TIDEE: Tidying Up Novel Rooms using Visuo-Semantic Commonsense Priors

Gabriel Sarch[1*], Zhaoyuan Fang[1], Adam W. Harley[1], Paul Schydlo[1],

Michael J. Tarr[1], Saurabh Gupta[2], and Katerina Fragkiadaki[1]

[1] Carnegie Mellon University
[2] University of Illinois at Urbana-Champaign

[*]Correspondence to gsarch@andrew.cmu.edu

Abstract. We introduce TIDEE, an embodied agent that tidies up a disordered scene based on learned commonsense object placement and room arrangement priors. TIDEE explores a home environment, detects objects that are out of their natural place, infers plausible object contexts for them, localizes such contexts in the current scene, and repositions the objects. Commonsense priors are encoded in three modules: i) visuo-semantic detectors that detect out-of-place objects, ii) an associative neural graph memory of objects and spatial relations that proposes plausible semantic receptacles and surfaces for object repositions, and iii) a visual search network that guides the agent's exploration for efficiently localizing the receptacle-of-interest in the current scene to reposition the object. We test TIDEE on tidying up disorganized scenes in the AI2THOR simulation environment. TIDEE carries out the task directly from pixel and raw depth input without ever having observed the same room beforehand, relying only on priors learned from a separate set of training houses. Human evaluations on the resulting room reorganizations show TIDEE outperforms ablative versions of the model that do not use one or more of the commonsense priors. On a related room rearrangement benchmark that allows the agent to view the goal state prior to rearrangement, a simplified version of our model significantly outperforms a top-performing method by a large margin. Code and data are available at the project website: `https://tidee-agent.github.io/`.

1 Introduction

For robots to operate in home environments and assist humans in their daily lives, they need to be more than step-by-step instruction followers: they need to proactively take action in circumstances that violate expectations, priors, and norms, and effectively interpret incomplete or noisy instructions by human users. Consider Figure 1. A robot should realize the remote is out-of-place, should be able to infer alternative plausible repositions, and tidy-up the scene by rearranging the objects to their regular locations. Such understanding would also permit the robot to follow incomplete instructions from human users, such as *"put the*

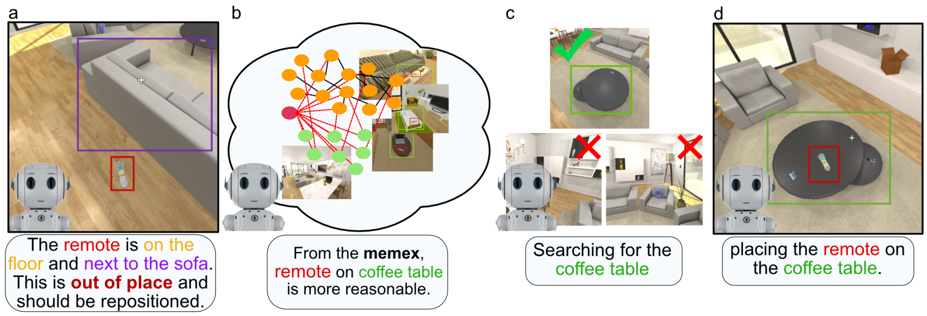

Fig. 1. TIDEE is an embodied agent that tidies up disorganized scenes using commonsense knowledge of object placements and room arrangements. (a) It explores the scene to detect out-of-place (OOP) objects (in this case the remote control). (b) It then infers plausible receptacles (the coffee table) through graph inference over a neural graph memory of objects and relations. (c-d) It then searches for the inferred receptacle (the coffee table) guided by a visual search network and repositions the object.

remote away". For this, a robot needs to have commonsense knowledge regarding contextual, object-object, and object-room spatial relations.

What is the form of this commonsense knowledge and how can it be acquired? There are two sources of commonsense knowledge: i) communication of such knowledge via natural language, for example, *"the lamp should be placed on the bed stand"*, and ii) acquisition of such knowledge via visually observing the world and encoding statistical relationships between objects and places. These two sources are complementary. Commonsense in natural language is easy to specify and modify through instruction, while commonsense through visual observation is scalable and often more expressive. Consider, for example, tall yellow IKEA lamps that are often placed on the floor, while shorter lamps are usually placed on bed stands and are appropriately centered and oriented towards the bed. In this example, object contextual relationships depend on more than the category label "lamp"; they depend on sub-categorical information, which is easily encoded in the visual features of the objects [25].

We introduce Teachable Interactive Decluttering Embodied Explorer (TIDEE), which combines semantic and visual commonsense knowledge with embodied components to tidy up disorganized home environments it has never seen before, from raw RGB-D input. TIDEE explores a home environment to detect objects that are not in their normal locations (that therefore need to be repositioned), as shown in Figure 1(a). When an out-of-place (OOP) object is detected, TIDEE infers plausible receptacles for the object to be placed onto, through graph inference over the union of a neural memory graph of objects and spatial relations and the scene graph of the room at hand (Figure 1(b)). It then actively explores the scene to find instances of the predicted receptacle category guided by a visual search network, and repositions the detected out-of-place objects

(Figure 1(c-d)).[1] TIDEE uses both visual features and semantic information to encode commonsense knowledge. This knowledge is encoded in the weights of the out-of-place detectors, the neural memory graph weights, and the visual search network weights, and is learned end-to-end to optimize objectives of the rearrangement task, such as classifying out-of-place objects, inferring plausible repositions, and efficiently locating an object of interest. To the best of our knowledge, this is the first work that attempts to tidy up novel room environments directly from pixel and depth input, without any explicit instructions for object placements, relying instead on learned prior knowledge to solve the task.

We test TIDEE in tidying up kitchens, living rooms, bathrooms and bedrooms in the AI2THOR simulation environment [23]. We generate untidy scenes by applying random forces that push or pull objects within each room. We show that human evaluators prefer TIDEE's rearrangements more often than those obtained by baselines or ablative versions of our model that do not use semantics for out-of-place detection, do not use a learnable graph memory (defaulting instead to most common placement), or do not have neural guidance during object search. We further show that TIDEE can be adapted to respect preferences of users by fine-tuning its out-of-place visuo-semantic object classifier based on individual instructions. Finally, we test a reduced version of TIDEE on the recent scene rearrangement benchmark [3, 37], where an AI agent is tasked to reposition the objects to bring the scene to a desirable target configuration. TIDEE outperforms the current state of the art. We attribute TIDEE's excellent performance to the modular organization of its architecture and the object-centric scene representation TIDEE uses to reason about rearrangements.

2 Related Work

Embodied AI. The development of learning-based embodied AI agents has made significant progress across a wide variety of tasks, including: scene rearrangement [3,17,37], object-goal navigation [1,6,8,19,40,42], point-goal navigation [1,19,30,31,39], scene exploration [7,10], embodied question answering [12, 18], instructional navigation [2,34], object manipulation [14,43], home task completion with explicit instructions [27,34,35], active visual learning [9,15,20,38], and collaborative task completion with agent-human conversations [29]. While these works have driven much progress in embodied AI, ours is the first agent to tackle the task of tidying up rooms, which requires commonsense reasoning about whether or not an object is out of place, and inferring where it belongs in the context of the room. Progress in embodied AI has been accelerated tremendously through the availability of high visual fidelity simulators, such as, Habitat [31], GibsonWorld [33], ThreeDWorld [16], and AI2THOR [23]. Our work builds upon AI2THOR by relying on the (approximate) dynamic manipulation the simulator enables for household objects.

[1] We follow the terminology from AI2THOR [23] and define a receptacle as a type of object that can contain or support other objects. Sinks, refrigerators, cabinets, and tabletops are some examples of receptacles.

Representing commonsense knowledge regarding object spatial relations. Visual commonsense knowledge is often represented in terms of a knowledge graph, namely, a graph of visual entity nodes (objects, parts, attributes) where edge types represent pairwise relationships between entities. Knowledge graphs have been successfully used in visual classification and detection [11, 26], zero-shot classification of images [36], object goal navigation [42], and image retrieval [22].

Closest to our work is the work of Yang et. al. [42] where a knowledge graph is used to help an agent navigate to semantic object goals. While in the knowledge graph of Yang et. al. [42] each node stands for an object category described by its semantic embedding, in our case each node is an object instance described by both semantic and visual features, similar to the earlier work of Malisiewicz and Efros on visual Memex [25]. Moreover, we consider tidying up rooms, where navigation to semantic goals is one submodule of what the agent needs to do. Lastly, while [42] maps images to actions directly trained with reinforcement learning, and graph indexing provides simply an additional embedding to concatenate to the agent's state, our model is modular and hierarchical, using a "theory" of out-of-place objects, inferring regular object placements, exploration to localize placements in the scene, and then taking actions to achieve the inferred object re-arrangement. We show that TIDEE outperforms non-modular image-to-action mapping agents in the scene re-arrangement benchmark in Section 4.5.

3 Teachable Interactive Decluttering Embodied Explorer (TIDEE)

The architecture of TIDEE is illustrated in Figure 2. The agent navigates a home environment and receives RGB-D images at each time step alongside egomotion information. We consider both groundtruth depth and egomotion, as well as noisy versions of both, and estimated depth in our experimental section. The agent builds geometrically consistent spatial 2D and 3D maps of the environment by fusing RGB-D input, following prior works [7] (Section 3.1). TIDEE detects objects and classifies them as in or out-of-place (OOP) using a combination of visual and semantic features (Section 3.2). When an OOP object is detected, the agent infers plausible object context (i.e., plausible receptacle categories for the OOP object to be repositioned on) through inference over a memory graph of objects and relations (Memex) and the current scene graph (Section 3.3). The agent then searches the current scene to find instances of the receptacle category and a visual search network guides its exploration by proposing locations in the scene to visit (Section 3.4). Once the receptacle is detected, the agent places the OOP object on it. Navigation actions move the agent in discrete steps. For picking up and placing objects, the agent must specify an object to interact with via a relative coordinate (x, y) in the (ego-centric) frame.

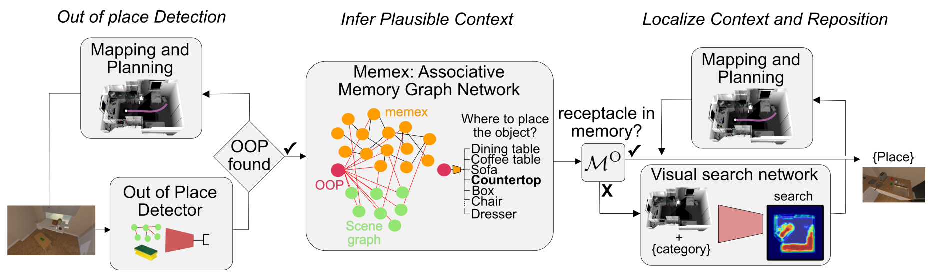

Fig. 2. Architecture of TIDEE. TIDEE explores the scene, detects objects and classifies whether they are in-place or out-of-place. If an object is out-of-place, TIDEE uses graph inference in its joint external graph memory and scene graph to infer plausible receptacle categories. It then explores the scene guided by a visual search network that suggests where instances of a receptacle category may be found, given the scene spatial semantic map. TIDEE iterates the steps above until it cannot detect any more OOP objects, in which case it concludes that the room has been tidied up.

3.1 Background: Semantic 3D mapping

TIDEE builds 3D semantic maps of the home environment it visits augmented with 3D object detection centroids. These maps are used to infer spatial relations among objects and to guide exploration to objects-of-interest. Specifically, TIDEE maintains two spatial visual maps of the environment that it updates at each time step from the input RGB-D stream, similar to previous works [8]: i) a 2D overhead occupancy map $\mathbf{M}_t^{\mathrm{2D}} \in \mathbb{R}^{H \times W}$ and, ii) a 3D occupancy and semantics map $\mathbf{M}_t^{\mathrm{3D}} \in \mathbb{R}^{H \times W \times D \times K}$, where K is the number of semantic object categories; we use $K = 116$. The $\mathbf{M}^{\mathrm{2D}}$ maps is used for exploration and navigation in the environment. More details on our exploration and planning strategy can be found in the supplementary.

We detect objects from K semantic object categories in each input RGB image using the state-of-the-art d-DETR detector [45], pretrained on the MS-COCO datasets [24] and finetuned on images from the AI2THOR training houses. We obtain 3D object centroids by using the depth input to map detected 2D object bounding boxes into a 3D box centroids. We add these in the 3D semantic map with one channel per semantic class, similar to Chaplot et. al. [9], but in 3D as opposed to a 2D overhead map. We did not use 3D object detectors directly because we found that 2D object detectors are more reliable than 3D ones likely because of the tremendous pretraining in large-scale 2D object detection datasets, such as MS-COCO [24]. Finally, to create the 3D maps $\mathbf{M}^{\mathrm{3D}}$, we concatenate the 3D occupancy maps with the 3D semantic maps .

We further maintain an object memory $\mathcal{M}^{\mathrm{O}}$ as a list of object detection 3D centroids and their predicted semantic category labels $\mathcal{M}^{\mathrm{O}} = \{[(X, Y, Z)_i, \ell_i \in \{1 \ldots K\}], i = 1 \ldots N\}$, where N is the number of objects detected thus far. The object centroids are expressed with respect to the coordinate system of the agent, and, similar to the semantic maps, are updated over time using egomotion.

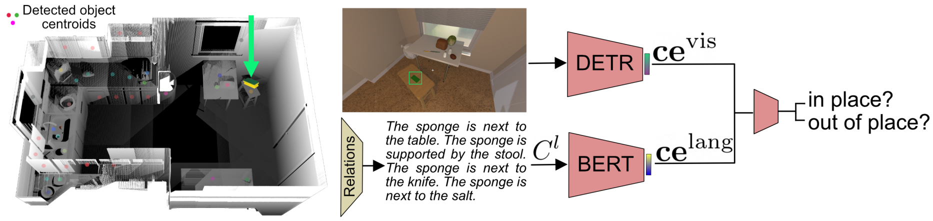

Fig. 3. out-of-place objection classification using spatial language description features $\mathbf{ce}^{\text{lang}}$ and visual features $\mathbf{ce}^{\text{vis}}$.

3.2 Detecting out-of-place objects

TIDEE detects objects and classifies whether each one is in or out-of-place (OOP) using both visual object features and language descriptions of the object's spatial relations with its surrounding objects, such as *"The alarm clock is on the sofa. The alarm clock is next to the coffee table."* We train three OOP classifiers: one that relies only on visual features, one that relies only on language descriptions of the relations of the object with its surroundings that can more easily adapt to user preferences, and one that fuses both visual and language features, as shown in Figure 3.

The visual OOP classifier (`dDETR-OOP`) builds upon our d-DETR detector. Specifically, we augment our d-DETR detector with a second decoding head and jointly train it under the tasks of localizing objects and predicting their semantic categories, as well as their in or out-of-place status. We consider the query embedding of the d-DETR decoder as relevant visual features $\mathbf{ce}^{\text{vis}}$ for OOP classification.

The language OOP classifier (`BERT-OOP`) infers the relations of the detected object to surrounding objects and describes them in language form. We consider the following spatial relations: (i) *A supported-by B*, where B is a receptacle class, (ii) *A next-to B*, *A closest-to B*. We detect these pairwise relations using Euclidean distances on detected 3D object centroids in the object memory $\mathcal{M}^{\text{O}}$. For more details on our object spatial relation detection, please see the supplementary. We represent all detected pairwise relations as sentences of the form "The {detection class} is {relation} the {related class}", and concatenate the sentences to form a paragraph, as shown in Figure 3. We map this object spatial context description paragraph into a neural vector $\mathbf{ce}^{\text{lang}}$ for the relation set given by the [CLS] token from the BERT model [13] pretrained on a language masking task and then trained for plausible/non-plausible classification in our training set. A benefit of the language OOP classifier is that it can adapt to user's specifications without any visual exemplars of plausible/implausible object arrangements. Consider, for example, the instruction *"I want my alarm clock on the bed stand"*. Using such instruction, we generate positive and negative descriptions of in and out-of-place alarm clocks by adapting the preference into a positive sample (e.g. "Alarm clock supported-by the bed stand"), and

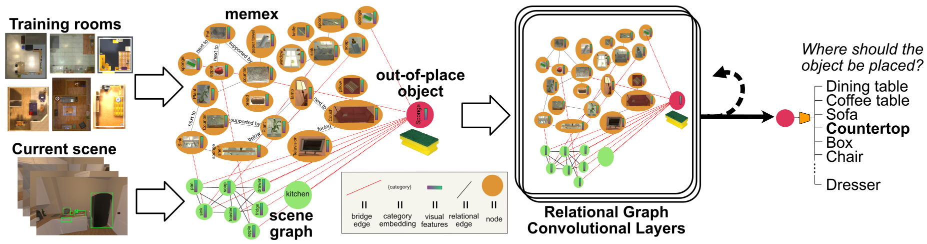

Fig. 4. Graph inference over the union of the Memex graph and the current scene graph infers plausible receptacle categories for an out-of-place object.

taking relations in the training set that include the alarm clock and a different receptacle class as negative samples ("Alarm clock supported-by the desk").

The multimodal classifier (`dDETR+BERT-OOP`) concatenates $\mathbf{ce}^{\text{vis}}$ and $\mathbf{ce}^{\text{lang}}$ as input to predict OOP classification labels for the detected object.

3.3 Inferring plausible object contexts with a neural associative graph memory

Once an OOP object is detected and picked up, TIDEE infers a plausible placement location for the object in the current scene. As shown in Figure 4, TIDEE includes a neural graph module which is trained to predict plausible object placement proposals of OOP objects by passing information between the OOP object to be placed, a memory graph encoding plausible contextual relations from training scenes, and a scene graph encoding the object-relation configuration in the current scene. Message passing is trained end-to-end to predict one of the possible receptacle classes in AI2THOR to place the OOP object on.

We instantiate an OOP node, denoted n_{OOP}, consisting of the detected OOP object for which we want to infer a plausible receptacle category by concatenating the ROI-pooled detector backbone features and a category embedding of the predicted object category.

The structure of the memory graph (nodes and edges) is instantiated from 5 out of 20 training houses. Each object in the scene is given a node in the graph that consists of a category embedding and ROI-pooled detector backbone features using the bounding box of the object at a nearby egocentric viewpoint. Edge weights in the memory graph correspond to spatial relations detected between pairs of object instances that are within a distance threshold. We consider six spatial relations and corresponding edge types: *above*, *below*, *next to*, *supported by*, *aligned with*, and *facing* [21]. We infer these using spatial relation classifiers that operate on ground-truth 3D oriented bounding boxes. Though the graph may contain noisy, non-important edges between object instances, for example, *"the coffee table is next to the bed"* which may introduce a spurious dependence between a bed and a coffee table instance, the edge kernel weights are trained end-to-end to infer plausible receptacles for OOP objects, and thus graph inference can learn to ignore such spurious edges. We call our memory graph

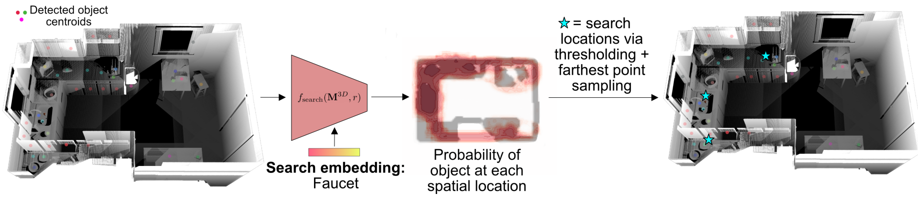

Fig. 5. The visual search network conditions on an object category of interest, and proposes locations for the agent to visit in the scene to find instances of that category.

"Memex" to highlight that nodes represent object instances, similar to [25], and not object categories as in previous works [42].

The structure of the scene graph [22] is instantiated from observations obtained while mapping the current scene, as in Section 3.1. Nodes in the scene graph represent ROI-pooled features and category embeddings of objects detected by the agent in $\mathcal{M}^{\text{O}}$. We include an additional node for the room type. We fully-connect all nodes within the scene graph. Compared to the Memex graph, we do not include separate edge weights for relations as most of the Memex relations require accurate 3D bounding boxes that we do not have access to at inference time.

We add "bridge edges", as additional learnable edge weights, between nodes in the scene graph and Memex nodes with the same category, following [44], to allow information to flow between the current scene and the memory graph. We further connect n_{OOP} to all current scene nodes and to the room type node. After message passing, we pass the updated n_{OOP} through an MLP to get logits for each possible receptacle class in AI2THOR.

The network is trained for predicting plausible receptacles for OOP objects in 15 training houses. We use 15/20 houses to train the weights so as to not overlap with the houses used for the memory graph. Relation-specific edge weights are learned end-to-end by Relational Graph Convolutions (rGCN) [32]. We supervise the network via a cross-entropy loss using ground-truth receptacle categories for each "pickupable" object from the AI2THOR original scene configurations. More details of our graph inference can be found in the supplementary.

3.4 Intelligent exploration using a visual search network

After inferring a target receptacle category, TIDEE localizes it in the scene and places the OOP object on top of it. In the case that instances of the target receptacle category have already been detected in the scene, our agent navigates to the corresponding instance using its navigation path planning controllers from Section 3.1. In the case that the target receptacle category has not yet been detected, our model predicts plausible locations to search for the receptacle using a category-conditioned visual search network $f_{\text{search}}(\mathbf{M}^{\text{3D}}, r)$.

The visual search network $f_{\text{search}}(\mathbf{M}^{\text{3D}}, r)$ takes as input a 3D spatial semantic map $\mathbf{M}^{\text{3D}}$ and a receptacle category label r represented by a learned category embedding and outputs a distribution over 2D overhead locations in the current environment for TIDEE to navigate towards and find the receptacle, as shown in Figure 5. f_{search} convolves the features of the 3D semantic map with the category category features of r and predicts an overhead heatmap, trained with a standard binary cross entropy loss. We threshold the predicted heatmap m and use non-maximum suppression via farthest point sampling to obtain a set of search locations. We rank the search locations based on their score and visit them sequentially until the target receptacle category is detected with high probability. Further architectural details for f_{search} can be found in the supplementary.

4 Experiments

We test TIDEE on reorganizing untidy rooms in the test houses of the AI2THOR simulation environment. Our experiments aim to answer the following questions:

(i) How well does TIDEE perform in tidying up scenes? Section 4.2

(ii) How much does the combination of visual and semantic features help in detecting out-of-place objects over visual features alone? Section 4.3

(iii) How much does exploration guided by the proposed visual search network improve upon random exploration for detecting objects of interest? Section 4.4

(iv) How well does TIDEE perform in the task of scene rearrangement [3]—which requires memorization of a specific prior scene configuration? Section 4.5

(v) How well can TIDEE adapt zero-shot to human instructions and alter placement priors accordingly? Section 4.6

4.1 Tidying-up task definition

Dataset We create untidy scenes by selecting a subset of "pickupable" objects[2]. We displace each object from its default location by moving the object to a random location in the scene and either dropping the object or applying a force in a random direction and allowing the AI2THOR physics engine to resolve the object's end location. We consider all available room types, namely bedrooms, living rooms, kitchens and bathrooms. We generate 8000 training, 200 validation, and 100 testing messy configurations. The goal of the agent is to manipulate the messy objects back to plausible locations within the room. An episode ends once the agent executes the "done" action or a maximum of 1000 steps have been taken. For more details on the task and dataset, please see the supplementary.

4.2 Object repositioning evaluation

We have TIDEE and all baselines perform the tidy task to detect out-of-place objects and reposition them within the scene.

[2] Pickupable objects are a predefined set of 62 object classes in AI2THOR [23] that are able to be picked up and repositioned by the agent, such apple, book, and laptop.

Evaluation metrics Quantitative evaluation of object repositioning is difficult: an object may have multiple plausible locations in a scene, and therefore measuring the distance from a single initial ground-truth 3D location is usually not reflective of performance. We thus evaluate the plausibility of object repositions of our model from those of baseline models by querying human evaluators in Amazon Mechanical Turk (AMT). Given two candidate repositions by for the same object TIDEE and a baseline, we ask human evaluators to select the one they find most plausible. We include the AMT interface we used in the supplementary.

Table 1. Percent of human evaluators that prefer TIDEE object repositions versus baselines. Reported is mean and standard error across subjects (n=5). All preferences are significantly above chance (*p<0.05, **p<0.01, Binomial test). Bold indicates higher preference for TIDEE.

TIDEE vs `CommonMemory`	**54.30±3.32***
TIDEE vs `WithoutMemex`	**54.32±4.67***
TIDEE vs `3DSmntMap2Place`	**57.69±1.29****
TIDEE vs `RandomReceptacle`	**64.59±2.94****
TIDEE vs `MessyPlacement`	**92.06±1.57****
TIDEE vs `AI2THORPlacement`	34.00±3.13**

Table 2. Evaluating visual search performance for finding objects of interest in test scenes for TIDEE and an exploration baseline that uses our 2D overhead occupancy maps to propose random search locations [41].

	% Success ↑	Time Steps ↓
TIDEE	**72.4**	**88.8**
w/o VSN	64.8	100.9

Baselines We compare TIDEE against baselines that vary in their way of inferring plausible receptacle categories for repositioning of out-of-place objects. All baselines use the same mapping and planning for navigation, the same multimodal classifier for detecting out-of-place objects (`dDETR+BERT-OOP`), and the visual search network for localizing receptacle instances of a category. We compare placements from TIDEE against the following baselines: (i) `CommonMemory`: A model that considers the most common receptacle in the training set for the out-of-place object category. (ii) `WithoutMemex`: A model that uses the scene graph but not the Memex for graph inference. (iii) `3DSmntMap2Place`: A model that proposes repositioning locations within the current scene by conditioning the visual search network on the category label of the out-of-place object. We threshold all predicted map locations and do farthest point sampling to obtain a set of diverse object placement proposals. The proposals are sorted by confidence value and visited sequentially until any receptacle is found within the local region of the proposed location. (iv) `RandomReceptacle`: A model that selects as the target receptacle the first receptacle detected by a random exploration agent. (v) `AI2THORPlacement`: The location of the OOP object in the original (tidy) AITHOR scene. The default object positions usually follow commonsense priors of scene arrangements. (vi) `MessyPlacement`: The location of the OOP object in the messy scene.

We report human preferences for OOP object repositions for our model versus each of the baselines in Table 1. TIDEE is preferred 54.3% of the time over `CommonMemory`, the most competitive of the baselines. `CommonMemory` does not consider the visual features of the out-of-place object, rather, only its semantic category, and thus cannot reason using sub-categorical information regarding object placements. TIDEE is still preferred 34% of the time over the `AI2THORPlacement` placements indicating that its re-placements are plausible and competitive with an oracle. We note that a perfect model would at best obtain a (50-50) preference compared to these placements provided by the AITHOR environment designers.

4.3 Out-of-place detector evaluation

In this section, we evaluate TIDEE's accuracy for detecting objects in and out-of-place from images collected from the test home environments. An in-place object is one in its default location in the AITHOR scene, while an out-of-place object is one moved out-of-place as defined at the beginning of Section 4. We compute average precision (AP) at IOU thresholds of 0.25 and 0.5 for in-place (IP) and out-of-place (OOP) objects, as well as the meanAP (mAP) for visual only (`dDETR-OOP`), language only (`BERT-OOP`) and multimodal (`dDETR+BERT-OOP`) classifiers described in Section 3.2. We also compare against an oracle BERT classifier that assumes access to ground-truth 3D object centroids, bounding boxes, and category labels to detect relations and form descriptive utterances of in and out-of-place objects, which we call `oracle-BERT-OOP`.

We show quantitative comparisons in Table 3. Combining language and visual features performs slightly better than using language or visual features alone for out-of-place object detection. The benefit of the language classifier is that it can be re-trained on-demand to adjust to human instructions without any visual training data, as we explain in Section 4.6. The good performance of the oracle BERT classifier suggests that simple relations inferred from accurate 3D centroids likely suffice to classify in- and out-of-place objects in AI2THOR scenes if perception is perfect.

Table 3. Average precision (AP) for in and out-of-place object detection. Combining vision and language features helps detection performance. IP = in place; OOP = out of place.

	$\text{mAP}_{0.25}$	$\text{AP}^{IP}_{0.25}$	$\text{AP}^{OOP}_{0.25}$	$\text{mAP}_{0.5}$	$\text{AP}^{IP}_{0.5}$	$\text{AP}^{OOP}_{0.5}$
`dDETR+BERT-OOP`	**51.09**	**58.41**	**43.78**	**46.26**	**53.64**	**38.88**
`dDETR-OOP`	49.98	57.60	42.37	44.98	52.79	37.17
`BERT-OOP`	31.71	41.13	22.30	25.25	33.79	16.71
`oracle-BERT-OOP`	–	–	–	90.70	96.24	85.16

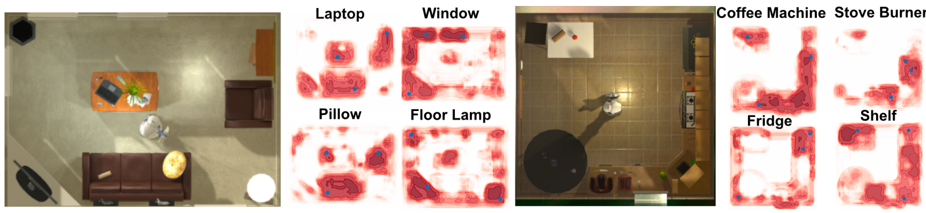

Fig. 6. Visual Search Network predictions encode object location priors for different object categories.

4.4 Visual search network evaluation

In this section, we compare exploration for finding objects of interest in test scenes (one category of the possible 116 per episode) guided by TIDEE's visual search network against an exploration agent that uses the 2D overhead occupancy map and samples unvisited locations to visit, similar to Yamauchi [41]. We adopt the success criteria similar to the object goal navigation [4] and define a successful trial as one where the agent is within a radius of *any* target object category instance and the object is visible within view. We report the percentage of successful episodes performed by the agent and average number of time steps across all episodes in Table 2. If an agent fails an episode, the number of time steps defaults to the maximum allowable steps for each episode (200). TIDEE outperforms the exploration baseline. We show visualizations of the network predictions in Figure 6, and also in the supplementary.

4.5 Scene Rearrangement Challenge

We test TIDEE to generalize to the recent scene rearrangement benchmark of [37], which considers an AI agent tasked with repositioning objects in a scene in order to match the prior configuration of an identical scene. We consider the two-phase rearrangement setup where in the first "walkthrough" phase, the agent observes a room in its initial configuration, and in the second so called "unshuffle" phase, observes the same room with some objects in new configurations and is tasked to rearrange the room back to its initial configuration. While the challenge considers both rearranging objects to different locations within a room and changing their open/close states, we only consider repositioning of objects because our current model does not handle opening and closing receptacles.

We simplify TIDEE's architecture and only maintain the 2D & 3D occupancy map for navigation and the object memory $\mathcal{M}^{\mathrm{O}}$ for keeping track of objects and their labels over time. We start each phase by exploring the scene and detecting objects. As in Section 3.2, we infer the relations for all pickupable objects in the object memory $\mathcal{M}^{\mathrm{O}}$ in the initial and shuffled scenes. We consider an object of the initial scene displaced if its category label has been detected in the shuffled scene and the proportion of inferred relations that are different across the two scenes ($\{\#\text{ same relations}\}/\{\#\text{ different relations}\}$), initial and shuffled, is less than a threshold (we use 0.35). For example, a bowl with relations *bowl next to sink, bowl supported by countertop, bowl next to cabinet* in the initial scene, and relations *bowl next to chair, bowl supported by dining table, bowl next to lamp* in the shuffled scene is considered misplaced by TIDEE. Then, our agent navigates to the object's 3D location detected in the initial scene and places it there. Our agent uses the navigation controllers from Section 3.1.

We use the evaluations metrics described in Weihs et. al. [37] : (1) Success (↑): the trial is a success if the initial configuration is fully recovered in the unshuffle phase; (2) % FixedStrict (↑): the proportion of objects that were misplaced initially but ended in the correct configuration (if a single in-place object is moved out-of-place, this metric is set to 0); (3) % Energy (↓): the energy is a measure for the similarity of the rearranged scene and the original scene, the lower the more similar (for more details, refer to Weihs et. al. [37]); (4) % Misplaced (↓): this metric equals the number of misplaced objects at the end of the episode divided by the number of misplaced objects at the start.

We report TIDEE's performance compared to the top performing methods for the two-phase re-arrangement in Table 4. The model from Weihs et. al. [37] trains a reinforcement learning (RL) agent with proximal policy optimization (PPO) and imitation learning (IL) given RGB images as input and includes a semantic mapping component adapted from the Active Neural SLAM model [7]. We additionally show the robustness of TIDEE to realistic sensor measurements. We consider three different versions of TIDEE depending on the source of egomotion and depth information: **(i)** TIDEE uses ground-truth egomotion and depth. **(ii)** TIDEE*+noisy pose* uses ground-truth depth and egomotion from the LocoBot agent in AI2THOR with Gaussian movement noise added to each movement based on measurements of the real LocoBot robot [28] (forward movement $\sigma = 0.005$ meters; rotation $\sigma = 0.5$ degrees). **(iii)** TIDEE*+est. depth* uses ground-truth egomotion and depth obtained from the depth prediction model of Blukis et. al. [5], which takes in egocentric RGB images. The model is pre-trained and then finetuned on the training scenes of ALFRED [34].

4.6 Updating placement priors by instruction

In this section, we test whether we can alter the OOP classifier on-demand using language specifications for in and out-of-place. Since alarm clocks are often found on desks in AI2THOR, we tested whether augmenting training by pairing the sentence *"alarm clock is supported by desk"* with the out-of-place label would allow us to alter the OOP classifier's output. As shown in Table 5, across three test scenes where alarm clocks are found on desks, the initial OOP object classifier gives us low probability that the alarm clock on the desk is out-of-place. We then add in the language description *"alarm clock is supported by desk"* for a small amount of additional iterations. As shown in Table 5, we find that our procedure suffices to alter the priors of the classifier. We provide additional examples using various object-relation pairings in the supplementary.

Table 4. Test set performance on 2-Phase Rearrangement Challenge (2022). TIDEE outperforms the baseline of [37] even with realistic noise.

	% FixedStrict ↑	% Success ↑	% Energy ↓	% Misplaced ↓
TIDEE	**11.6**	**2.4**	**93**	**94**
TIDEE *+noisy pose*	7.7	1.2	101	101
TIDEE *+est. depth*	5.9	0.6	97	97
TIDEE *+noisy depth*	11.4	2.0	94	95
Weihs *et al.* [37]	0.5	0.0	110	110

Limitations. TIDEE has the following two limitations: i) It does not consider open and closed states of objects, or their 3D pose as part of the messy and reorganization process, which are direct avenues for future work. ii) The messy rooms we create by randomly misplacing objects may not match the messiness in human environments.

Table 5. Altering priors with instructions. The confidence of the out-of-place classifier for clocks found on desks in three test scenes increases when the additional spatial description for indicating out-of-place clocks.

	Before	After
Clock #1	.08	.73
Clock #2	.10	.62
Clock #3	.12	.76

5 Conclusion

We have introduced TIDEE, an agent that tidies up rooms in home environments using commonsense priors encoded in visuo-semantic out of place detectors, visual search networks that guide exploration to objects, and a Memex neural graph memory of objects and relations that infers plausible object context. We evaluate with human evaluators, and find that TIDEE outperforms agents that lack it's modular architecture, as well as modular agents that lack TIDEE's commonsense priors. TIDEE can be instructed in natural language to follow on-demand specifications for object placement. Finally, we establish a new state-of-the-art for the scene rearrangement challenge of Weihs et. al. [37] by simplifying TIDEE's architecture to memorize a single scene as opposed to using a prior learned across multiple environments. We believe TIDEE takes an important step towards embodied visuo-motor commonsense reasoning.

Acknowledgements. This material is based upon work supported by National Science Foundation grants GRF DGE1745016 & DGE2140739 (GS), a DARPA Young Investigator Award, a NSF CAREER award, an AFOSR Young Investigator Award, and DARPA Machine Common Sense. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Army, the National Science Foundation, or the United States Air Force.

References

1. 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)
2. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., Van Den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3674–3683 (2018)
3. 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. ArXiv **abs/2011.01975** (2020)
4. 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)
5. Blukis, V., Paxton, C., Fox, D., Garg, A., Artzi, Y.: A persistent spatial semantic representation for high-level natural language instruction execution. In: Conference on Robot Learning. pp. 706–717. PMLR (2022)
6. Chang, M., Gupta, A., Gupta, S.: Semantic visual navigation by watching youtube videos. Advances in Neural Information Processing Systems **33**, 4283–4294 (2020)
7. Chaplot, D.S., Gandhi, D., Gupta, S., Gupta, A., Salakhutdinov, R.: Learning to explore using active neural slam. In: International Conference on Learning Representations (ICLR) (2020)
8. Chaplot, D.S., Gandhi, D.P., Gupta, A., Salakhutdinov, R.R.: Object goal navigation using goal-oriented semantic exploration. Advances in Neural Information Processing Systems **33** (2020)
9. Chaplot, D.S., Jiang, H., Gupta, S., Gupta, A.: Semantic curiosity for active visual learning. In: European Conference on Computer Vision. pp. 309–326. Springer (2020)
10. Chen, T., Gupta, S., Gupta, A.: Learning exploration policies for navigation. In: International Conference on Learning Representations (2019), `https://openreview.net/pdf?id=SyMWn05F7`
11. Chen, X., Li, L.J., Fei-Fei, L., Gupta, A.: Iterative visual reasoning beyond convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7239–7248 (2018)
12. Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., Batra, D.: Embodied question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–10 (2018)
13. 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, `https://aclanthology.org/N19-1423`
14. Fan, L., Zhu, Y., Zhu, J., Liu, Z., Zeng, O., Gupta, A., Creus-Costa, J., Savarese, S., Fei-Fei, L.: Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In: Conference on Robot Learning. pp. 767–782. PMLR (2018)

15. Fang, Z., Jain, A., Sarch, G., Harley, A.W., Fragkiadaki, K.: Move to see better: Self-improving embodied object detection. The British Machine Vision Conference (2021)
16. Gan, C., Schwartz, J., Alter, S., Schrimpf, M., Traer, J., De Freitas, J., Kubilius, J., Bhandwaldar, A., Haber, N., Sano, M., et al.: Threedworld: A platform for interactive multi-modal physical simulation. arXiv preprint arXiv:2007.04954 (2020)
17. Gan, C., Zhou, S., Schwartz, J., Alter, S., Bhandwaldar, A., Gutfreund, D., Yamins, D.L., DiCarlo, J.J., McDermott, J., Torralba, A., Tenenbaum, J.B.: The threedworld transport challenge: A visually guided task-and-motion planning benchmark towards physically realistic embodied ai. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 8847–8854 (2022). https://doi.org/10.1109/ICRA46639.2022.9812329
18. Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., Farhadi, A.: Iqa: Visual question answering in interactive environments. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4089–4098 (2018)
19. Gupta, S., Davidson, J., Levine, S., Sukthankar, R., Malik, J.: Cognitive mapping and planning for visual navigation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
20. Haber, N., Mrowca, D., Fei-Fei, L., Yamins, D.L.: Learning to play with intrinsically-motivated self-aware agents. 32nd Conference on Neural Information Processing Systems (2018)
21. Hayward, W.G., Tarr, M.J.: Spatial language and spatial representation. Cognition **55**, 39–84 (1995)
22. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3668–3678 (2015)
23. 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)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
25. Malisiewicz, T., Efros, A.A.: Beyond categories: The visual memex model for reasoning about object relationships. In: NIPS (December 2009)
26. Marino, K., Salakhutdinov, R., Gupta, A.: The more you know: Using knowledge graphs for image classification. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20–28 (2017). https://doi.org/10.1109/CVPR.2017.10
27. Min, S.Y., Chaplot, D.S., Ravikumar, P., Bisk, Y., Salakhutdinov, R.: Film: Following instructions in language with modular methods (2021)
28. Murali, A., Chen, T., Alwala, K.V., Gandhi, D., Pinto, L., Gupta, S., Gupta, A.: Pyrobot: An open-source robotics framework for research and benchmarking. arXiv preprint arXiv:1906.08236 (2019)
29. Padmakumar, A., Thomason, J., Shrivastava, A., Lange, P., Narayan-Chen, A., Gella, S., Piramuthu, R., Tur, G., Hakkani-Tur, D.: Teach: Task-driven embodied agents that chat (2021)
30. Ramakrishnan, S.K., Al-Halah, Z., Grauman, K.: Occupancy anticipation for efficient exploration and navigation. In: European Conference on Computer Vision. pp. 400–418. Springer (2020)

31. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al.: Habitat: A platform for embodied ai research. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9339–9347 (2019)
32. Schlichtkrull, M., Kipf, T.N., Bloem, P., Van Den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: European semantic web conference. pp. 593–607. Springer (2018)
33. Shen, B., Xia, F., Li, C., Martín-Martín, R., Fan, L., Wang, G., Pérez-D'Arpino, C., Buch, S., Srivastava, S., Tchapmi, L.P., Tchapmi, M.E., Vainio, K., Wong, J., Fei-Fei, L., Savarese, S.: igibson 1.0: a simulation environment for interactive tasks in large realistic scenes. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems. p. accepted. IEEE (2021)
34. 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10740–10749 (2020)
35. Suglia, A., Gao, Q., Thomason, J., Thattai, G., Sukhatme, G.S.: Embodied bert: A transformer model for embodied, language-guided visual task completion. In: EMNLP 2021 Workshop on Novel Ideas in Learning-to-Learn through Interaction (2021), https://www.amazon.science/publications/embodied-bert-a-transformer-model-for-embodied-language-guided-visual-task-completion
36. Wang, X., Ye, Y., Gupta, A.: Zero-shot recognition via semantic embeddings and knowledge graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6857–6866 (2018)
37. Weihs, L., Deitke, M., Kembhavi, A., Mottaghi, R.: Visual room rearrangement. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2021)
38. Weihs, L., Kembhavi, A., Ehsani, K., Pratt, S.M., Han, W., Herrasti, A., Kolve, E., Schwenk, D., Mottaghi, R., Farhadi, A.: Learning generalizable visual representations via interactive gameplay. International Conference on Learning Representations (2021)
39. Wijmans, E., Kadian, A., Morcos, A.S., Lee, S., Essa, I., Parikh, D., Savva, M., Batra, D.: Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In: ICLR (2020)
40. Wortsman, M., Ehsani, K., Rastegari, M., Farhadi, A., Mottaghi, R.: 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. pp. 6750–6759 (2019)
41. Yamauchi, B.: 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'. pp. 146–151. IEEE (1997)
42. Yang, W., Wang, X., Farhadi, A., Gupta, A., Mottaghi, R.: Visual semantic navigation using scene priors. In: Proceedings of (ICLR) International Conference on Learning Representations (May 2019)
43. Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., Levine, S.: Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In: Conference on Robot Learning. pp. 1094–1100. PMLR (2020)
44. Zareian, A., Karaman, S., Chang, S.F.: Bridging knowledge graphs to generate scene graphs. In: European Conference on Computer Vision. pp. 606–623. Springer (2020)

45. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable {detr}: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=gZ9hCDWe6ke