# Housekeep: Tidying Virtual Households using Commonsense Reasoning

Yash Kant[1,2⋆], Arun Ramachandran[2], Sriram Yenamandra[2], Igor Gilitschenski[1], Dhruv Batra[2,3], Andrew Szot[2†], and Harsh Agrawal[2†]

[1]University of Toronto, [2]Georgia Tech, [3]Meta AI
[†] Equal Contribution

**Abstract.** We introduce **Housekeep**, a benchmark to evaluate commonsense reasoning in the home for embodied AI. In Housekeep, an embodied agent must tidy a house by rearranging misplaced objects *without explicit instructions specifying which objects need to be rearranged.* Instead, the agent must learn from and is evaluated against human preferences of which objects *belong* where in a tidy house. Specifically, we collect a dataset of where humans typically place objects in tidy and untidy houses constituting 1799 objects, 268 object categories, 585 placements, and 105 rooms. Next, we propose a modular baseline approach for Housekeep that integrates planning, exploration, and navigation. It leverages a fine-tuned large language model (LLM) trained on an internet text corpus for effective planning. We find that our baseline planner generalizes to some extent when rearranging objects in unknown environments. See our webpage for code, data and more details: `https://yashkant.github.io/housekeep/`

## 1 Introduction

Imagine your house after a big party: there are dirty dishes on the dining table, cups left on the couch, and maybe a board game lying on the coffee table. Wouldn't it be nice for a household robot to clean up the house *without needing explicit instructions specifying which objects are to be rearranged*?

Building AI reasoning systems that can perform such housekeeping tasks is an important scientific goal that has seen a lot of recent interest from the embodied AI community. The community has recently tackled various problems such as navigation [3, 7, 21, 33, 46, 69], interaction and manipulation [19, 64], instruction following [4, 62], and embodied question answering [17, 22, 71]. Each of these tasks defines a goal, e.g. navigating to a given location, moving objects to correct locations, or answering a question correctly. However, defining a goal for tidying a messy house is more tedious – one will have to write down a rule for where every object can or cannot be kept. Previous works in semantic reasoning frameworks for physical and relational commonsense [1, 9, 10, 16, 38, 39] are often

⋆ Work done partially when visiting Georgia Tech

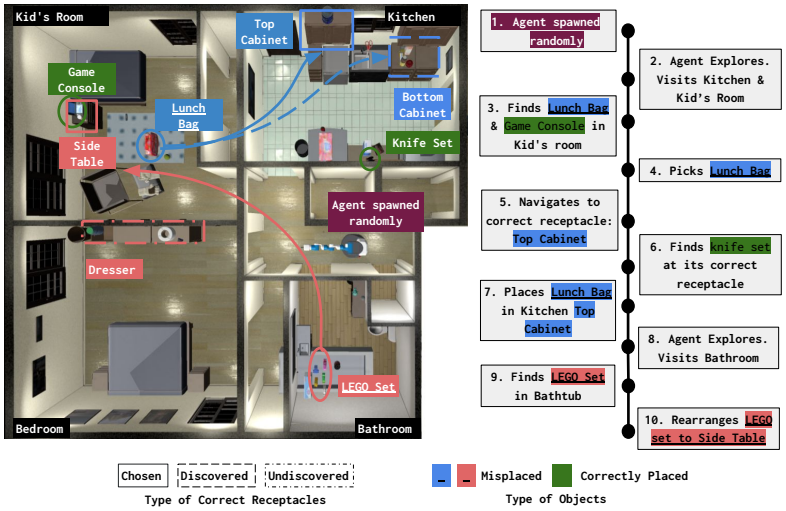



Fig. 1: In Housekeep, an agent is spawned in an untidy environment and tasked with rearranging objects to suitable locations without explicit instructions. The agent explores the scene and discovers misplaced objects, correctly placed objects, and receptacles where objects belong. The agent rearranges a misplaced object (like a lunch box on the floor in the kid's room) to a better receptacle like the top cabinet in the kitchen.

limited to specific settings (*e.g.* evaluating multi-relational embeddings) without instantiating these tasks in a physically plausible scenario, or by not capturing the full context of a complete household (*e.g.* table-top organization). We believe the time may be right to bridge the gap between the above two lines of research.

We introduce the Housekeep task to benchmark the ability of embodied AI agents to use physical commonsense reasoning and infer rearrangement goals that mimic human-preferred placements of objects in indoor environments. Figure 1 illustrates our task, where the Fetch robot is randomly spawned in an unknown house that contains unseen objects. Without explicit instructions, the agent must then discover objects placed in the house, classify the misplaced ones (LEGO set and lunch bag in Figure 1), and finally rearrange them to one of many suitable receptacles (matching color-coded square). We collect a dataset of human preferences of object placements in tidy and untidy homes and use this dataset for: a) generating semantically meaningful initializations of unclean houses, and b) defining evaluation criteria for what constitutes a clean house. This dataset contains rearrangement preferences for 1799 objects, in 585 placements, in 105 rooms, constituting 1500+ hours of effort from 372 total annotators with 268 object categories curated from the Amazon-Berkeley [29], YCB objects [77], Google Scanned Objects [53], and iGibson [61] datasets. Housekeep evaluates how an agent is able to rearrange novel objects not seen during training.

Housekeep is a challenging task for several reasons. First, agents need to reason about the correct placement of *novel* objects. Second, agents in Housekeep must operate in unseen environments using only egocentric visual observations.

In the absence of any goal specification, the agent must *explore* areas that get cluttered frequently (*e.g.* coffee table, kitchen counter) for discovering potentially misplaced objects, and also find their suitable receptacles. Finally, since the environment is partially observable, the agent must continuously re-plan for when and where to rearrange objects via commonsense reasoning. For instance, on discovering a toy on the coffee table in the living room, the agent may choose to not rearrange it immediately if it hasn't discovered a more suitable receptacle such as the closet in the kid's room yet. The agent also has to reason about multiple potentially correct receptacles for any given object. For example, a toy could go in the closet in the master bedroom or in the kid's room.

We propose a modular baseline and demonstrate that embodied (physical) commonsense extracted from large language models (LLMs) [11, 40] or traditional GloVe [49] vectors serves as an effective planner. Specifically, we find that finetuning these embeddings on a subset of human preferences generalizes well, and helps to reason about correct rearrangements for novel objects never seen during training. We integrate this planning module into a hierarchical policy that coordinates navigation, exploration, and planning as a baseline approach to Housekeep. Our hierarchical approach with the aid of few perfect sensors achieves an object success rate of 0.23 for unseen (versus 0.30 on seen objects). We also qualitatively analyze different failure cases of our baseline.

## 2 Related Work

**Embodied AI Tasks**. In recent times, we have seen a proliferation of Embodied AI tasks. Benchmarks on indoor navigation use point-goal specification [24, 60], object-goal [7, 69], room navigation [46], and language-guided navigation [4, 66]. Some interactive tasks study the agent's ability to follow natural language instruction such as ALFRED [62] and TEACh [48] while others focus on rearranging objects following a geometric goal or predicate based specification [21,63,64,70]. [6] provides a summary of rearrangement tasks. All these tasks require an explicit goal specification lifting the burden of learning semantic compatibility of objects and their locations in the house from the agent. In contrast, in this work, we argue that agents shouldn't require an explicit goal specification to perform household tasks such as tidying up the house. Instead, it should use its common sense knowledge to infer the human-preferred goal state.

**Capturing Human Preferences.** Several works (summarized in Appendix A) in robotics model human preferences for assistive robots. Some [31] looked at furniture rearrangement based on surrounding human activities (*e.g.* standing by the kitchen shelf) while others [1, 32] looked at table-top or a shelf rearrangement conditioned on a user. We differ from these works because we are interested in tidying up *entire houses* instead of a particular shelf or a table-top. In addition, the agent needs to operate with partial observations, and generalize to unseen environments and object types. [65] comes closest to our work. They learn a spatial model of object placements in a tidy environment. Our benchmark has a

larger scale (1799 objects spanning 268 categories vs $\leq 55$ object instances; 100+ room configurations vs 1 scene in [65]). Our benchmark also tests generalization to *unseen* objects, utilizing a dataset of human preferences instead of learning from a small set of tidy house instances. Dealing with unseen objects is important for real applications since humans can bring new objects into the home.

**Commonsense Reasoning**. Prior work in Natural Language Processing has studied the problem of imbuing commonsense knowledge in AI systems, from social common-sense knowledge [10, 35, 56, 58, 59, 73] to understand the likely intents, goals, and social dynamics of people, abductive commonsense reasoning [8], next event prediction [74,75], to temporal common sense knowledge about temporal order, duration, and frequency of events [2,23,44,76]. Most similar to our work is the study of physical commonsense knowledge [9] about object affordances, interaction, and properties (such as flexibility, curvature, porousness). However, these benchmarks are static in nature (as a dataset of textual or visual prompts). Our task, on the other hand, is instantiated in an embodied interactive environment and more realistic – the environment is partially observed, and the agent has to explore unseen regions, discover misplaced objects and use common-sense reasoning to infer compatibility between objects and receptacles.

**Application of Large Language Models**. With the introduction of Transformer [67] style architectures, we have seen a diverse range of applications of large language models (LLMs) pre-trained on web-scale textual data. They have not only performed well on natural language processing tasks [40, 67], but the implicit knowledge learned by these models have shown to be effective for other unrelated tasks [42]. LLMs has had a lot of success in vision-and-language tasks like Visual Question Answering (VQA) [41, 68] and image captioning [27, 37], external knowledge-based question answering [11, 54] and construction [10]. They have also been shown to improve performance on Embodied AI tasks like vision-and-language navigation [43, 45], instruction following [25], and planning for embodied tasks [28, 36]. In our work, we explore if language models can display common-sense knowledge of how humans prefer to tidy up their homes.

## 3 Housekeep: Task and Dataset

Here, we define the Housekeep task and its instantiation in the Habitat [60, 64].

### 3.1 Task Specification

**Definition**: Recall, in Housekeep an embodied agent is required to clean up the house by rearranging misplaced objects to their correct location within a limited number of time steps. The agent is spawned randomly in an unseen environment and has to explore the environment to find misplaced objects and put them in their correct locations (receptacles).

**Scenes and Rooms**: We use 14 interactive and realistic iGibson scenes [61]. These scenes span 17 room types (*e.g.* living room, garage) and contain multiple rooms with an average of 7.5 rooms per scene. We remove one scene from the original iGibson dataset (`benevolence_0_int`) because it's unfurnished.

**Receptacles**: We define *receptacles* as flat horizontal surfaces in a household (furniture, appliances) where objects can be found – misplaced or correctly placed. We remove assets that are neither objects nor receptacles (*e.g.* windows, paintings, etc) and end up with 395 unique receptacles spread over 32 categories. An iGibson scene can contain between 19-78 receptacles. Notice that a valid object-receptacle placement requires the additional context of what room the receptacle is situated in. For example, a counter in the kitchen is a suitable receptacle to place a fruit basket, however, a counter in the bathroom may not be. Hence, we care about the diversity in combinations of room-receptacle occurrences for Housekeep. Overall, there are 128 distinct room-receptacles in the iGibson scenes.

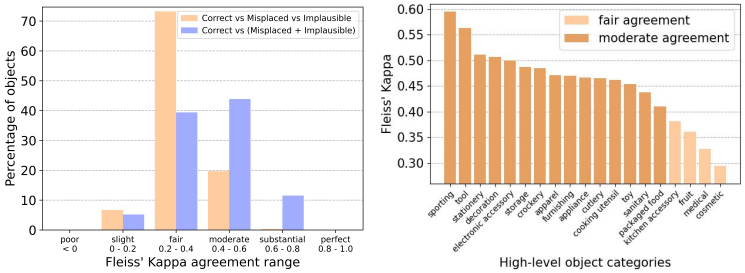
**Objects**: We collect objects from four popular asset repositories – Amazon Berkeley Objects [29], Google Scanned Objects [53], ReplicaCAD [64], and YCB Objects [12]. We filter out objects with large dimensions (*e.g.* ladders, televisions), and objects that do not usually move in a household (*e.g.* garbage cans). After filtering, we have 1799 unique objects spread across 268 categories. We further categorize these objects into 19 high-level semantic categories such as stationery, food, electronics, toys, etc. More details about the filtering, semantic classes, and high/low-level object categories are in the Appendix B.

**Agent**: We simulate a Fetch robot [55], which has a wheeled base with a 7-DoF arm manipulator, parallel-jaw gripper, and an RGBD camera (90° FoV, $128 \times 128$ pixels) on the robot's head. The robot moves its base and head through five discrete actions – move forward by 0.25m, rotate base right or left by 10°, rotate head camera up or down (pitch) by 10°. The robot interacts with objects through a "magic pointer abstraction" [6] where at any step the robot can select a discrete "interact" action. We provide more details in Appendix E.2.

### 3.2 Human Preferences Dataset: Where Do Objects Belong?

The central challenge of Housekeep is understanding how humans prefer to put everyday household objects in an organized and disorganized house. We want to capture where objects are typically found in an unorganized house (before tidying the house), and in a tidy house where objects are kept in their correct position (after the person has tidied the house). To this end, we run a study on Amazon MTurk [15, 57] with 372 participants. Each participant is shown an object (*e.g.* salt-shaker), a room (*e.g.* kitchen) for context, and asked to classify all the receptacles present in the room into the following categories:

- `misplaced`: subset of receptacles where object is found *before* housekeeping.
- `correct`: subset of receptacles where object is found *after* housekeeping.




(a) Object Category Agreement (b) High-Level Category Agreement

Fig. 2: Analysis of agreement between reviewer ratings in the Housekeep human rearrangement preferences dataset.

- `implausible`: subset of receptacles where object is unlikely to be found either in a clean or an untidy house.

We also ask each participant to rank receptacles classified under `misplaced` and `correct`. For example, given a can of food, someone may prefer placing it in kitchen cabinets while others will rank pantry over the kitchen cabinet.

For each object-room pair ($268 \times 17$), we collect 10 human annotations. We collect human annotations through multiple batches of smaller annotation tasks. In a single annotation task, we ask participants to classify-then-rank receptacles for 10 randomly sampled object-room pairs. On average a participant took 21 minutes to complete one annotation task. Overall, participants spent 1633 hours doing our study. Appendix C provides more details about the instructions page, user interface, training videos, and FAQs provided in the beginning of the task.

**Agreement analysis.** We evaluate the quality of our human annotations, using the Fleiss' kappa (`FK`) metric [20], which is widely used to assess the reliability of agreement between raters when classifying items. Recall that we collect 10 annotations to classify receptacles for each object-room pair into `correct`, `misplaced`, or `incompatible` bins. In Figure 2a, we report `FK` agreement per object across all room-receptacle pairs ($269 \times 128$) after keeping 8/10 annotations with the highest inter-human agreement. We use the agreement ranges proposed by [34] to interpret the `FK` scores. We also show agreement when combining the `misplaced` and `implausible` categories. Figure 2a demonstrates about 90% of our collected data has fair to moderate agreement between annotators. Figure 2b shows the mean agreement for high-level semantic categories. The agreement is higher for sporting, tool, and stationery categories because they go to specific places (office desks, garage, etc). The agreement is low for objects like fruits, medicines, packaged foods because people differ in where they like to keep these objects (packaged food can go in cabinets, shelves, kitchen counters, refrigerators). Overall, these results indicate that our data defines a high-quality source of ground truth rearrangement preferences agreed upon by the majority of annotators.

### 3.3 Episodes

Each Housekeep episode is created by instantiating 7-10 objects within a scene, out of which 3-5 objects are misplaced and the remaining are placed correctly. Next, we concretely define the notions of *correct* and *misplaced* objects. For a given scene, let $\mathcal{R}$ be the set of receptacles available, and $\mathcal{O}$ be the set of all the objects which could be instantiated on these. Given an object $o \in \mathcal{O}$, let $c_{or}$, $m_{or}$ respectively be the ratio of annotators who placed receptacle $r \in \mathcal{R}$ in `correct` and `misplaced` bins respectively. We call an object *correctly placed* if $c_{or} > 0.5$, and *misplaced* if $m_{or} > 0.5$, where both cannot be simultaneously true.

**Splits**: We create three non-overlapping sets of objects – `seen` (fork, gloves, etc.), `val-unseen` (chopping board, dishtowel, etc.), and `test-unseen` (banana, scissors, etc.). `seen`, `val-unseen` and `test-unseen` contains 8, 2 and 9 high-level object categories respectively. Note that *only* 40% of all objects are provided for training, making Housekeep a strong benchmark to test generalization to unseen objects.

We also split the 14 scenes into `train`, `val` and `test` with 8:2:4 scenes each respectively. We provide five different splits to test agents on a wide array of commonsense reasoning and rearrangement capabilities.

- **train**: 9K episodes with `seen` objects and `train` scenes
- **val-seen**: 200 episodes with `seen` objects and `val` scenes
- **val-unseen**: 200 episodes with `unseen` objects and `val` scenes
- **test-seen**: 800 episodes with `seen` objects and `test` scenes
- **test-unseen**: 800 episodes with `unseen` objects and `test` scenes

More details on episode statistics, and generation are in Appendix D.

### 3.4 Evaluation

We evaluate agents in three different dimensions of rearrangement quality, efficiency, and exploration. All metrics are reported per episode and then aggregated across multiple episodes to report averages and standard errors. While we only describe these metrics informally here, a more nuanced discussion with formal definitions for these can be found in Appendix D.3.

***Metrics for Rearrangement.*** These metrics evaluate the relative change in the placement of objects between start and end states of the episode.

- **Episode Success (`ES`)**: Strict binary (*all* or *none*) metric that is one if and only if all objects (irrespective of whether initially misplaced or correctly placed) in the episode are correctly placed at the end of the episode.
- **Object Success (`OS`)**: Fraction of the objects placed correctly.
- **Soft Object Success (`SOS`)**: The ratio of reviewers that agree that an object is placed correctly.

- **Rearrange Quality (RQ)**: A normalized value in $[0, 1]$ (via mean reciprocal rank [14]) is given to each object-receptacle based on the ranking collected from human preferences, 0 is given if misplaced.

Metrics OS, SOS and RQ are averaged across objects that are *initially misplaced* or *ever picked up* by the agent during the episode.

***Exploration and Efficiency Metrics:*** We also study how well the agent explores an unseen environment as well as efficiency at rearranging objects.

- **Map Coverage (MC)**: The % of the navigable map area explored.
- **Misplaced Objects Coverage (MOC)**: The fraction of misplaced objects discovered. Agent discovers an object when it appears in FoV at any point.
- **Pick and Place Efficiency (PPE)**: The minimum number of picks and places required to solve the episode divided by the number of picks and places made by agent in the episode.

## 4 Methods

In this section, we describe our hierarchical baseline for the Housekeep benchmark. Our baseline breaks the multi-stage rearrangement into three natural components: a) exploration and mapping, b) planning, and c) navigation and rearrangement. The planning module communicates with all the other modules and determines what the agent does (explore or rearrange). Before we dive into the details of our baseline, we discuss some additional sensors that our baseline has access to.
**Additional Sensors**: In the Housekeep specification the agent operates from an RGBD sensor. However, to scope the problem and focus on the planning and commonsense reasoning we allow access to the following:

- *semantic* and *instance* sensor: Provides two pixel-wise masks aligned with egocentric RGB observations. The semantic segmentation mask maps every pixel to an object or receptacle category (*e.g.* bowl, cabinet). The instance mask maps every pixel to a unique instance ID, which helps to disambiguate between instances of the same object/receptacle category.
- *relationship* sensor: Given instance IDs of an object and a receptacle in the egocentric view, the relationship sensor predicts a binary value if the object is on top of the receptacle or not.
- *receptacle-room* map: Receptacles are static within a scene, so we also assume access to a mapping that provides us with the room name for any receptacle discovered (*e.g.* an oven maps to the kitchen).

In the future, these sensors can be easily swapped with their learned counterparts. [13,30] demonstrate it is possible to learn a segmentation sensor for indoor scenes, and [5] shows it is possible to learn to infer relationships between 3D objects.

### 4.1 Mapping and Exploration

**`Mapping:`** At the start of an episode, this module initializes an empty top-down allocentric map. As the agent navigates through the environment, it continuously updates the map at each step using egocentric observations and camera projection matrix. We further use the RGBD-aligned pixel-wise instance and semantic masks to localize objects and receptacles and update our allocentric map with them. Finally, the mapping module also keeps track of the room and relationship information of discovered objects and receptacles via the *relationship sensor* and known *receptacle-room map*.

**`Exploration:`** To discover misplaced objects as well as suitable receptacles to place them on, our exploration module aims to maximize the area on the map it has seen. This module only requires the hyperparameter $\mathsf{n}_e$ — the number of exploration steps — as input and executes low-level actions via the navigation module. We use frontier-based exploration [72] (FRT) for our main experiments, which iteratively visits unexplored frontiers, which are the edges between visited and unvisited space. We keep our implementation details same as those in [52].

### 4.2 Planning

Our planner communicates with all the modules to build a high-level rearrangement plan that the agent follows. It consists of:

**`Rearrange` submodule**: Stores a list of locations of discovered objects and receptacles. From this list, it produces a list of object-receptacle pairs indicating the order of rearrangements to perform. There are 3 key decisions the rearrange submodule needs to make to create this list: 1) what objects are misplaced, 2) what order to arrange misplaced objects, and 3) what receptacle to place each misplaced object on. It makes these decisions via a **`Ranker`** submodule which ranks potential object-receptacle pairings by modeling the joint distribution $\mathbb{P}(\text{receptacle}, \text{room}|\text{object})$. To solve (3), for a given object the agent picks the receptacle in the room with the highest joint probability. We model the joint distribution of the receptacle and room because the context of a receptacle will change based on the room. For example, a plate belongs on the counter in the kitchen, but not a counter in the bathroom. Section 4.3 describes how we compute $\mathbb{P}(\text{receptacle}, \text{room}|\text{object})$, and also how we solve (1). To solve (2), we evaluate 4 heuristic orderings which are described in Appendix G.2.

**`Planner` submodule**: At any given step, the planner decides to explore only if there are no more pending rearrangements. The agent explores for a fixed number of steps ($\mathsf{n}_e$). Intuitively, higher values of $\mathsf{n}_e$ will encourage the agent to explore the environment at the beginning of the episode whereas lower values of $\mathsf{n}_e$ will encourage the agent to rearrange as soon as a better receptacle is found. While exploring, the planner ensures that map and rearrange modules are synchronized at each step. At the end of the exploration phase, the planner uses the rank ($\mathsf{L}$)

module to update compatibility scores by considering newly discovered objects and receptacles. We provide the planner pseudocode in Algorithm 2.

**Navigation and Pick-Place**: Please see Appendix E for details.

### 4.3 Extracting Embodied Commonsense from LLMs

One of the main goals of Housekeep is to equip the agent with commonsense knowledge to reason about the compatibility of an object with different receptacles present across different rooms. Large Language Models (LLMs) trained on unstructured web-corpora have been shown to work well for several embodied AI tasks like navigation [25, 26, 28, 36, 43]. We study whether we can use LLMs to extract physical (embodied) common sense about how humans prefer to rearrange objects to tidy a house. For this, we build a ranking module (`L`) which takes as input a list of objects and a list of receptacles in rooms and then outputs a sequence of desired rearrangements based on which object receptacle pairings are most likely. We select the rearrangements that maximize $\mathbb{P}(\text{receptacle}, \text{room}|\text{object})$. We decompose computing this probability into a product of two probabilities:

- Object Room `[OR]` -- $\mathbb{P}(\texttt{room}|\texttt{object})$ : Generate compatibility scores for rooms for a given object.
- Object Room Receptacle `[ORR]` -- $\mathbb{P}(\texttt{receptacle}|\texttt{object}, \texttt{room})$: Generate compatibility scores for receptacles within a given room and for a given object.

Both of these are learned from the human rearrangement preferences dataset. From the compatibility scores in the `ORR` task, we first determine which objects in our list of objects are misplaced and which are correctly placed. To do this, we compute a hyperparameter $s_L$ — the score threshold — from our `val` episodes using a grid search. Receptacles whose scores are above $s_L$ for a given object-room pair are marked as correct, while those whose scores are below $s_L$ are marked as incorrect. We then treat this as a classification task and pick $s_L$ that maximizes the F1 score on the `val` episodes.

Next, to determine the ranking of receptacles for a given misplaced object, we use the probabilities from both the `OR` and `ORR` tasks. For a given object, we first rank the rooms in descending order of $\mathbb{P}(\text{room}|\text{object})$. Then, for each object-room pair in the ranked room list, we rank the *correct* receptacles in the room in descending order of $\mathbb{P}(\text{receptacle}|\text{object}, \text{room})$. Finally, we place the *incorrect* receptacles at the end of our list.

To learn the probability scores in the `OR` and `ORR` tasks, we start by extracting word embeddings from a pretrained RoBERTa LLM [40] of all objects, receptacles. We experiment with various contextual prompts [50,51] for extracting embeddings of paired room-receptacle (*e.g.* "`<receptacle> of <room>`") and object-room (*e.g.* "`<object> in <room>`") combinations. Next, we implemented the following 2 methods of using these embeddings to get the final compatibility scores:

**Zero-Shot Ranking via MLM (`ZS-MLM`).** Masked Language Modeling (MLM) is used extensively for pretraining LLMs [18, 40], which involves predicting a

masked word (*i.e.* `[mask]`) given the surrounding context words. This objective can be extended for zero-shot ranking using various contextual prompts. We use a frozen LLM to compute log-likelihood scores of prompts and use these scores to rank rooms and receptacles for the `OR` and `ORR` tasks. For `ORR`, we use the prompt "`in <room>, usually you put <object> <spatial-preposition> [mask]`" to rank receptacles given an object, a room, and a spatial preposition (*e.g.* in or on). For `OR`, we use the prompt "`in a household, it is likely that you can find <object> in the room called [mask]`".

**Finetuning by Contrastive Matching (`CM`).** Apart from using prompts in a zero-shot manner, we also train a 3-layered MLP on top of language embeddings generated by the LLM used in `ZS-MLM` and compute pairwise cosine similarity between any two embeddings. Embeddings are trained using objects from `seen` split. We train separate models for `ORR` and `OR`. For `ORR`, we match an object-room pair to the receptacle with the best average rank across annotators. We use contrastive loss [47] to promote similarity between an object-room pair and the matching receptacle. For `OR`, we match an object with all rooms that have at least one `correct` receptacle for it. In this case, we use the binary cross entropy (BCE) loss to handle multiple rooms per object.

We compare these ranking approaches with other baselines in Section 5.1. We provide training details of our ranking module in Appendix F.

## 5 Experiments

We first test whether LLMs can capture the embodied commonsense reasoning needed for planning in Housekeep. Then we deploy our modular agent equipped with this LLM-based planner to benchmark its ability to generalize to unseen environments cluttered with novel objects from seen (*i.e.* `test-seen`) and unseen (*i.e.* `test-unseen`) categories. Finally, we perform a thorough qualitative analysis of its failure modes and highlight directions for further improvements.

### 5.1 Language Models Capture Embodied Commonsense

**Methods.** We evaluate `CM` and `ZS-MLM` using RoBERTa [40] as our base LLM. We also compare these with GloVe-based [49] embeddings, and a baseline that randomly ranks rooms (for `OR` task) and receptacles (for `ORR` task).

**Evaluation.** We evaluate mean average precision (mAP) across objects to compare the ranked list of rooms/receptacles obtained from our ranking module to the list of rooms/receptacles deemed `correct` by the human annotators. Recall from section 3.3, for a given object, a receptacle

Table 1: We report mAP scores on train, and unseen objects splits of val and test for both `OR` and `ORR` matching tasks. The finetuning with `CM` objective is performed using objects *only* from train split

| | | ORR | | | OR | | |
|---|---|---|---|---|---|---|---|
| # | **Method** | train | val-u | test-u | train | val-u | test-u |
| 1 | RoBERTa+CM | 0.81 | **0.79** | **0.81** | **1.0** | **0.65** | 0.65 |
| 2 | GloVe+CM | **0.88** | 0.76 | 0.76 | **1.0** | **0.65** | **0.66** |
| 3 | ZS-MLM | 0.43 | 0.46 | 0.42 | 0.51 | 0.54 | 0.52 |
| 4 | Random | 0.47 | 0.47 | 0.46 | 0.58 | 0.52 | 0.59 |

Table 2: Results using our modular baseline on Housekeep `test-seen` and `test-unseen` splits. `OR`: Oracle, `LM`: LLM-based ranking, `FT`: Frontier exploration. `GLV`: GloVe

| | | Modules | | Rearrange | | Soft-Score | | Explore | | Efficiency |
|---|---|---|---|---|---|---|---|---|---|---|
| | # | Rank | Explore | ES ↑ | OS ↑ | SOS ↑ | RQ ↑ | MC ↑ | OC ↑ | PPE ↑ |
| t-seen | 1 | OR | OR | 1.00 ± 0.00 | 1.00 ± 0.00 | 0.65 ± 0.00 | 0.63 ± 0.00 | – | 1.00 ± 0.00 | 1.00 ± 0.00 |
| | 2 | OR | FTR | 0.35 ± 0.02 | 0.64 ± 0.01 | 0.49 ± 0.01 | 0.41 ± 0.01 | 73 ± 1 | 0.73 ± 0.01 | 1.00 ± 0.00 |
| | 3 | LM | OR | 0.04 ± 0.01 | 0.44 ± 0.01 | 0.46 ± 0.00 | 0.30 ± 0.01 | – | 1.00 ± 0.00 | 0.57 ± 0.01 |
| | 4 | LM | FTR | 0.01 ± 0.00 | 0.30 ± 0.01 | 0.39 ± 0.00 | 0.19 ± 0.01 | 77 ± 1 | 0.76 ± 0.01 | 0.41 ± 0.01 |
| | 5 | GLV | FTR | 0.01 ± 0.00 | 0.29 ± 0.01 | 0.36 ± 0.00 | 0.19 ± 0.01 | 71 ± 1 | 0.73 ± 0.01 | 0.39 ± 0.01 |
| t-unseen | 6 | OR | OR | 1.00 ± 0.00 | 1.00 ± 0.00 | 0.64 ± 0.00 | 0.61 ± 0.00 | – | 1.00 ± 0.00 | 1.00 ± 0.00 |
| | 7 | OR | FTR | 0.35 ± 0.02 | 0.65 ± 0.01 | 0.49 ± 0.01 | 0.40 ± 0.01 | 74 ± 1 | 0.74 ± 0.01 | 1.00 ± 0.00 |
| | 8 | LM | OR | 0.02 ± 0.00 | 0.32 ± 0.01 | 0.42 ± 0.00 | 0.20 ± 0.01 | – | 1.00 ± 0.00 | 0.42 ± 0.01 |
| | 9 | LM | FTR | 0.01 ± 0.00 | 0.23 ± 0.01 | 0.36 ± 0.00 | 0.14 ± 0.01 | 73 ± 1 | 0.74 ± 0.01 | 0.35 ± 0.01 |
| | 10 | GLV | FTR | 0.00 ± 0.00 | 0.23 ± 0.01 | 0.34 ± 0.00 | 0.15 ± 0.01 | 72 ± 1 | 0.74 ± 0.01 | 0.26 ± 0.01 |

is considered `correct` when at least 6 annotators vote for it, and a room is considered `correct` if it has at least one `correct` receptacle within it. Higher AP score indicates `correct` items are likely to ranked higher than the `incorrect` items.

**Results.** Table 1 shows that `RoBERTa+CM` outperforms `ZS-MLM` by a large margin even when fintuned on a relatively small-sized training set (~40% of total data, see Section 3.4). We find good transfer of results from `val` to `test` splits by `RoBERTa+CM` method on both tasks demonstrating the better generalization capabilities of LLMs. On the other hand, `GloVe+CM` does not seem to transfer well for the `ORR` task. Also, `ZS-MLM` performs worse than the `Random` baseline. We found that predictions of ZS-MLM baseline are biased towards certain receptacles (e.g. chair, and carpet are in top-4 most frequent choices). This bias is frequently not aligned with human preferences. We hypothesize this is likely an artifact of the original training data. Finally, notice that `Random` baseline performs relatively well on room-matching (`OR`) task, which is expected since there are ample of rooms with at least one correct receptacle for any given object.

### 5.2 Main Results for Housekeep

We use `RoBERTa+CM` as scoring function in `Ranker` module to continuously rerank (thus replan) discovered rooms and receptacles while exploring Housekeep episodes.

**Oracle Modules.** We show oracle agent's performance, by swappping `Ranker` and `Explore` modules with their oracle (perfect) counterparts. Oracle ranker uses the ground truth human preferences to rank the objects and receptacles found. Oracle exploration gives a complete map of the environment, *i.e.* agent knows all objects, receptacles and their respective locations.

**Upper Bounds.** In Table 2, we show results on both `test-seen` and `test-unseen` splits. Rows 1, 6 with oracle ranking and exploration denote the upper bounds achievable across all metrics. Note that Soft Object Success (`SOS`) and Rear-

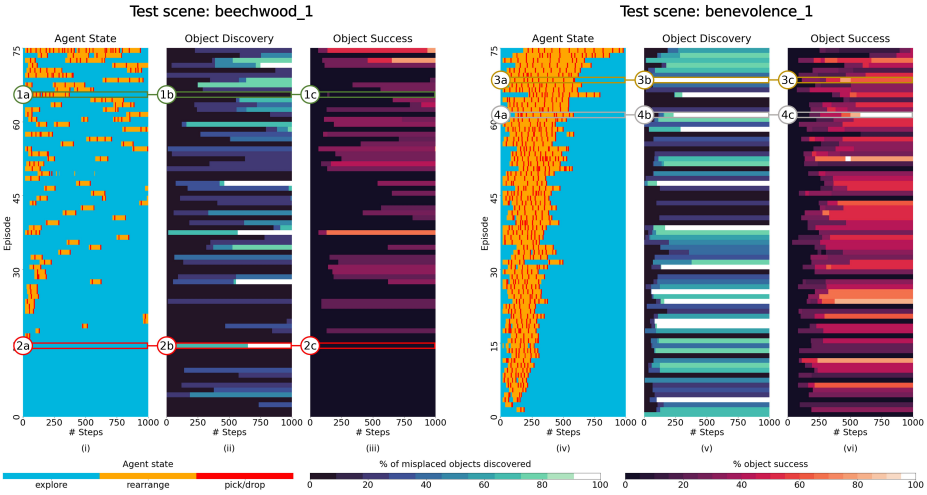



Fig. 3: Visually depicting agent's progress on 75 randomly-sampled episodes from two test scenes, `beechwood_1` and `benevolence_1`. Plots (i) and (iv) depict Agent's state, (ii) and (v) show % of objects discovered, (iii) and (vi) show % object success, and x-axis is the timestep. All 3 plots of same scenes are aligned, *i.e.* show same episodes on y-axis.

rangement Quality (`RQ`) are not perfect since human agreement across correct receptacles is not 100%.

**Frontier Exploration, Full baseline.** Using Frontier exploration (rows 1,2), `OS` drops by 47%. This drop in performance signifies the importance of task-driven exploration needed for Housekeep to find `misplaced` objects or `correct` receptacles quickly. Finally, we evaluate the fully non-oracle baseline (row 4) which achieves a 30% object success rate. From rows 4 and 9, we see that `OS` drops by 7%, but `SOS` drops only by 3% across seen vs unseen objects, which demonstrates some level of generalization capability to unseen environments. We also evaluate our baseline agent with a GloVe-based ranker (rows 5, 10) and observe similar `OS` performance to the LLM ranker.

We put additional experiments analyzing the effect of exploration steps ($\mathtt{n}_e$), exploration strategies in Appendix G, and qualitative results in Appendix H.

### 5.3 Qualitative Analysis

Figure 3 visually depicts the baseline agent's progress across episodes on two test scenes. Agent State plots show the *module currently being executed*: `explore` (blue), `rearrange` (orange), or `pick/place` (red). Object Discovery plots show the *percentage of misplaced objects discovered* until any given time step. Object Success plots show the *object success* at any given time step. Dark to light shade corresponds to an increasing number of misplaced objects found/increasing object success. Each row corresponds to one episode, and the x-axis denotes time step.

**Agent cannot classify discovered objects as misplaced.** For beechwood_1, row 2a in **(i)** and rows below it show that in approximately a quarter of the episodes, the agent only explores and never rearranges. The corresponding row 2b in **(ii)** tells us that all the misplaced objects were discovered by $\approx 700$ time steps. From row 2a and 2b, we can conclude that the ranking module fails to identify objects as misplaced even after discovering them.

**Agent rearranges incorrect objects.** Next, looking at orange regions in row 1a, we know that the agent rearranges several objects. However, the corresponding row 1b in **(ii)** is fully black, indicating that the agent discovered 0% of misplaced objects. This means that the reasoning module misidentifies correctly placed objects as misplaced and asks the agent to rearrange them. Moreover, the exploration module fails to locate misplaced objects.

**Scene layouts affect object discovery.** Our agent explores differently in different scene layouts. In Figure 3, the agent discovers misplaced objects much more quickly in benevolence_1 than in beechwood_1, and correctly rearranges a higher fraction of them. Rows 3a and 3b show this trend – all objects are discovered within the first 200 steps of the episode in stark contrast to beechwood_1 episodes. Row 4c even shows an episode with 100% object success. This is explained by the fact that benevolence_1 is a smaller home with just one partitioning wall (4 rooms) versus beechwood_1 (8 rooms), making exploration and object discovery easier. We provide top-down maps of both scenes in Appendix H.1.

## 6 Conclusion

We presented the Housekeep benchmark to evaluate commonsense reasoning in the home for embodied AI. We collected a dataset of human preferences of where objects go in tidy and untidy houses, and used it to generate episodes and evaluate agent performance. Then we proposed a modular baseline that plans using commonsense reasoning extracted from a large language model. Housekeep is a challenging task, and the overall episode success rate remains low despite the use of additional sensors (*e.g.* segmentation, relationship) needed for planning and commonsense reasoning. Two areas of improvement are exploration module and reasoning module. A learned exploration module can visit areas that get cluttered more frequently, and optimize object coverage instead of map coverage. Improving the reasoning module's recall and precision at identifying misplaced objects can increase performance on our task. Finally, replacing additional sensors related with their learned counterparts will make baselines more realistic.

**Acknowledgements** We thank the Habitat team for their support. The Georgia Tech effort was supported in part by NSF, AFRL, DARPA, ONR YIPs, ARO PECASE, Amazon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government, or any sponsor.

## References


1. Abdo, N., Stachniss, C., Spinello, L., Burgard, W.: Robot, organize my shelves! tidying up objects by predicting user preferences. 2015 IEEE International Conference on Robotics and Automation (ICRA) (2015)
2. Agrawal, H., Chandrasekaran, A., Batra, D., Parikh, D., Bansal, M.: Sort story: Sorting jumbled images and captions into stories. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (2016)
3. Anderson, P., Chang, A., Chaplot, D.S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., et al.: On evaluation of embodied navigation agents. arXiv preprint arXiv:1807.06757 (2018)
4. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I.D., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018 (2018)
5. Armeni, I., He, Z., Zamir, A.R., Gwak, J., Malik, J., Fischer, M., Savarese, S.: 3d scene graph: A structure for unified semantics, 3d space, and camera. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019 (2019)
6. Batra, D., Chang, A.X., Chernova, S., Davison, A.J., Deng, J., Koltun, V., Levine, S., Malik, J., Mordatch, I., Mottaghi, R., Savva, M., Su, H.: Rearrangement: A challenge for embodied ai (2020)
7. Batra, D., Gokaslan, A., Kembhavi, A., Maksymets, O., Mottaghi, R., Savva, M., Toshev, A., Wijmans, E.: Objectnav revisited: On evaluation of embodied agents navigating to objects. arXiv preprint arXiv:2006.13171 (2020)
8. Bhagavatula, C., Bras, R.L., Malaviya, C., Sakaguchi, K., Holtzman, A., Rashkin, H., Downey, D., Yih, W., Choi, Y.: Abductive commonsense reasoning. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020 (2020)
9. Bisk, Y., Zellers, R., LeBras, R., Gao, J., Choi, Y.: PIQA: reasoning about physical commonsense in natural language. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020 (2020)
10. Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., Choi, Y.: COMET: Commonsense transformers for automatic knowledge graph construction. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)
11. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
12. Calli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.M.: The ycb object and model set: Towards common benchmarks for manipulation research. In: 2015 international conference on advanced robotics (ICAR). IEEE (2015)

13. Cartillier, V., Ren, Z., Jain, N., Lee, S., Essa, I., Batra, D.: Semantic mapnet: Building allocentric semanticmaps and representations from egocentric views. arXiv preprint arXiv:2010.01191 (2020)
14. Craswell, N.: Mean reciprocal rank. In: Encyclopedia of Database Systems (2009)
15. Crowston, K.: Amazon mechanical turk: A research tool for organizations and information systems scholars. In: Shaping the future of ict research. methods and approaches (2012)
16. Daruna, A., Liu, W., Kira, Z., Chernova, S.: Robocse: Robot common sense embedding (2019)
17. Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., Batra, D.: Embodied question answering. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018 (2018)
18. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (2019)
19. Ehsani, K., Han, W., Herrasti, A., VanderBilt, E., Weihs, L., Kolve, E., Kembhavi, A., Mottaghi, R.: ManipulaTHOR: A framework for visual object manipulation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2021)
20. Fleiss, J., et al.: Measuring nominal scale agreement among many raters. Psychological Bulletin **76**(5) (1971)
21. Gan, C., Schwartz, J.I., Alter, S., Schrimpf, M., Traer, J., de Freitas, J.L., Kubilius, J., Bhandwaldar, A., Haber, N., Sano, M., Kim, K., Wang, E., Mrowca, D., Lingelbach, M., Curtis, A., Feigelis, K.T., Bear, D.M., Gutfreund, D., Cox, D., DiCarlo, J.J., McDermott, J., Tenenbaum, J., Yamins, D.L.K.: Threedworld: A platform for interactive multi-modal physical simulation. NeurIPS **abs/2007.04954** (2020)
22. Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., Farhadi, A.: IQA: visual question answering in interactive environments. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018 (2018)
23. Granroth-Wilding, M., Clark, S.: What happens next? event prediction using a compositional neural network model. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA (2016)
24. Habitat: Habitat Challenge (2021), https://aihabitat.org/challenge/2021/
25. Hill, F., Mokra, S., Wong, N., Harley, T.: Human instruction-following with deep reinforcement learning via transfer-learning from text. ArXiv **abs/2005.09382** (2020)
26. Hong, Y., Wu, Q., Qi, Y., Rodriguez-Opazo, C., Gould, S.: A recurrent vision-and-language BERT for navigation. In: ECCV (2021)
27. Hu, X., Yin, X., Lin, K., Wang, L., Zhang, L., Gao, J., Liu, Z.: Vivo: Surpassing human performance in novel object captioning with visual vocabulary pre-training. ArXiv **abs/2009.13682** (2020)
28. Huang, W., Abbeel, P., Pathak, D., Mordatch, I.: Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. ArXiv **abs/2201.07207** (2022)
29. Jasmine, C., Shubham, G., Achleshwar, L., Leon, X., Kenan, D., Xi, Z., F, Y.V.T., Himanshu, A., Thomas, D., Matthieu, G., Jitendra, M.: Abo: Dataset and bench-

marks for real-world 3d object understanding. arXiv preprint arXiv:2110.06199 (2021)
30. Jiang, J., Zheng, L., Luo, F., Zhang, Z.: Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. arXiv preprint arXiv:1806.01054 (2018)
31. Jiang, Y., Lim, M., Saxena, A.: Learning object arrangements in 3d scenes using human context. In: Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012 (2012)
32. Kapelyukh, I., Johns, E.: My house, my rules: Learning tidying preferences with graph neural networks. In: CoRL (2021)
33. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon, D., Zhu, Y., Gupta, A., Farhadi, A.: AI2-THOR: An Interactive 3D Environment for Visual AI. arXiv (2017)
34. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. biometrics (1977)
35. Levesque, H.J., Davis, E., Morgenstern, L.: The winograd schema challenge. In: KR (2011)
36. Li, S., Puig, X., Du, Y., Wang, C., Akyürek, E., Torralba, A., Andreas, J., Mordatch, I.: Pre-trained language models for interactive decision-making. ArXiv **abs/2202.01771** (2022)
37. Li, X., Yin, X., Li, C., Hu, X., Zhang, P., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., Gao, J.: Oscar: Object-semantics aligned pre-training for vision-language tasks. In: ECCV (2020)
38. Liu, W., Bansal, D., Daruna, A., Chernova, S.: Learning Instance-Level N-Ary Semantic Knowledge At Scale For Robots Operating in Everyday Environments. In: Proceedings of Robotics: Science and Systems (2021)
39. Liu, W., Paxton, C., Hermans, T., Fox, D.: Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects. arXiv preprint arXiv:2110.10189 (2021)
40. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 (2019)
41. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada (2019)
42. Lu, K., Grover, A., Abbeel, P., Mordatch, I.: Pretrained transformers as universal computation engines. ArXiv **abs/2103.05247** (2021)
43. Majumdar, A., Shrivastava, A., Lee, S., Anderson, P., Parikh, D., Batra, D.: Improving vision-and-language navigation with image-text pairs from the web. ArXiv **abs/2004.14973** (2020)
44. Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P., Allen, J.: A corpus and cloze evaluation for deeper understanding of commonsense stories. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2016)
45. Moudgil, A., Majumdar, A., Agrawal, H., Lee, S., Batra, D.: Soat: A scene-and object-aware transformer for vision-and-language navigation. Advances in Neural Information Processing Systems **34** (2021)

46. Narasimhan, M., Wijmans, E., Chen, X., Darrell, T., Batra, D., Parikh, D., Singh, A.: Seeing the un-scene: Learning amodal semantic maps for room navigation. CoRR **abs/2007.09841** (2020)
47. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
48. Padmakumar, A., Thomason, J., Shrivastava, A., Lange, P., Narayan-Chen, A., Gella, S., Piramithu, R., Tur, G., Hakkani-Tür, D.Z.: Teach: Task-driven embodied agents that chat. ArXiv **abs/2110.00534** (2021)
49. Pennington, J., Socher, R., Manning, C.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014)
50. Petroni, F., Lewis, P., Piktus, A., Rocktäschel, T., Wu, Y., Miller, A.H., Riedel, S.: How context affects language models' factual predictions. In: Automated Knowledge Base Construction (2020)
51. Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.: Language models as knowledge bases? In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (2019)
52. Ramakrishnan, S.K., Jayaraman, D., Grauman, K.: An exploration of embodied visual exploration (2020)
53. Research, G.: Google Scanned Objects. `https://app.ignitionrobotics.org/GoogleResearch/fuel/collections/Google%20Scanned%20Objects` (2020), [Online; accessed Feb-2022]
54. Roberts, A., Raffel, C., Shazeer, N.: How much knowledge can you pack into the parameters of a language model? In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP) (2020)
55. robotics, F.: Fetch. `http://fetchrobotics.com/` (2020)
56. Sakaguchi, K., Le Bras, R., Bhagavatula, C., Choi, Y.: Winogrande: An adversarial winograd schema challenge at scale. In: AAAI (2020)
57. Salganik, M.J.: Bit by Bit: Social Research in the Digital Age. Open review edition edn. (2017)
58. Sap, M., Bras, R.L., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N.A., Choi, Y.: ATOMIC: an atlas of machine commonsense for if-then reasoning. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019 (2019)
59. Sap, M., Rashkin, H., Chen, D., Le Bras, R., Choi, Y.: Social IQa: Commonsense reasoning about social interactions. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (2019)
60. Savva, M., Malik, J., Parikh, D., Batra, D., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V.: Habitat: A platform for embodied AI research. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019 (2019)
61. Shen, B., Xia, F., Li, C., Martín-Martín, R., Fan, L., Wang, G., Buch, S., D'Arpino, C., Srivastava, S., Tchapmi, L.P., et al.: igibson, a simulation environment for interactive tasks in large realistic scenes. arXiv preprint arXiv:2012.02924 (2020)

62. Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., Fox, D.: ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020 (2020)
63. Srivastava, S., Li, C., Lingelbach, M., Mart'in-Mart'in, R., Xia, F., Vainio, K., Lian, Z., Gokmen, C., Buch, S., Liu, C.K., Savarese, S., Gweon, H., Wu, J., Fei-Fei, L.: Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In: CoRL (2021)
64. Szot, A., Clegg, A., Undersander, E., Wijmans, E., Zhao, Y., Turner, J., Maestre, N., Mukadam, M., Chaplot, D.S., Maksymets, O., et al.: Habitat 2.0: Training home assistants to rearrange their habitat. Advances in Neural Information Processing Systems **34** (2021)
65. Taniguchi, A., Isobe, S., Hafi, L.E., Hagiwara, Y., Taniguchi, T.: Autonomous planning based on spatial concepts to tidy up home environments with service robots. Advanced Robotics **35** (2021)
66. Thomason, J., Murray, M., Cakmak, M., Zettlemoyer, L.: Vision-and-dialog navigation. CoRL (2019)
67. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA (2017)
68. Wang, W., Bao, H., Dong, L., Wei, F.: Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. ArXiv **abs/2111.02358** (2021)
69. Wani, S., Patel, S., Jain, U., Chang, A.X., Savva, M.: Multion: Benchmarking semantic map memory using multi-object navigation. In: NeurIPS (2020)
70. Weihs, L., Deitke, M., Kembhavi, A., Mottaghi, R.: Visual room rearrangement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (2021)
71. Wijmans, E., Datta, S., Maksymets, O., Das, A., Gkioxari, G., Lee, S., Essa, I., Parikh, D., Batra, D.: Embodied question answering in photorealistic environments with point cloud perception. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019 (2019)
72. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: cira. vol. 97 (1997)
73. Zellers, R., Bisk, Y., Farhadi, A., Choi, Y.: From recognition to cognition: Visual commonsense reasoning. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019 (2019)
74. Zellers, R., Bisk, Y., Schwartz, R., Choi, Y.: SWAG: A large-scale adversarial dataset for grounded commonsense inference. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (2018)
75. Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., Choi, Y.: HellaSwag: Can a machine really finish your sentence? In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)
76. Zhou, B., Khashabi, D., Ning, Q., Roth, D.: "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (2019)
77. Çalli, B., Singh, A., Walsman, A., Srinivasa, S., Abbeel, P., Dollar, A.: The ycb object and model set: Towards common benchmarks for manipulation research. 2015 International Conference on Advanced Robotics (ICAR) (2015)