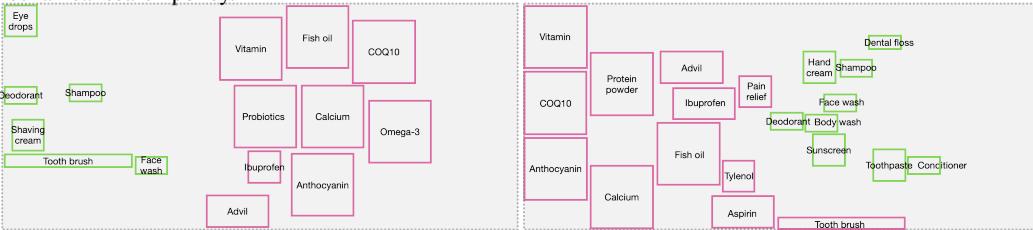


Figure 7: Two examples of layouts for the simulated scenes from the birds-eye view, where the top is the rear of the shelf and the bottom is the front. The layouts are generated from the procedure in Section E.1. The left scene is an example of a layout for a scene with 15 objects and right scene is a layout for a scene with 21 objects. Green rectangles represent objects in personal care categories and pink rectangles represent objects in supplements.

D.2 LAX-RAY

LAX-RAY[4] is a mechanical search policy for shelf environments. LAX-RAY have utilized geometric information by considering object geometries and camera perspective (e.g., tall target objects cannot be occluded by short objects and objects in the center of an image occlude more areas) to facilitate the search. It consists of a perception module and a greedy action selection module. The perception module takes the depth observation and predicts the geometric/spatial occupancy distribution to encode the geometric information. LAX-RAY learns this module on a simulation dataset, with the ground-truth occupancy distribution calculated using Minkowski sum. A greedy action selection module called Distribution Area Reduction (DAR) selects robot actions to greedily reduce the overlap between objects and the distribution. Another search policy has been proposed in [3], named Distribution Entropy Reduction (DER). DER selects the action that would reduce the entropy of the distribution the most after taking the action. We denote the searching pipeline with DER to be LAX-RAY (DER).

We show the SMS pipeline specifically for constrained environments with LAX-RAY in Figure 6. This pipeline was used to conduct the simulation mechanical search experiments in Section E.2.1 and the physical mechanical search experiments in Section E.4.

E Experiments

E.1 Scene Generation

The taxonomy defines a tree where each category is a node and each object name is a leaf node. To create a scene with N objects in a given domain, we begin by uniformly sampling N objects without replacement from the total objects available in that domain. We then generate scenes in

Table 7: Simulation Experiment Results.

	Pharmacy Domain							
	12 objects		15 objects		18 objects		21 objects	
	Successes	# Actions	Successes	# Actions	Successes	# Actions	Successes	# Actions
LAX-RAY	168/190	4.06 ± 0.23	160/186	5.17 ± 0.28	144/188	5.78 ± 0.44	104/177	8.24 ± 0.67
SMS-E	176/190	2.90 ± 0.18	159/186	3.77 ± 0.26	146/188	5.05 ± 0.42	110/177	5.69 ± 0.54
SMS-LLM	176/190	2.66 ± 0.14	162/186	3.26 ± 0.19	150/188	4.25 ± 0.34	118/177	5.47 ± 0.43
	Kitchen Domain							
	12 objects		15 objects		18 objects		21 objects	
	Successes	# Actions	Successes	# Actions	Successes	# Actions	Successes	# Actions
LAX-RAY	185/192	2.15 ± 0.14	182/194	2.97 ± 0.23	177/193	3.99 ± 0.29	159/191	4.36 ± 0.38
SMS-E	186/192	1.56 ± 0.08	188/194	2.15 ± 0.15	184/193	3.00 ± 0.27	167/191	3.07 ± 0.25
SMS-LLM	184/192	1.60 ± 0.10	184/194	2.04 ± 0.13	179/193	2.97 ± 0.26	163/191	3.17 ± 0.28
	Office Domain							
	12 objects		15 objects		18 objects		21 objects	
	Successes	# Actions	Successes	# Actions	Successes	# Actions	Successes	# Actions
LAX-RAY	172/194	2.60 ± 0.18	152/188	4.15 ± 0.38	136/190	4.64 ± 0.37	115/181	5.86 ± 0.56
SMS-E	173/194	3.01 ± 0.22	152/188	3.80 ± 0.31	140/190	4.78 ± 0.44	115/181	5.33 ± 0.50
SMS-LLM	172/194	2.33 ± 0.13	161/188	3.50 ± 0.31	142/190	3.75 ± 0.32	123/181	5.50 ± 0.49

a top-down recursive manner using the taxonomy tree. At the root, we start with the whole shelf available to us. At each node, we split the shelf in half either horizontally or vertically with 50% probability each and recursively continue scene generation in these sub-shelves. If a node has more than 8 descendants, however, we always split the scene horizontally to avoid overcrowding resulting from the aspect ratio of the shelf. At each level of recursion, we accumulate random noise to the eventual placement of each object in the current branch, uniformly sampled from -2 cm to 2 cm. At the last non-leaf node, we place all leaves in random positions within the current level’s sub-shelf. We resolve collisions by iteratively moving objects along the displacement vector between colliding objects and discard scenes where such a procedure takes longer than 1 second to run. We also discard scenes where there is no potential target object that is invisible from the camera’s perspective at the start of the rollout. We reiterate that the taxonomy is *independent* of the language models used to generate affinities. The LLMs are applicable beyond manual semantic categorizations like the Google Taxonomy, but we use this resource for evaluation purposes. The scenes for all simulation, physical, and object detection experiments are generated by this procedure.

We use approximate sizes of these items to generate collision-free scenes. In simulation, we also scale these objects down in order to be able to run experiments on the same-sized shelf, which has an effect similar to running experiments in a larger shelf where more items could originally fit. The scaling factors for the pharmacy and kitchen domains are 0.7, but 0.4 in the office domain due to overall larger objects unable to easily fit and move within a small shelf.

E.2 Simulation Experiments

E.2.1 Simulation Experiments with LAX-RAY

We run an extensive suite of experiments using the same simulator as prior work in mechanical search on shelves [3] and study the benefit brought by SMS. We use a grid search on the average number of actions required in the pharmacy domain with 15 objects to tune the Gaussian smoothing σ to be 50 pixels. We use the same parameters for the other two domains.

We generate scenes with various numbers of objects: $N = 12, 15, 18$, and 21 . We generate 200 scenes for each value of N . In Figure 7, we show example layouts of the scenes as created by the procedure in Section E.1. We discard scenes where the target object starts out visible, resulting in just under 200 scenes for each value of N . Termination occurs when at least $X = 1\%$ of the target object becomes visible or reaching maximum action number $2N$. The reason for the low threshold is that the DAR policy has trouble making progress on a partially revealed target object [3], which may dilute the comparison between different methods for generating semantic distributions.

We report results for all numbers of objects N in Table 7, SMS-LLM outperforms both SMS-E, while also beating LAX-RAY across various values of N in terms of success rate (by an additional 30/741 scenes) and average number of actions required (by 32.4%). A point of note is that the action differential percentage grows as the number of objects increases. At 21 objects, LAX-RAY requires 8.24 actions on average, whereas SMS requires just 5.47. This trend agrees with intuition that it is unscalable to search large environments with no semantic intuition.

E.3 Simulation Experiments with Object Detection Noise

Method	No noise	10% Noise	50% Noise	90% Noise	LAX-RAY
# of Actions	3.81 ± 0.31	4.20 ± 0.38	4.44 ± 0.41	4.83 ± 0.47	5.12 ± 0.43

Table 8: Experiment to determine the impact of object detection noise on task performance (# of actions). For SMS-LLM, we randomly perturb the object detection (i.e. randomly select a label from the object list) with probability P . We do 400 rollouts over the categories of 12, 15, 18, 21 objects in the scene for the pharmacy domain. We report the average number of actions taken to reveal the target object and standard error. We see the general trend as object detection noise increases the task performance decreases.

We study the impact of the object detection accuracy on the task performance. We randomly change the object labels with a probability P . The results are shown in Table 8, where $P = 0.1, 0.5, 0.9$. The number of actions needed to find the occluded object increases as P increases. This is because random perturbations can cause the semantic distribution to approach a uniform distribution thus not modifying the existing action of the downstream policy. Therefore, Table 8 indicates there is also a strong positive correlation between object detection accuracy and task performance.

E.3.1 Simulation Experiments with DER

Policy	12 objects		15 objects		18 objects		21 objects	
	Success	# Actions						
LAX-RAY (DER)	84%	5.79 ± 0.38	74%	7.69 ± 0.54	62%	8.08 ± 0.64	42%	9.52 ± 0.72
SMS-LLM	90%	4.42 ± 0.39	81%	5.06 ± 0.43	71%	7.11 ± 0.60	45%	6.87 ± 0.67

Table 9: Simulation experiments results of SMS-LLM with DER for the Pharmacy domain. We ablate the downstream policy and see that SMS-LLM outperforms LAX-RAY with DER. We report the number of rollouts that were successful and the mean actions to retrieve the occluded object and the standard error.

We integrate SMS with a different downstream policy Distribution Entropy Reduction (DER) from [3]. We multiply the semantic distribution with the geometric distribution as the input to DER. DER selects the action minimizing the distribution entropy after taking the action. We use the same setup and scenes as in Section E.2.1. We report the results for 100 scenes with 12 objects in Pharmacy domain in Table 9.



Figure 8: Here are 3 example scenes from the physical mechanical search experiments in the constrained environment setting.

E.4 Physical Experiments

We generate the physical scenes with the scene generation procedure outlined in Section E.1 to ensure the scenes were not biased. Examples of physical environment layouts are shown in Figure 8.

Because the RealSense camera is not able to capture the fine details of the text on the objects when observing the entire scene at resolution 640×480 pixels, we perform a three-stage scan of the scene by moving the end-effector to 3 adjacent positions, all of which are closer to the shelf, where the text is more easily readable. At each of these poses, we take a picture of the scene, project the known world position of the objects to the new camera frame, identify text with OCR, and assign

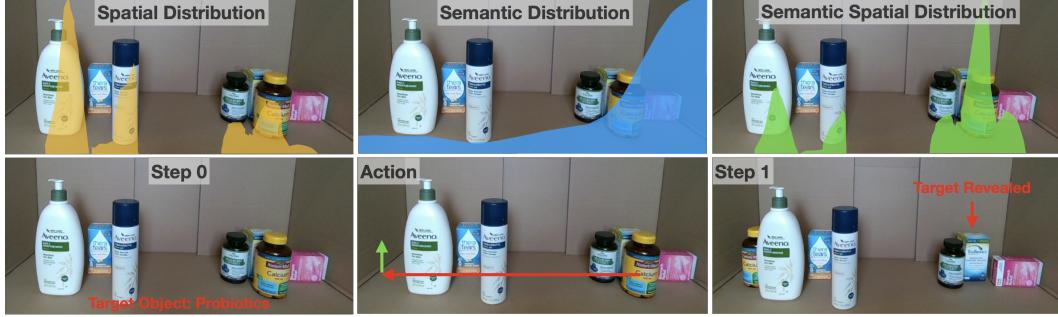


Figure 9: Physical rollout example with the target object being the probiotics. **Top:** the spatial distribution, semantic distribution and semantic spatial distribution for step 0. **Bottom:** RGB observations at step0, the action given by DAR and the RGB observation after executing the action.

each text detection to the object it is contained in. If there are detections on the same object from multiple scan locations, we use the OCR that has the lowest entropy for its distribution, a measure of confidence. During the physical experiments rollouts, when the action given by the policy causes unintentional toppling or a missed grasp due to depth sensor noise, we reset the object to undo the action and run the policy again.

We show a physical experiment rollout with the target object being the probiotics as in Figure 9. In this rollout, the spatial distribution generated based on geometric information by LAX-RAY indicates the left side of the shelf occludes more area. However, the semantic distribution generated by SMS indicates the target object is more likely to be on the right. This is because other objects from the supplements category where the target object probiotics belongs to are visible on the right. Combining the spatial distribution and semantic distribution into the semantic spatial distribution takes into account both the geometry and semantic information and results in a more accurate distribution.

E.4.1 Ablating Object Lists for Semantic Distribution for Physical Experiments

Metric	Uniform Dist.	SMS-LLM w/o Object List	SMS-LLM w/ Object List
JSD ↓	0.554 ± 0.006	0.421 ± 0.032	0.382 ± 0.036

Table 10: We measure the deviation in the semantic distribution generated by these methods and the ground truth using JSD. SMS with the object list, which uses Object Detection+OCR, outperforms SMS without the object list, which uses the Crop Generation + Image Captioning pipeline.

To evaluate the benefit of object lists, we compare the performance of our method on shelves with and without access to the object list by computing the Jensen-Shannon Distance (JSD) [109] between the generated distribution and the ground truth distribution on the 12 physical shelves as in Section 5.2.2. From Table 10, we see that SMS achieves a better semantic distribution compared to a uniform prior in both cases. The SMS-LLM w/o Object List uses the same Crop Generation + Image Captioning pipeline as the open-world experiments which is equivalent to using a VLM for scene understanding. The SMS-LLM w/Object List uses the object list for object detection and to refine the labels using OCR. We see that knowing the object list improves results, which is expected as it reduces the noise in the scene understanding and leads to a higher quality of the semantic distribution.

E.5 Experiments for Open-World Environments

Ablations	w/o CLIP Weighting	BLIP-IC	w/o SAM	SMS-E
IoU	0.307 ± 0.038	0.310 ± 0.043	0.286 ± 0.038	0.391 ± 0.039

Table 11: IoU results for ablated SMS-E. **w/o CLIP Weighting** doesn't use CLIP to refine the generated captions as described in Section 4.2. **BLIP-IC** use BLIP-IC to get the descriptions for each crops instead of BLIP-2. BLIP-IC is linked in the Appendix Section E.5. **w/o SAM** doesn't use crops given by SAM and crops generated by multi-scale sliding windows are used.

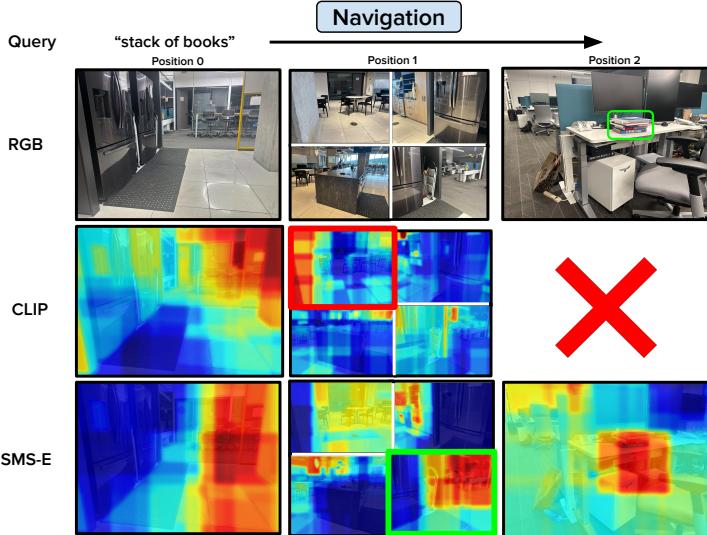


Figure 10: **Object navigation experiment in BAIR Office Kitchen.** A short horizon navigation example where we start at position 0 and end at position 2. SMS is able to correctly maneuver to the stack of books while CLIP fails because its bag-of-words nature is susceptible to incorrectly assigning high probability to the pillars in the scene as they are semantically related to “stack.”

We also ablate the modules in SMS-E with results shown in Table 11, where we study the impact of using CLIP weighting, different image captioning models and SAM crops. As mentioned previously that image captioning can be noisy, we use CLIP to verify the captions. We refer this as CLIP weighting. Without this, the performance drops by 21%. When we use BLIP-IC instead of BLIP-2 for image captioning, the performance drops by 21%. Finally, without cropping using SAM to get object-centered crops, the performance drops by 27%.

More examples of the semantic distribution comparison between SMS and CLIP-based models are shown below. **BLIP-IC** is the large image captioning model of BLIP.

We also conduct a preliminary navigation experiment where a mobile robot follows the downstream navigation policy described in Section 4.3 and selects the physical location to move to based on the semantic distribution from CLIP and SMS-E. As shown in Figure 10, CLIP makes an incorrect turn (at position 1 it continues in the direction of the view with the red box) because of its bag-of-words behavior and attributes “stack of books” to having higher semantic similarity to concrete pillars in the scene rather than the area with office desks and chairs. SMS-E continues towards the office (green box in position 1) and finds a stack of books successfully.

E.6 Object Lists in Closed-World Environments

Pharmacy Domain : vitamins , fish oil , omega-3 , calcium , probiotics , protein powder , COQ10 , anthocyanin , shampoo , conditioner , toothpaste , toothbrush , dental floss , face wash , sunscreen , lotion , hand cream , body wash , aspirin , tylenol , ibuprofen , advil , pain relief , shaving cream , eye drops , deodorant , band-aid

Kitchen Domain : spoon , ladle , spatula , tongs , whisk , fork , peeler , grater , saucepan , frying pan , salt , pepper , cumin , coriander , basil , turmeric , parsley , oregano , sugar , flour , cornstarch , oats , quinoa , rice

Office Domain : pen , pencil , highlighter , sticky note , binder paper , printer paper , index card , paper clip , rubber band , stapler , staples , tape dispenser , 3-hole punch , dry erase marker , sharpie , label maker , notebook , eraser , white-out , calculator , thumbtack , pencil sharpener , bubble wrap , styrofoam , packing tape , shipping boxes , ethernet cable , modem , router , network card , network bridge , headphones , speakers , aux cable , microphone , keyboard , mouse , USB adapter , hard drive , flash drive

F Object Lists and Examples for Open-World Environments

In this section, we show more examples of the semantic distributions generated by different methods from the static dataset and all the scenes.

Grocery Object list: 'blender', 'juicer', 'spatula', 'spray tan', 'sunglasses', 'gardening gloves', 'grass seeds', 'headphones', 'pruning shears', 'SD card', 'crayons', 'paper towel', 'plunger', 'Powerade', 'router', 'bottle opener', 'corkscrew', 'Danimals', 'Paneer', 'yogurt', 'bagel', 'baguette', 'daisy', 'danish pastry', 'red rose', 'toaster strudel', 'cocoa powder', 'incense sticks', 'succulents', 'condensed milk', 'kale', 'lotion', 'scotch tape', 'garlic bread', 'moisturizing masks'

Office Object list : 'box of paper', 'cat food', 'ice', 'leftover meatloaf', 'three ring binder', 'Budweiser beer', 'coke can', 'lion figurine', 'tequila', 'panda soft toy', 'acetone', 'facial cotton pad', 'HDMI cable', 'lipstick', 'pillow', 'beef patties', 'expo marker', 'mayo', 'relish', 'USB flash drive', 'thumb tacks', 'chain', 'Pringles', 'trail mix', 'iPhone charger', 'throw blanket', 'shears', 'emergency whistle', 'office party notice', 'yogurt', 'napkin holder', 'Academy Award'

Home Object list : 'bar soap', 'boarding pass', 'empty bottle', 'pajamas', 'toothpaste', 'microphone', 'mail', 'laundry brush', 'paint roller', 'hand wraps', 'paddle', 'alarm clock', 'duvet', 'medal', 'pool balls', 'pool rack', 'soap', 'tissues', 'Neosporin', 'HP scanner', 'resistance bands', 'jump rope'

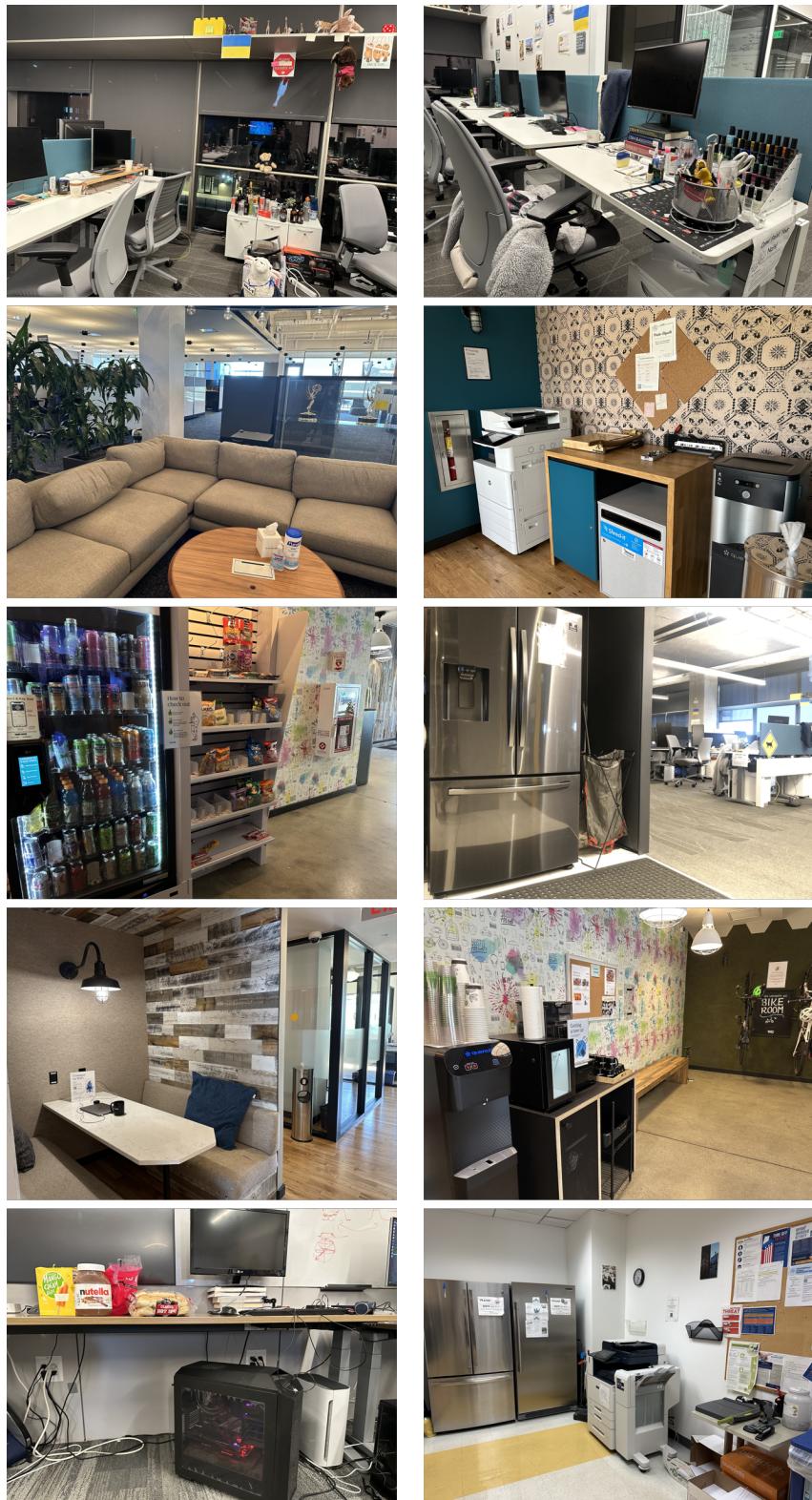


Figure 11: Office environments.

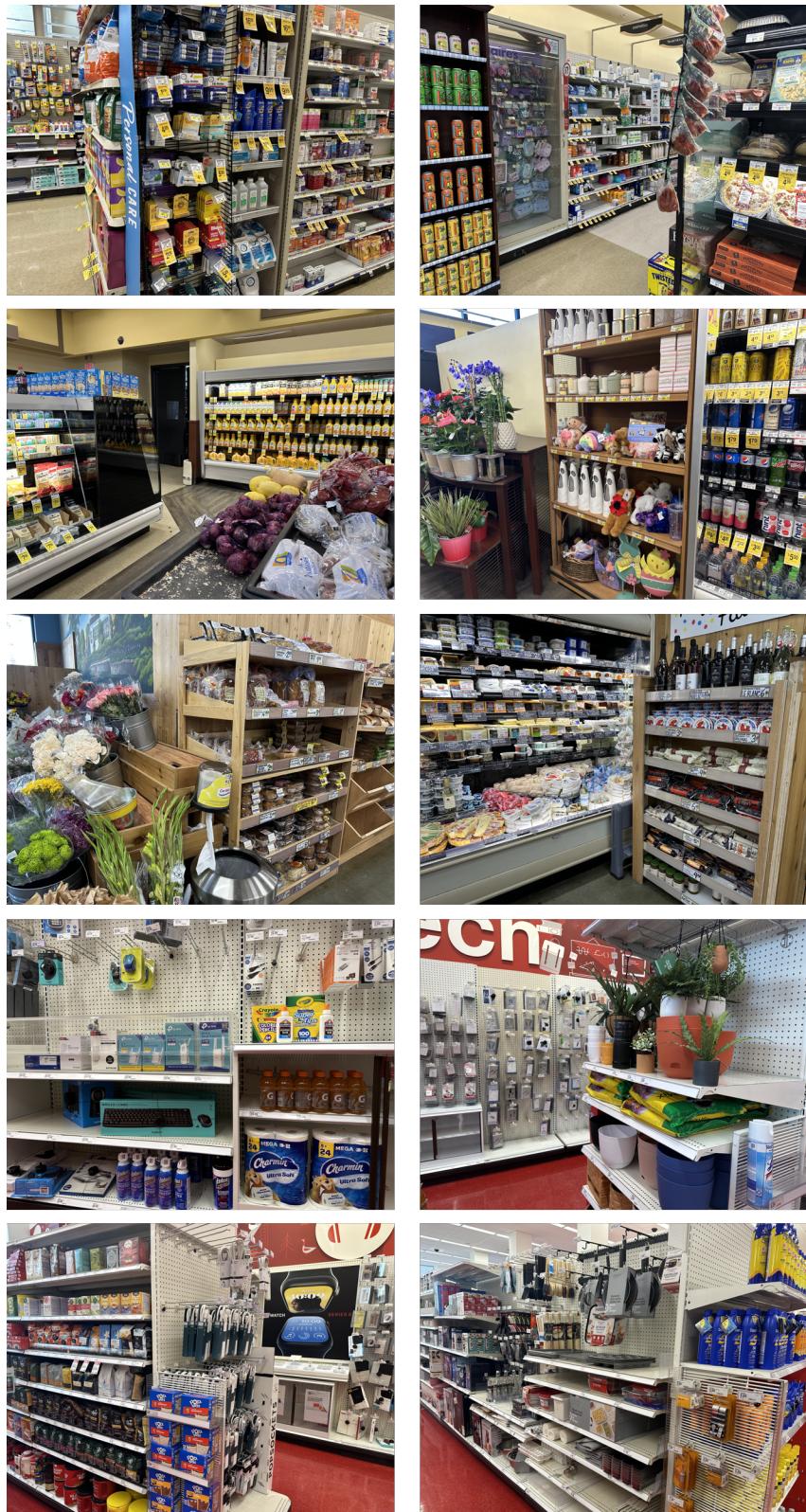


Figure 12: Grocery stores environments.

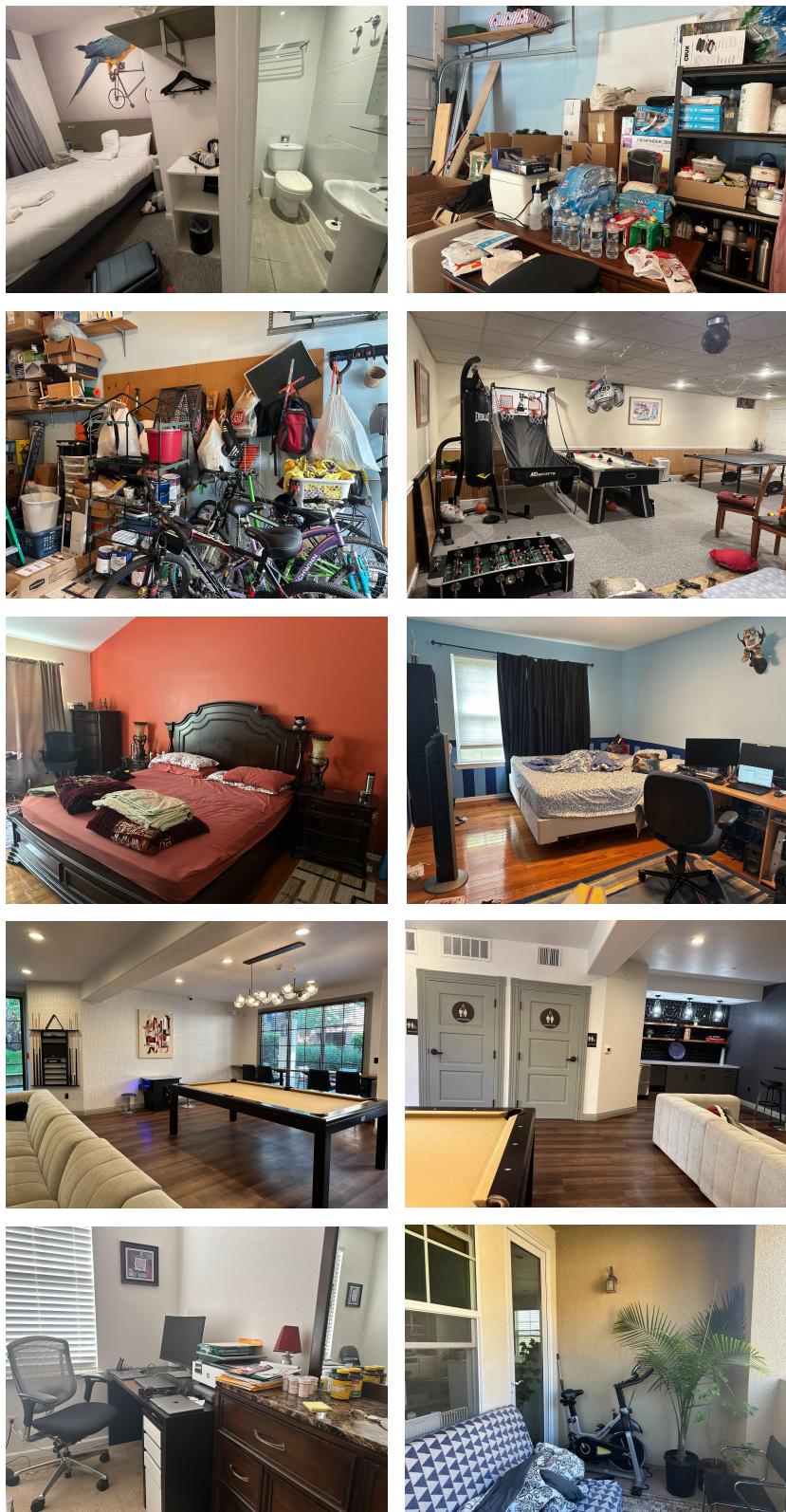


Figure 13: Home environments.

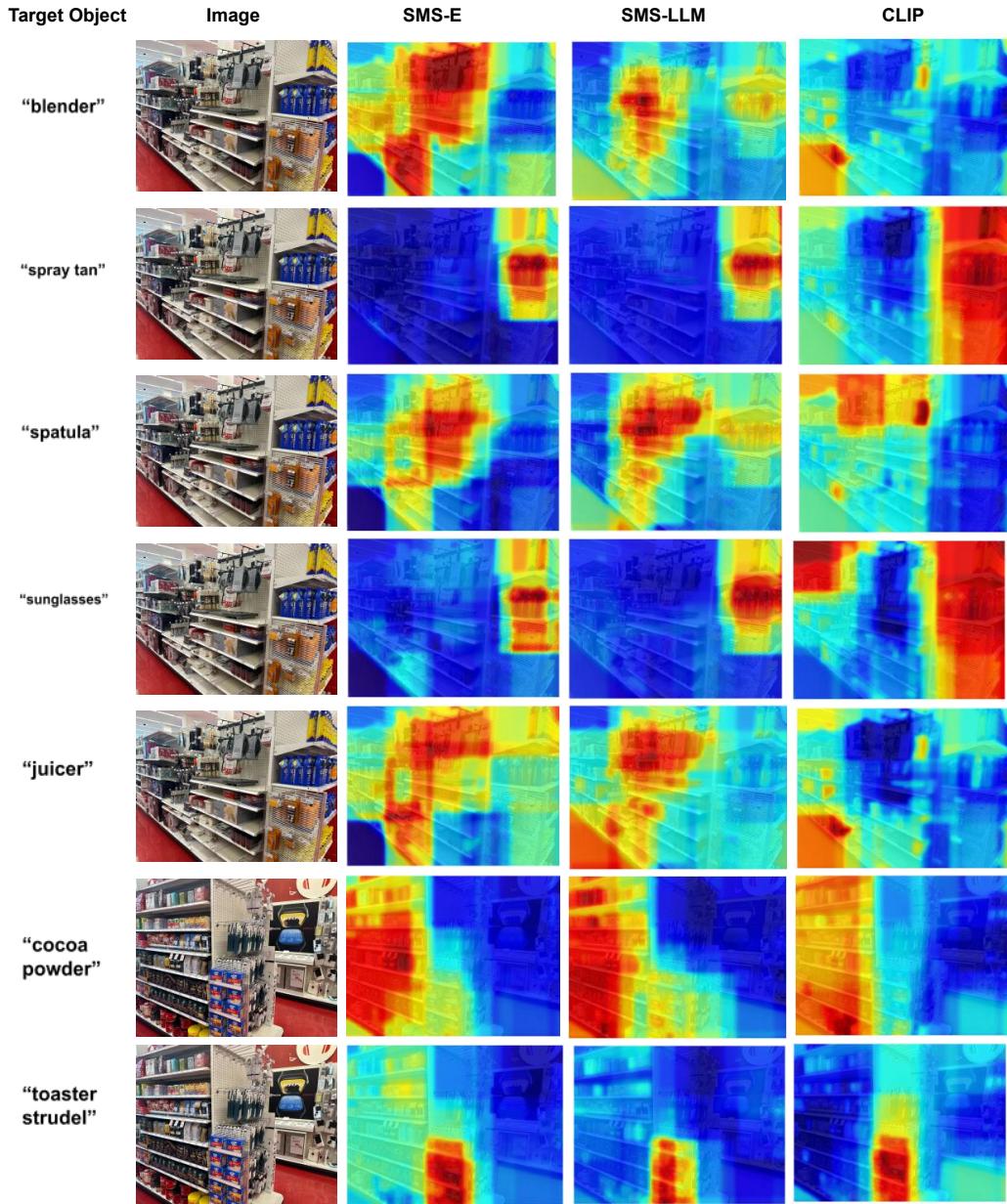


Figure 14: Example set 1 of the semantic distributions generated by different methods for grocery stores.

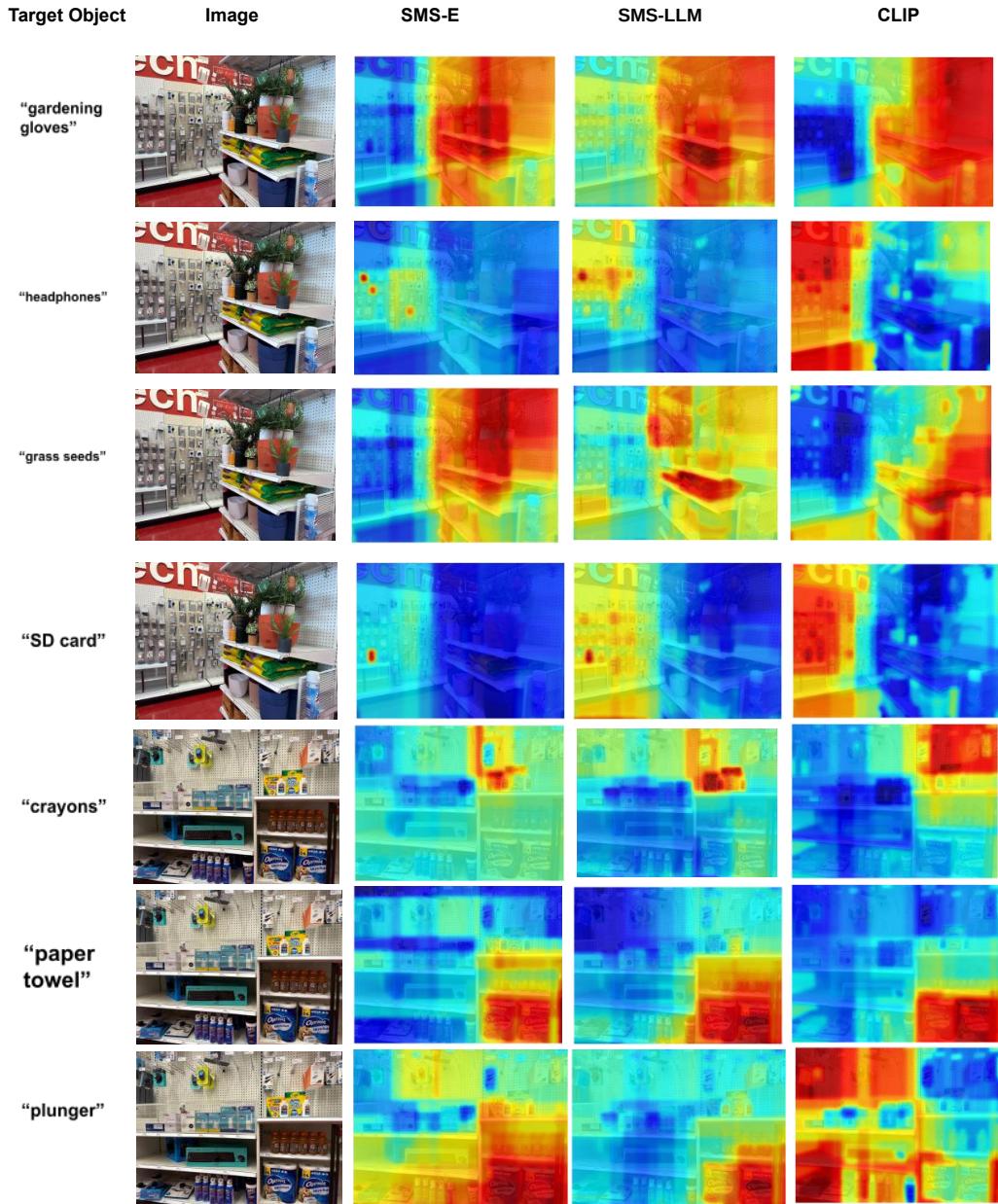


Figure 15: Example set 2 of the semantic distributions generated by different methods for grocery stores.

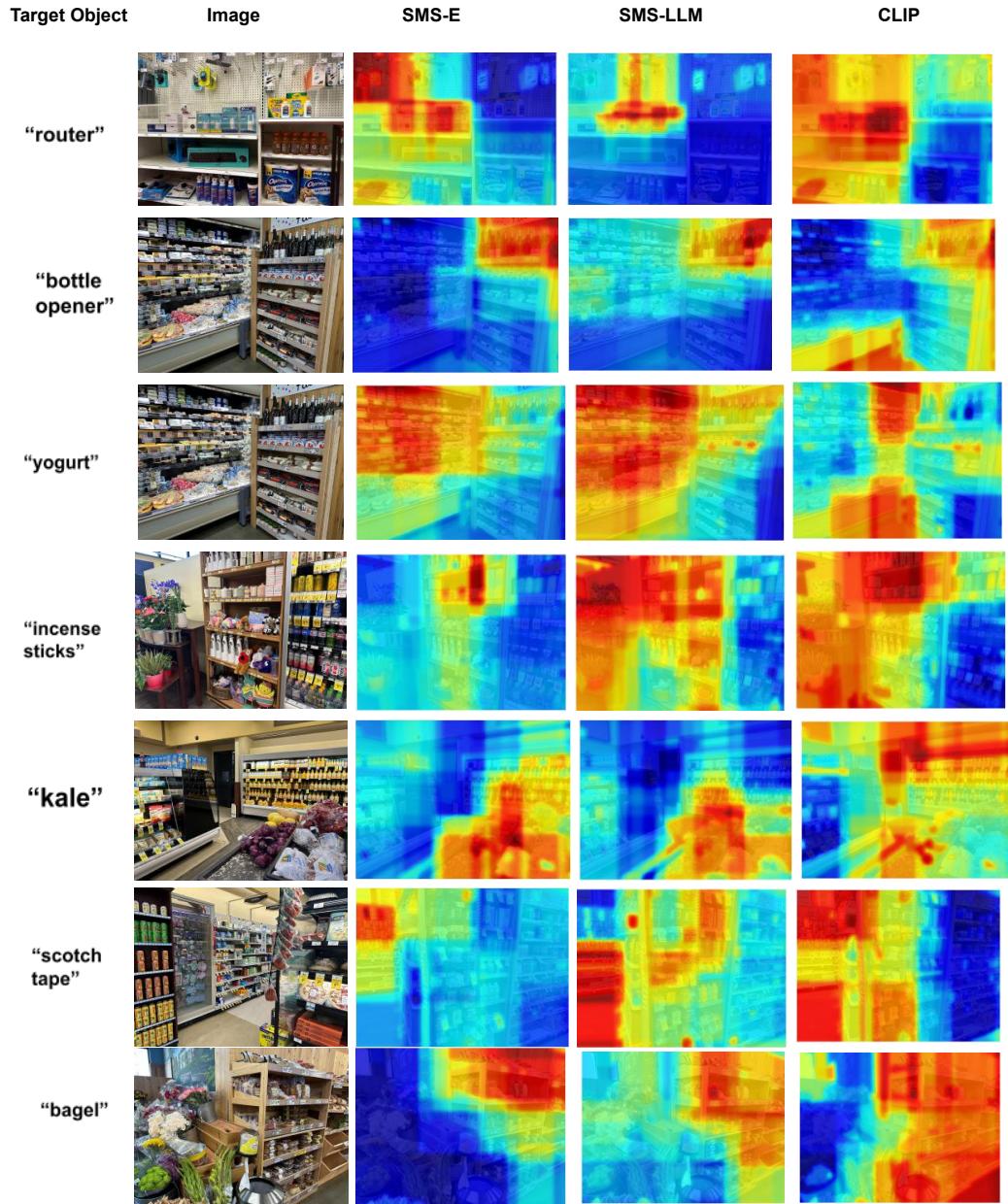


Figure 16: Example set 3 of the semantic distributions generated by different methods for grocery stores.

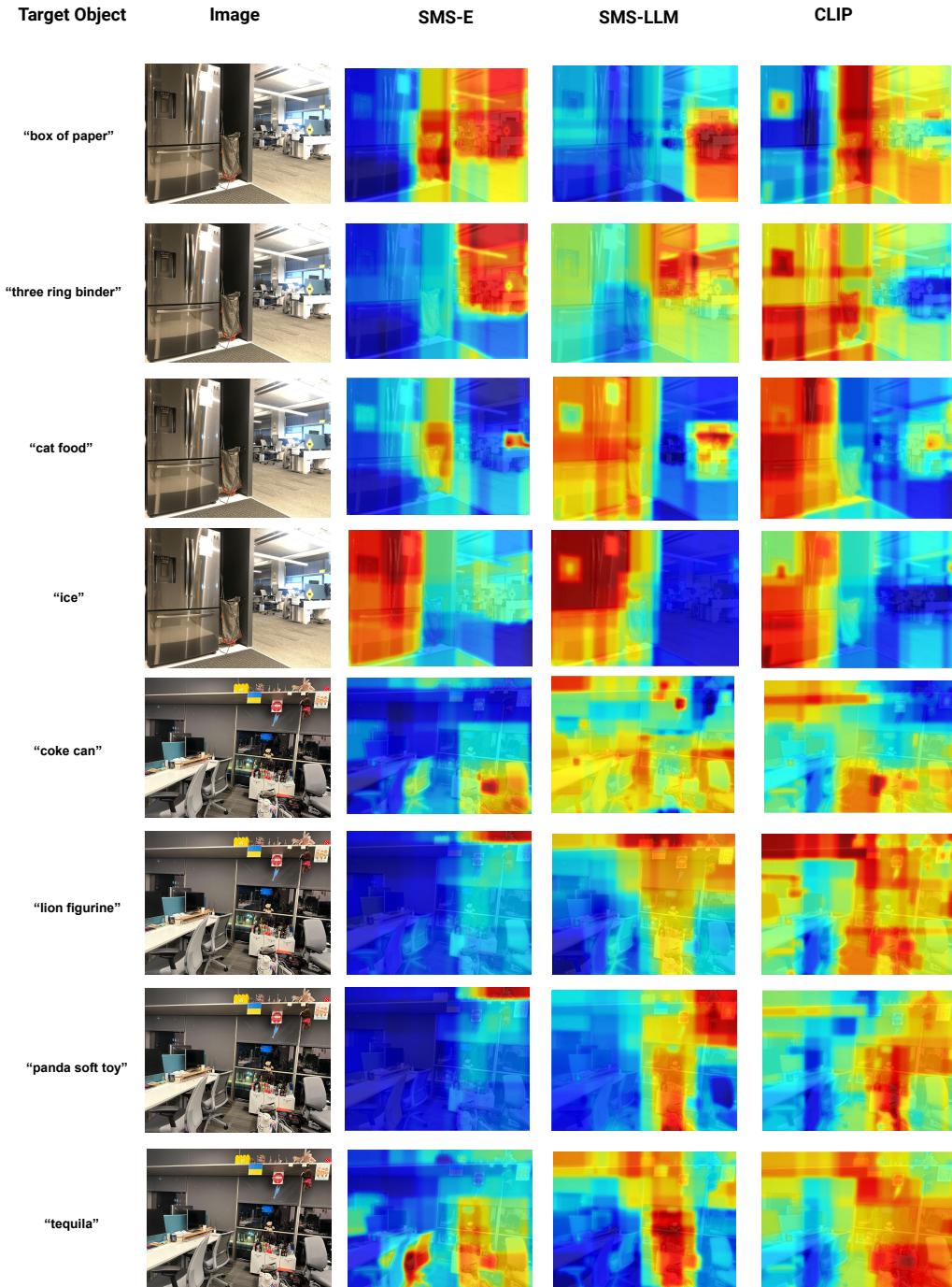


Figure 17: Example set 1 of the semantic distributions generated by different methods for offices.

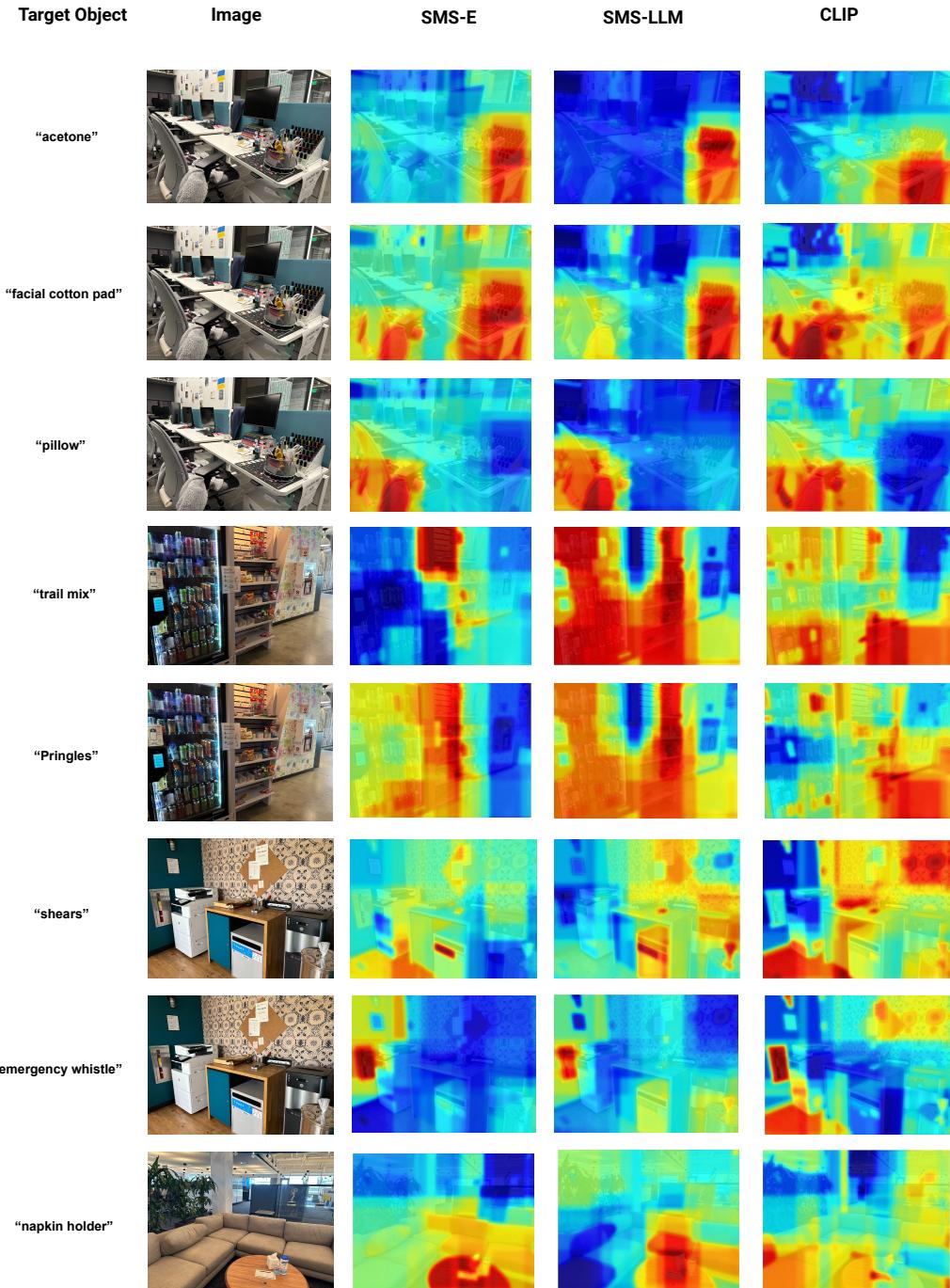


Figure 18: Example set 2 of the semantic distributions generated by different methods for offices.

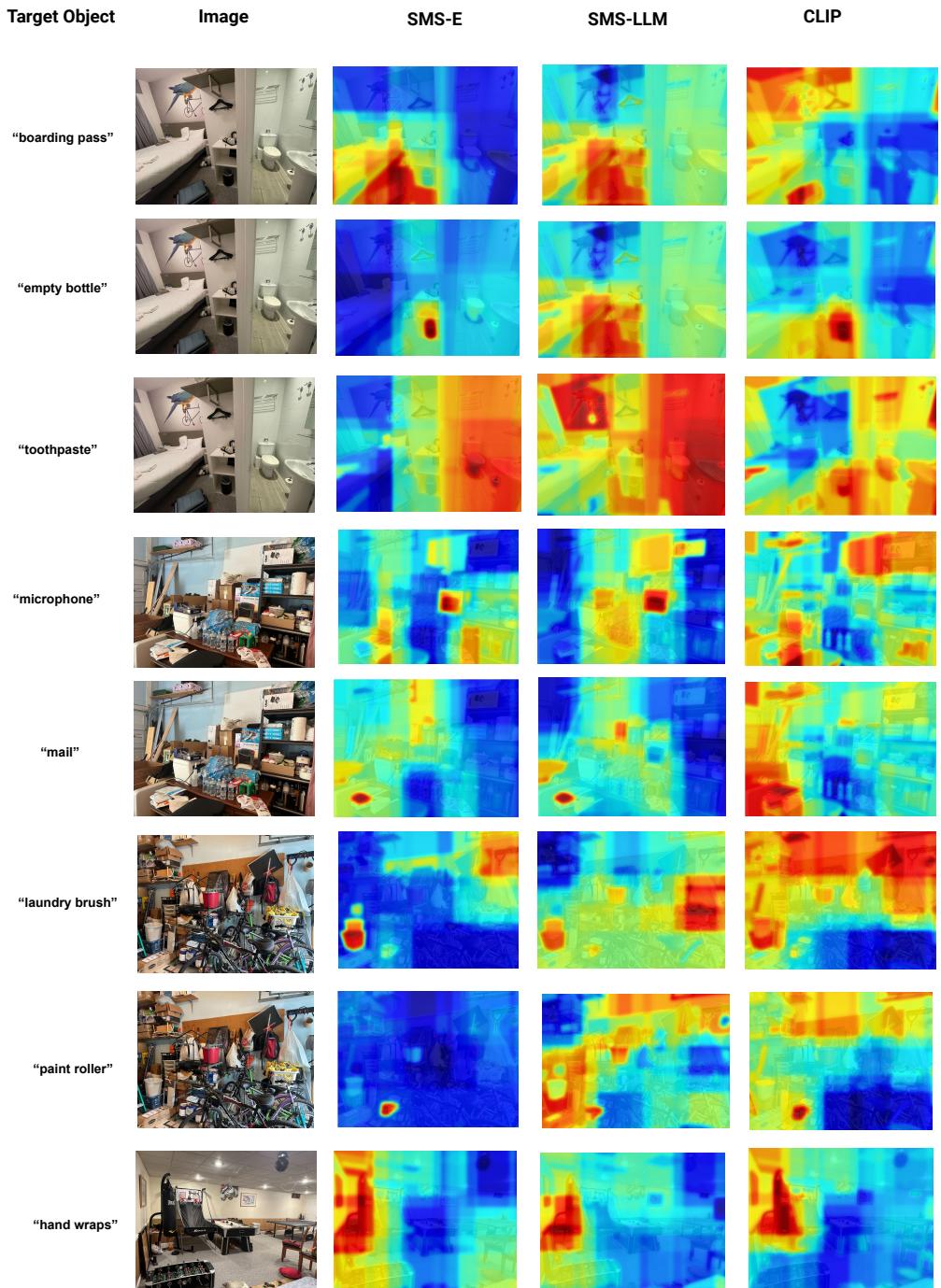


Figure 19: Example set 1 of the semantic distributions generated by different methods for houses.

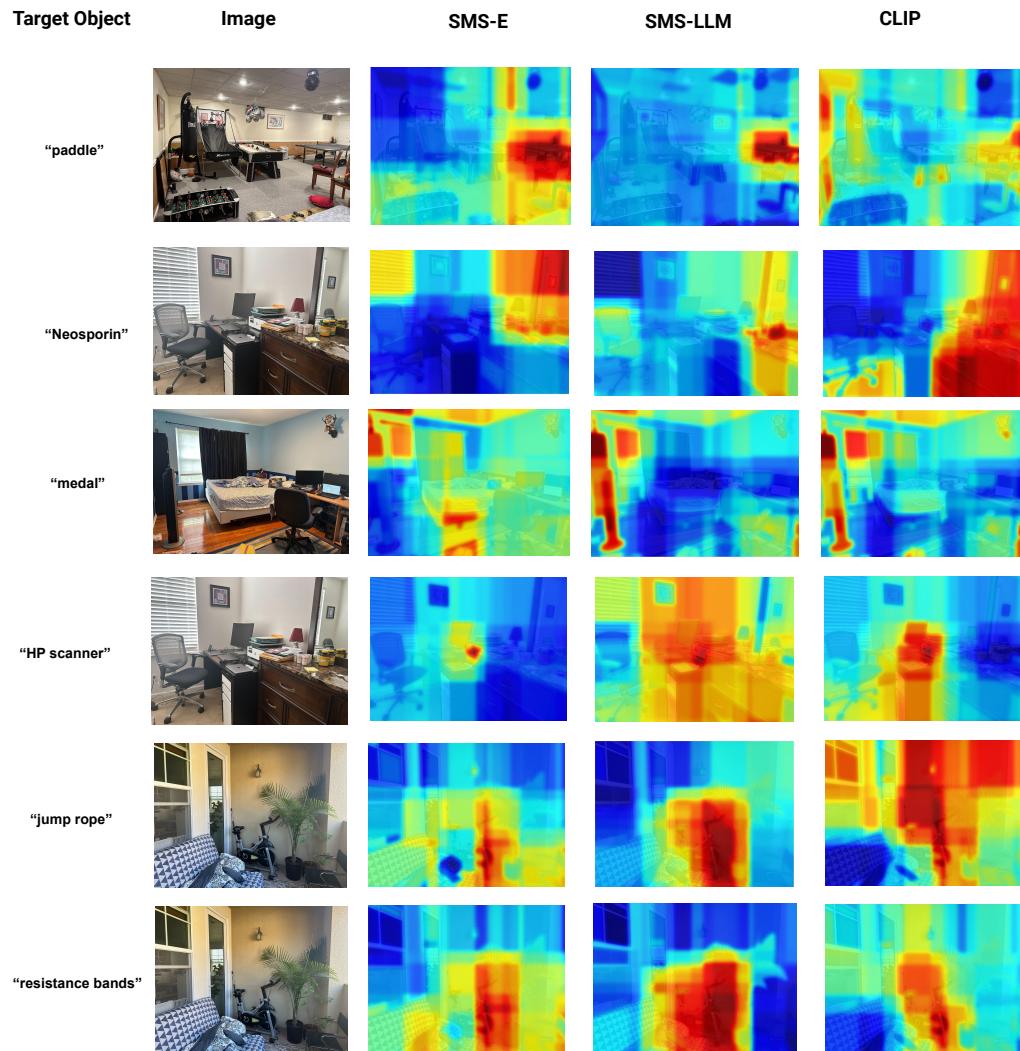


Figure 20: Example set 2 of the semantic distributions generated by different methods for houses.