ESC: Exploration with Soft Commonsense Constraints for Zero-shot Object Navigation

Kaiwen Zhou [1] **Kaizhi Zheng** [1] **Connor Pryor** [1] **Yilin Shen** [2] **Hongxia Jin** [2] **Lise Getoor** [1] **Xin Eric Wang** [1]

Abstract

The ability to accurately locate and navigate to a specific object is a crucial capability for embodied agents that operate in the real world and interact with objects to complete tasks. Such object navigation tasks usually require large-scale training in visual environments with labeled objects, which generalizes poorly to novel objects in unknown environments. In this work, we present a novel zero-shot object navigation method, Exploration with Soft Commonsense constraints (ESC), that transfers commonsense knowledge in pre-trained models to open-world object navigation without any navigation experience nor any other training on the visual environments. First, ESC leverages a pre-trained vision and language model for open-world prompt-based grounding and a pre-trained commonsense language model for room and object reasoning. Then ESC converts commonsense knowledge into navigation actions by modeling it as soft logic predicates for efficient exploration. Extensive experiments on MP3D (Chang et al., 2017), HM3D (Ramakrishnan et al., 2021), and RoboTHOR (Deitke et al., 2020) benchmarks show that our ESC method improves significantly over baselines, and achieves new state-of-the-art results for zero-shot object navigation (e.g., 288% relative Success Rate improvement than CoW (Gadre et al., 2022) on MP3D).

1. Introduction

Object navigation (ObjNav) is a task in which an embodied agent must navigate to a specific goal object within an unknown environment (Batra et al., 2020). This task is fundamental to other navigation-based embodied tasks because navigating to a goal object is the preliminary for the agent to interact with it. While current state-of-the-art methods for object navigation achieve good results when trained on specific datasets with limited goal objects and similar environments, they often perform poorly when faced with novel objects or environments due to distribution shifts. Real-world situations often involve diverse objects and varied environments, making it difficult and costly to collect extensive, annotated trajectory data. As a result, generalized zero-shot object navigation, in which the navigation agent can adapt to novel objects and environments without additional training, is a crucial area of study.

Figure 1. **Commonsense reasoning in object navigation.** In object navigation, our agent first does a semantic understanding of the current scene (red text in the figure) and then performs commonsense reasoning (blue text in the figure). The agent reasons that a fireplace is likely to be in a living room, so it decides to explore the unobserved part of the living room (the frontier adjacent to the observed part of the living room).

Successfully navigating to a goal object requires two key abilities, (1) *semantic scene understanding*, which involves identifying objects and rooms in the environment, and (2) *commonsense reasoning*, which involves making logical inferences about the location of the goal object based on commonsense knowledge. For example, as in Fig. 1, a fireplace is very likely in a living room, so the agent decides to explore the unseen area in the living room to find a fireplace. However, current zero-shot object navigation methods have not yet effectively addressed this requirement and often lack commonsense reasoning abilities. Existing methods require training on other goal-oriented navigation tasks and envi-

[1]University of California, Santa Cruz [2]Samsung Research America. Correspondence to: Xin Eric Wang <xwang366@ucsc.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

ronments (Majumdar et al., 2022; Al-Halah et al., 2022), or use simple heuristics for exploration (Gadre et al., 2022).

Recent studies (He et al., 2021; Radford et al., 2021; Kojima et al., 2022; Li* et al., 2022) show that large pre-trained models have a strong generalization and reasoning ability for novel tasks under zero-shot scenarios. Building upon this success, in this work, we propose a zero-shot object navigation framework, named Exploration with Soft Commonsense constraints (ESC), that leverages these pre-trained models and can seamlessly generalize to unseen environments and novel object types. As shown in Fig. 1, we first use a prompt-based vision-and-language grounding model GLIP (Li* et al., 2022) for open-world object grounding and scene understanding, which can infer the object and room information of current agent views. Benefiting from large-scale image-text pre-training, GLIP can easily generalize to new objects via prompting. Then, we utilize a pre-trained commonsense reasoning language model that takes the room and object information as context to infer the correspondence between rooms and objects.

However, there still remains a gap in converting the commonsense knowledge inferred from large language models (LLMs) into executable actions. In addition, the relationship between entities is usually uncertain, e.g., the book has a high probability in the living room, but it is not deterministic. To address these challenges, our ESC method models "soft" commonsense constraints using Probabilistic Soft Logic (PSL) (Bach et al., 2017), a declarative templating language that defines a special class of Markov random fields with first-order logical rules. Those soft commonsense constraints are then incorporated into a classic exploration method, frontier-based exploration (FBE), to determine which frontier to explore next in a zero-shot manner. Unlike previous methods that rely on implicit training of commonsense using neural networks (Yang et al., 2019; Chaplot et al., 2020a), our method explicitly uses soft logic predicates to represent knowledge in a continuous value space, which is then assigned to each frontier, enabling more effective exploration.

We demonstrate the effectiveness of our framework on three object goal navigation benchmarks, MP3D (Chang et al., 2017), HM3D (Ramakrishnan et al., 2021), and RoboTHOR (Deitke et al., 2020), with different house sizes, styles, texture features, and object types. Compared with CoW (Gadre et al., 2022) that has the same setting as ours, our method achieves around 285% relative improvement in success rate weighted by length (SPL) and success rate (SR) on MP3D and 35% relative improvement in SPL and SR on RoboTHOR. Compared with ZSON (Majumdar et al., 2022) that requires training on the HM3D dataset, our method outperforms it by 196% relative SPL on MP3D and 85% relative SPL on HM3D. Note that on the MP3D dataset, our zero-shot method is comparable with previous state-of-the-art supervised methods and achieves the best SPL.

In summary, our contributions are threefold:

- We propose the Exploration with Soft Commonsense constraints (ESC) method for zero-shot object navigation, which leverages pre-trained vision and language models for open-world scene understanding and object-level and room-level commonsense reasoning.
- Our ESC approach models soft commonsense constraints and seamlessly converts them into navigation actions using Frontier-based Exploration and Probabilistic Soft Logic, which is training-free.
- We achieve state-of-the-art results on zero-shot object goal navigation and outperform baseline methods by a large margin across three object navigation datasets and benchmarks.

2. Problem Definition

In the conventional task of *object navigation*, an agent is randomly placed within an unseen environment E with a specified object category G as a goal to find (e.g., chair, fireplace, or cabinet). The agent's objective is to navigate to any instance of the object that belongs to the aforementioned category. At each time step t, the agent is presented with an observation $\mathcal{O}$, which consists of an egocentric RGB-D image I_t and in some benchmarks, pose readings P_t. The agent needs to select an action a from the action space $\mathcal{A}$, which includes a 'STOP' action to terminate the navigation process. The navigation is considered successful if the agent stops within d_s meters of the object and the object is visible without further moving.

In contrast to supervised object navigation, which trains the agent on the objects and environments it will navigate, this work focuses on *zero-shot object navigation*: given a new set of environments $\{E_{new}\}$ and a new set of goal objects $\{o_{new}\}$ that the agent has not seen before, the agent is required to perform object goal navigation in $\{E_{new}\}$ for $\{o_{new}\}$ without training on object goal related labels. Furthermore, we target an even more challenging zero-shot scenario—the agent performs zero-shot object navigation without training on any navigation data.

3. Our ESC Approach

In this section, we outline our Exploration with Soft Commonsense constraints (ESC) framework for zero-shot object navigation. As in Fig. 2, the ESC framework first converts the input image into a semantic understanding of the scene and projects it to a semantic map (Sec. 3.1). Then it leverages large language models to perform commonsense reasoning for the spatial relations between the goal object and

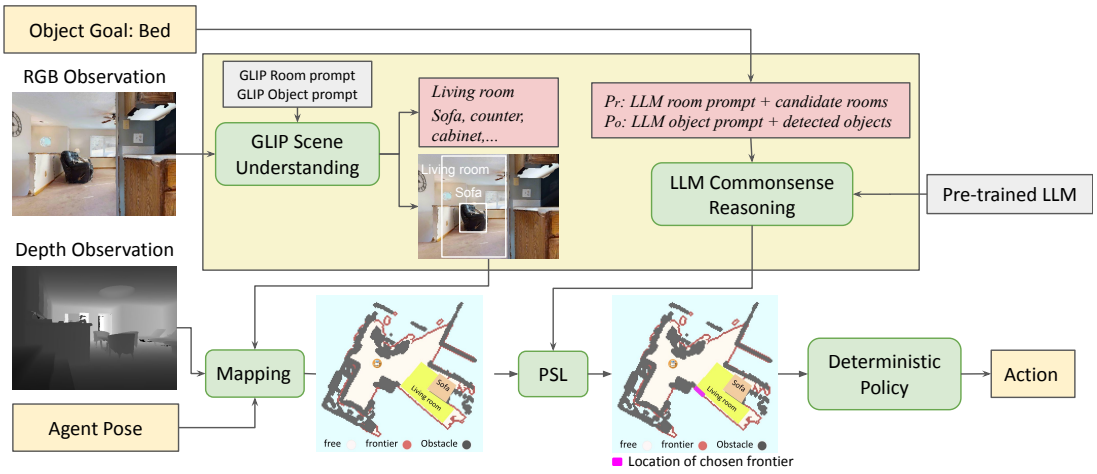

Figure 2. **The ESC framework.** During navigation, the agent performs scene understanding based on RGB observations and prompts. Meanwhile, the Mapping module constructs a semantic map containing room, object, and frontier information. Conditioned on the goal object and semantic scene information, the agent will then perform commonsense reasoning via a LLM to infer the probable location of the goal object, and select a frontier to explore using PSL.

common objects and rooms (Sec. 3.2). Lastly, it combines frontier-based exploration with semantic scene understanding and commonsense reasoning via PSL (Sec. 3.3).

3.1. Open-World Semantic Scene Understanding

Prompt-Based Scene Grounding To leverage large language models for navigation inference, we need to transform the input RGB images into semantic context in language form. To achieve this, we leverage a pre-trained grounded language-image model GLIP (Li* et al., 2022) using a text prompt. Unlike traditional object detection models such as Mask-RCNN (He et al., 2017), which is limited to fixed classes, GLIP formulates the detection task as a grounding problem by aligning the proposed image region with phrases in the text prompt and predicting the score of region-text alignment. Benefiting from large-scale image-text pretraining, GLIP can detect common indoor concepts (*e.g.* object, room) in an open-world setting. And it is easy to generalize to different environments and object goals to perform open-world object navigation.

We first define a set of common indoor objects $\{o_c\}$ [1], then we take the union of $\{o_c\}$ and all the possible goal objects $\{o_g\}$ to generate an object prompt for object grounding. The object prompt P_o will be the object names in $\{o_c\} \cup \{o_g\}$ joined by '. '. For example, if $\{o_c\} = \{cabinet, table\}$ and $\{o_g\} = \{chair, table\}$, then the object prompt P_o will be 'cabinet. chair. table.'.

However, object information in the current scene is a relatively low-level scene context. When humans search for a goal in an unseen environment, they will usually consider higher-level contexts (e.g., which room is likely to contain the goal?). Therefore, we define a set of common rooms $\{r_c\}$ in indoor environments for room prompt P_r to detect room information.

By inputting these prompts and an egocentric image into the GLIP model as in Fig. 2, we can get the detected objects $o_{t,i}$, rooms $r_{t,i}$ and bounding boxes from the current scene:

$$\{o_{t,i}, b^o_{t,i}\} = \text{GLIP}(I_t, P_o) \quad (1)$$

$$\{r_{t,i}, b^r_{t,i}\} = \text{GLIP}(I_t, P_r) \quad (2)$$

where $b^o_{t,i}$ and $b^r_{t,i}$ are the bounding boxes of the objects and rooms. Notice that these prompts can be easily extended to generalize to new test data to perform open-world semantic scene understanding.

Semantic Map Construction Based on the depth input D_t, agent location, and camera parameters, we can transform the pixels in a 2D image into 3D space, which is stored in a 3D voxel, where transformed pixels close to the floor are considered free space. Then we project the 3D voxel along the height dimension and obtain a 2D navigation map as shown in Fig. 2, which will be maintained during navigation. Through the navigation map, we can obtain the frontiers in the current map as explained in Sec. 3.3.1.

Furthermore, as shown in Fig. 2, we can project the detected room and object location into a semantic map. For object detection, we take the center of a bounding box and project it to a 2D location. For room detection, we project all the pixels in a bounding box into a 2D map and record the projected locations as the corresponding room.

[1] The full list is shown in Appendix A.1.

3.2. Commonsense Reasoning for ObjNav via LLM

In an in-door environment, a goal object will appear in certain rooms and near certain objects more frequently, and this kind of common sense is helpful for the agent to search for a goal object. Therefore, after detecting the room and object information in the current scene, we can leverage pre-trained large language models to perform commonsense reasoning conditioned on the goal object and semantic scene information via text prompt.

Specifically, for object-level and room-level inference, the large language models can reason on whether a goal object G is likely to be near each object o_i in the object prompt P_o and whether it is likely to be in each room r_i in the room prompt P_r. The prediction output of the large language models will be the real-value scores $S(G|o_i), S(G|r_i) \in [0, 1]$ of each (goal, object) pair and (goal, room) pair. How to get the scores from language models and the text prompt for the language model can vary between different LLMs. We mainly use Deberta v3 (He et al., 2021) in our method for its effectiveness and accessibility. Details about the LLMs can be found in Appendix A.3.

3.3. Commonsense Guided Exploration

3.3.1. Frontier-based Exploration

In object goal navigation, exploring the environment efficiently is very important to find the target object, as the object is often not seen in the agent's initial location. Gadre et al. (2022) use a heuristic exploration method, Frontier-based Exploration (FBE), to explore the environment, which shows superior performance compared with learning-based exploration methods. As shown in Fig. 1, a frontier in a map is defined as the border between the free area and the unseen area. Free area is defined as the area that the agent has seen and is not occupied by obstacles. One common strategy for frontier selection is to choose the closest frontier (with a distance threshold d_f) as the next subgoal (Gadre et al., 2022). However, choosing the closest frontier as a subgoal to explore may not be optimal in semantic-rich environments and may be against commonsense. For example, the agent might check the frontiers behind the couch in a living room to search for a bed.

Therefore, we propose to introduce commonsense knowledge in LLMs into frontier-based exploration. Our goal is to make the frontier selection decision based on not only the distances d_i from the agent but also object o^t and room r^t information around the frontiers:

$$P(F) = P(F|d_i, o^t, r^t) \tag{3}$$

Intuitively, we are more likely to choose a frontier close to an object near which the goal object is likely to appear, or a frontier in a room in which the goal object should be. Notice that this kind of rule represents a concept that is not always correct, as there may be multiple frontiers satisfying potentially many rules, and the conditions within these rules are continuously valued. Therefore, we need a system that can express these soft rules and logic well, i.e., Probabilistic Soft Logic (PSL).

3.3.2. Soft Commonsense Constraints

Now we describe how we combine commonsense reasoning with frontier-based exploration mentioned above via PSL and enable frontier selection with soft commonsense constraints. Probabilistic Soft Logic (PSL) (Bach et al., 2017) is a probabilistic programming language defining hinge-loss Markov random fields (HL-MRF) using a syntax based on first-order logic. Specifically, PSL models dependencies between relations and attributes of entities in a domain, defined as *atoms*, which are encoded with weighted first-order logical clauses and linear arithmetic inequalities referred to as *rules*. We define four following rules for object navigation.

a. Object reasoning We first consider object-level reasoning. To encourage the agent to explore those frontiers near some objects that are likely to appear around the goal object, we have the rule:

$$\begin{aligned} & w : \mathrm{IsCooccur}(\mathrm{Goal}, \mathrm{Object}) \\ \wedge\ & \mathrm{IsNearObj}(\mathrm{Frontier}, \mathrm{Object}) \\ \longrightarrow\ & \mathrm{ChooseFrontier}(\mathrm{Frontier}) \end{aligned} \tag{4}$$

The parameter w is the weight of the rule, quantifying its relative importance in the model. This rule includes three atoms: $\mathrm{IsCooccur}(\mathrm{Goal}, \mathrm{Object})$, $\mathrm{IsNearObj}(\mathrm{Frontier}, \mathrm{Object})$, and $\mathrm{ChooseFrontier}(\mathrm{Frontier})$. The value of $\mathrm{IsCooccur}(\mathrm{Goal}, \mathrm{Object})$ is the co-occurrence score $S(G|o_i)$ for (Goal, Object) pair predicted by the language models. The value of $\mathrm{IsNearObj}(\mathrm{Frontier}, \mathrm{Object})$ is the confidence of the object prediction by GLIP if the object is within d_o meters of the frontier according to the semantic map from Sec. 3.1; otherwise, $\mathrm{IsNearObj}(\mathrm{Frontier}, \mathrm{Object}) = 0$. Furthermore, to discourage the agent from going to those frontiers near some objects that are unlikely to be around the goal object, we have a corresponding negative rule:

$$\begin{aligned} & w : !\mathrm{IsCooccur}(\mathrm{Goal}, \mathrm{Object}) \\ \wedge\ & \mathrm{IsNearObj}(\mathrm{Frontier}, \mathrm{Object}) \\ \longrightarrow\ & !\mathrm{ChooseFrontier}(\mathrm{Frontier}) \end{aligned} \tag{5}$$

b. Room reasoning Similar to object reasoning, we encourage the agent to explore the frontiers near or in a room where the goal object is likely to appear, and discourage it from exploring the frontiers near or in a room where the goal object is unlikely to appear. Thus, we have two rules for room reasoning similar to Eq. (4) and Eq. (5) where 'Object' is substituted with 'Room'.

c. Distance constraint The vanilla frontier-based exploration method (Gadre et al., 2022) chooses the frontier with the shortest distance from the agent, which encourages the agent to continue exploring one area until there is nothing to explore. We also add a shortest-distance rule to encourage the agent to explore nearby frontiers:

$$\begin{aligned} &w : \text{ShortDist}(\text{Frontier}) \\ \longrightarrow\ &\text{ChooseFrontier}(\text{Frontier}) \end{aligned} \tag{6}$$

d. Sum constraint We adapt a PSL hard constraint to limit the sum of the scores of choosing all the frontiers to one:

$$\text{ChooseFrontier}(+\text{Frontier}) = 1 \tag{7}$$

This constraint prevents the degenerated solution where all the target variables are equal to one and encourages the frontiers to compete with each other.

PSL Inference During PSL inference, atoms will be instantiated with data and referred to as ground atoms. Taking equation 4 as an example. Ground atoms are mapped to either an observed variable X, like IsCooccur and IsNearObj, or a target variable Y, like ChooseFrontier. Then, valid combinations of ground atoms substituted in the rules create ground rules. Each ground rule creates one or more hinge-loss potentials defined over logical rules, which are relaxed using Łukasiewicz continuous valued logical semantics:

$$\phi(Y, X) = [max(0, l(Y, X))]^p \tag{8}$$

where l is a linear penalty function[2] defined by PSL. $\phi(Y, X)$ represents the *distance to satisfaction* of this ground rule. The values of X, Y are in the range $[0, 1]$, and $p \in \{1, 2\}$ optionally squares the potentials.

Given all the observed variables X and target variables Y, PSL defines an HL-MRF over the target variables:

$$P(Y|X) = \frac{1}{Z(Y)} exp(-\sum_{i=1}^{m} w_i \phi_i(Y, X)) \tag{9}$$

$$Z(Y) = \int_Y exp(-\sum_{i=1}^{m} w_i \phi_i(Y, X)) \tag{10}$$

where m denotes the number of potential functions, ϕ_i is the i^{th} potential function, w_i is the weight of the template rule for ϕ_i.

Therefore, the optimization for the distribution can be converted to a convex optimization problem:

$$Y^* = \underset{Y}{argmin} \sum_{i=1}^{m} w_i \phi(Y, X) \tag{11}$$

[2]Details of the penalty function can be found in Appendix A.2.

One-hot constraint PSL solver Normally, a PSL program can use convex optimization algorithms such as ADMM (Boyd et al., 2011) to find a solution. However, since the final choice of the agent is only one frontier, we further limit the solution space to one hot encoding space, where each one-hot encoding represents selecting one frontier. Our one-hot constraint solver is performed by calculating the violation of constraints for each one-hot encoding. The encoding with the lowest loss, representing the frontier with the lowest value of violated constraints, will be chosen. This approach can help us save the iteration time of optimization compared with convex optimization algorithms.

Navigation Policy The agent adapts a simple navigation policy with commonsense reasoning. It will choose a new frontier based on PSL inference after it reaches a frontier. After the agent detect the goal object, it will directly navigate to it. The agent is equipped with a deterministic policy as in Fig. 2 to help it navigate to the goal or a frontier. The full navigation algorithm is in Appendix A.4.

4. Experimental Setup

4.1. Benchmarks and Metrics

MP3D (Chang et al., 2017) is used in Habitat ObjectNav challenges, containing 2195 validation episodes on 11 validation environments with 21 goal object categories.

HM3D (Ramakrishnan et al., 2021) is used in Habitat 2022 ObjectNav challenge, containing 2000 validation episodes on 20 validation environments with 6 goal object categories.

RoboTHOR (Deitke et al., 2020) is used in RoboTHOR 2020, 2021 ObjectNav challenge, containing 1800 validation episodes on 15 validation environments with 12 goal object categories. Different from HM3D and MP3D, the goal objects in RoboTHOR are mainly small objects.

Metrics On all three benchmarks, the number of maximum navigation steps is 500. We report and compare Success Rate (SR) and Success Rate weighted by inverse path Length (SPL) (Anderson et al., 2018), of which SPL is the primary metric used in the Habitat and RoboTHOR challenges. In ablation studies, we also report SoftSPL (Datta et al., 2020), which reflects the navigation progress made by the agent considering navigation efficiency.

Agent Configurations The agent has a height of 0.88m, with a radius of 0.18m. The agent receives 640×480 RGB-D egocentric views from a camera with 79° HFoV placed 0.88m from the ground. All the agents have action space of $\mathcal{A} = \{$MoveForward, RotateRight, RotateLeft, LookUp, LookDown, Stop$\}$. The moving step is 0.25m, and each rotation turns the agent by 30°. In MP3D and HM3D datasets, the agent will receive its GPS location at each step.

Table 1. **Zero-shot object navigation results** on MP3D (Chang et al., 2017), HM3D (Ramakrishnan et al., 2021), and RoboTHOR (Deitke et al., 2020) benchmarks. Notice that our method and CoW (Gadre et al., 2022) are the only two zero-shot methods with no navigation training experience. Our method significantly outperforms previous zero-shot methods. * The training environment of ProcTHOR is similar to RoboTHOR using the same simulator.

Model	Supervised	Trained on environment	Navigation training	MP3D		HM3D		RoboTHOR	
				SPL↑	SR↑	SPL↑	SR↑	SPL↑	SR↑
PONI (Ramakrishnan et al., 2022)	Yes	Yes	No	12.1	31.8	-	-	-	-
ProcTHOR (Deitke et al., 2022)	Yes	Yes	Yes	-	-	31.8	54.4	28.8	65.2
ZSON (Majumdar et al., 2022)	No	Yes	Yes	4.8	15.3	12.6	25.5	-	-
ProcTHOR – ZS (Deitke et al., 2022)	No	No*	Yes	-	-	7.7	13.2	**23.7**	**55.0**
CoW (Gadre et al., 2022)	No	No	No	3.7	7.4	-	-	16.9	26.7
ESC (Ours)	No	No	No	**14.2**	**28.7**	**22.3**	**39.2**	22.2	38.1

4.2. Baselines

We compare our ESC method with the following two state-of-the-art (SOTA) methods for zero-shot object navigation:

- **ZSON** (Majumdar et al., 2022) uses a CLIP encoder to project the object and image goal into a same embedding space and feed the object goal embedding into an image goal navigation (Mezghani et al., 2021) network.
- **CLIP on Wheels (CoW)** (Gadre et al., 2022) use gradient-based visualization technique (Grad-CAM (Selvaraju et al., 2017)) on CLIP to localize goal object in egocentric view, and a frontier-based exploration technique (Yamauchi, 1997) for zero-shot object goal navigation.

In addition, we also compare our method with the following supervised methods:

- **PONI** (Maksymets et al., 2021) proposes a modular method that predicts goal object potential and explorable area potential to select a temporary goal from the semantic map, achieving SOTA results on MP3D.
- **ProcTHOR** (Deitke et al., 2022) synthesizes 10k indoor environments and performs large-scale ObjectNav training on those environments. Then it fine-tunes the agent on each specific dataset, achieving SOTA results on HM3D and RoboTHOR. We also compare with its zero-shot version in Table 1.

4.3. Implementation Details

There are several hyper-parameters in the ESC method. For the distance threshold d_f for selecting the closest frontier to explore, we use $d_f = 1.6m$ in all the experiments. For the threshold d_o determining whether a frontier is near an object, we fix $d_o = 1.6m$. For the threshold d_r determining whether a frontier is in a room, we fix $d_r = 0.6m$. We applied a weight of 1.0 for all PSL rules when only one of commonsense reasoning (object or room) was utilized. Moreover, we double the weight for the shortest distance rule in Eq. 6 to 2.0 when both levels of commonsense reasoning are employed.

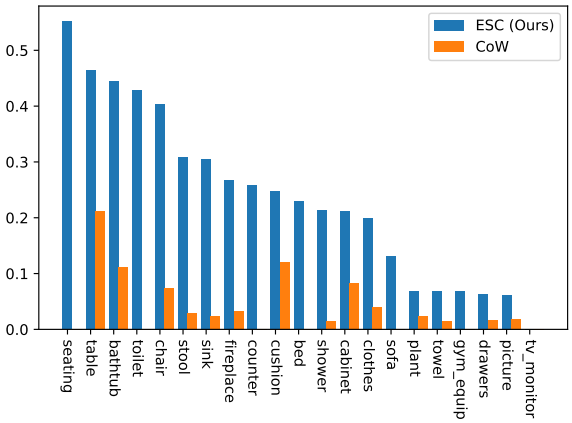

Figure 3. Comparison of the success rate of each goal category on MP3D between CoW (Gadre et al., 2022) and ESC.

5. Results and Analysis

5.1. Result Comparison with SOTA Methods

We compare the performance of our method with other zero-shot and supervised state-of-the-art (SOTA) methods in Table 1. Our ESC method significantly outperforms previous zero-shot methods on both SR and SPL metrics, *e.g.*, with 284% SPL and 288% SR relative improvements on MP3D over CoW that has the same evaluation setting with our method. Compared with ZSON (Majumdar et al., 2022), which trains the agent on HM3D datasets for the image-goal navigation task at a large scale, our method still outperforms it by a large margin. The zero-shot version of the ProcTHOR model is trained on ProcTHOR data and adapts to other datasets directly. Since the texture, style, and layout of the pre-training data of ProcTHOR are similar to RoboTHOR with the same simulator, the ProcTHOR agent achieves a much higher success rate on RoboTHOR dataset. However, ProcTHOR suffers from a severe performance drop on HM3D dataset, which is reconstructed from real-world architectures with photo-realistic images and on the Habitat simulator. ESC outperforms ProcTHOR on average on SR and SPL on these two datasets.

Moreover, all three other zero-shot methods suffer from in-

Table 2. Comparison between different detection models and different levels of commonsense reasoning on three datasets.

	MP3D			HM3D			RoboTHOR	
Commonsense Reasoning	SPL	SR	SoftSPL	SPL	SR	SoftSPL	SPL	SR
CLIP w/o Commonsense (CoW)	3.7	7.4	-	-	-	-	16.9	26.7
GLIP w/o Commonsense (GoW)	12.4	25.6	22.8	18.8	33.1	27.0	21.6	36.1
ESC w/o Room	13.8	28.3	23.7	21.8	**39.3**	30.6	21.4	35.6
ESC w/o Object	13.7	27.8	**24.1**	**22.3**	**39.3**	30.2	**23.1**	**39.3**
ESC	**14.2**	**28.7**	23.8	**22.3**	39.2	**31.1**	22.2	38.1

Table 3. Comparison between Deberta and ChatGPT on HM3D.

	Deberta			ChatGPT		
Model	SPL	SR	SoftSPL	SPL	SR	SoftSPL
Object	21.8	**39.3**	30.6	22.1	38.8	30.8
Room	**22.3**	**39.3**	30.2	21.6	38.0	29.7
Obj+Room	**22.3**	39.2	**31.1**	**22.4**	**39.0**	**31.1**

consistent performance when adapting to different datasets. This shows that their scene understanding and navigation policy are not generalized enough. Our ESC method, in contrast, performs consistently well on three datasets, demonstrating its strong generalizability.

What is more, our ESC method significantly reduces the gap between zero-shot methods and supervised methods on HM3D and RoboTHOR datasets, and even outperforms the supervised THDA method (Maksymets et al., 2021) on MP3D, which shows the great potential of zero-shot methods on object goal navigation tasks.

Fig. 3 illustrates a category-wise comparison between CoW and ESC on the MP3D dataset. It is evident that ESC outperforms CoW consistently on all the goal object types. Note that CoW fails on some goal objects with a strong tendency to appear or not to appear in specific rooms or in proximity to particular objects (e.g., 'toilet' and 'bed'), while our agent performs much better in those cases, indicating the efficacy of commonsense reasoning.[3]

5.2. Ablation Study

To demonstrate the efficacy of semantic scene understanding and commonsense reasoning, we design GLIP on Wheel (GoW). It uses a GLIP model for object detection and the vanilla fronter-based exploration method for exploration. As a replacement for commonsense reasoning in ESC, GoW always selects the closest frontier 1.6 meters away during exploration. Notice that GoW has the same navigation policy as ESC methods except the frontier selection policy.

Effect of semantic scene understanding and commonsense reasoning. In Table 2, GoW surpasses CoW on all metrics on MP3D and RoboTHOR, which demonstrates the effectiveness of open-world object grounding of GLIP. Furthermore, ESC further outperforms GoW on all the datasets and metrics, showing the superiority of commonsense reasoning compared with pure heuristic exploration.

Effect of different levels of commonsense reasoning. In Table 2, we also compare the performance of different levels of reasoning on three datasets. We remove object/room level common sense for comparison. From the results, we observe that both room and object reasoning improves over GoW, and room reasoning usually brings larger improvement than object reasoning. Using both room and object reasoning provides better results on MP3D and HM3D datasets. Due to the more random placement of objects in the RoboTHOR dataset, exploration without object reasoning performs the best, and incorporating object reasoning hurts the performance slightly.

Effect of different LLMs. Table 3 compares the performance of different LLMs for commonsense reasoning on HM3D. Both LLMs significantly improve the performance over GoW. ChatGPT performs similarly to Deberta, except for using room-level commonsense even without specific commonsense training. We mainly use Deberta in our framework due to its accessibility. See Appendix A.3 for more implementation details.

How commonsense reasoning helps exploration. In Table 4, we compare the exploration ability of GoW and ESC with different frontier selection strategies. GoW selects the closest frontier 1.6 meters away from the agent, while ESC selects the frontier based on the commonsense knowledge inferred from LLMs. First, we calculate the average distance of all the chosen frontiers to the closest target object of different methods. As shown in the first column of Table 4, the frontiers chosen by our ESC method are closer to the goal object on average, which indicates our method can perform better exploration consistently and help the agent get closer to the goal object.

Second, we demonstrate the error analysis of GoW and ESC models in Table 4. For failure navigation, we define three kinds of errors: *Detection error* is defined as the goal appearing in the vision, but the agent didn't correctly detect it,

[3]There are only 7 'TV_monitor' examples on MP3D, so its performance is not representative. ESC's SR for 'TV_monitor' on HM3D is 21.7%, which has 281 'TV_monitor' examples.

Table 4. Comparison of exploration efficiency with and without commonsense reasoning on HM3D dataset. FrontierDist measures the average distance between chosen frontiers and the closest goal object. The other three metrics is the error rate of different error types.

Model	FrontierDist (meter)	Exploration (%)	Detection (%)	Planning (%)
GoW	8.2	14.3	**40.6**	12.1
ESC	**7.6**	**10.6**	40.8	**9.5**

Table 5. Comparison of results from ADMM and one-hot constraint for solving PSL program on MP3D dataset. We use object reasoning in the comparison.

Solvers	SPL	SR	SoftSPL	Infer time
ADMM	12.9	27.0	22.1	2.13
One-hot	**13.7**	**27.8**	**24.1**	**0.25**

or the goal never appears, but the agent thought it detected a goal. *Planning error* is defined as the agent successfully detecting the target object but failing, or the agent never detecting the goal object and stuck within 1 meter for at least 400 steps, which indicates low-level navigation ability. *Exploration error* is an error that is not a planning or detection error, which means the agent never saw the goal object without stuck or false detection. The exploration error rate evaluates the ability to get close to the goal object.

In Table 4, we observe that the exploration error of our ESC method has the most decrease compared with GoW. This validates that our ESC method helps the agent better explore the environment and get close to the object. We also observe that most of the errors from both methods are detection errors, this indicates that leveraging limited labels to improve the zero-shot pre-trained VL models and a better strategy to transform the detection results into action are potential directions to improve zero-shot methods.

Effect of different PSL solvers. We compare the performance between ADMM and one-hot constraint for solving PSL optimization in Table 5. We find that the results of the one-hot constraint are slightly better than ADMM, and it runs much faster for frontier selection (0.25 vs 2.13 seconds for each PSL inference). Since the final choice of the agent is one-hot, the target variables that optimize Eq. 11 in the one-hot space achieve the best satisfaction of the rules among all the possible choices of the agent.

6. Related Work

Object Goal Navigation Recently, there have been mainly two lines of work in object goal navigation. Most of the current SOTA methods use a pre-trained visual encoder (He et al., 2016; Radford et al., 2021) to encode the egocentric images into feature vectors, then feed them into a navigation agent network trained by large-scale imitation learning or reinforcement learning (Ye et al., 2021; Maksymets et al., 2021; Khandelwal et al., 2022; Ramrakhya et al., 2022; Deitke et al., 2022; Chen et al., 2022b). The second line of work is to explicitly construct a semantic map and train a navigation policy based on the constructed semantic map from the training dataset to infer goal object location (Chaplot et al., 2020b;a; Min et al., 2022; Zheng et al., 2022). Compared with these methods, our visual understanding and navigation policy does not need data from a specific environment for training. Instead, we leverage the prompt-based text-image grounding model for scene understanding and commonsense knowledge in large language models to reason on both object and room levels in a zero-shot manner.

To solve the problems of supervised methods on generalization to new objects and environments, four recent works aimed at zero-shot object goal navigation (Gadre et al., 2022; Majumdar et al., 2022; Al-Halah et al., 2022; Deitke et al., 2022). Majumdar et al. (2022); Al-Halah et al. (2022) both train an image-goal navigation agent at scale in target environments and map the object goal to image-goal embedding space, which may not be generalized well to new datasets. Gadre et al. (2022) use GradCAM (Selvaraju et al., 2017) with CLIP to localize goal objects, and frontier-based exploration (Yamauchi, 1997) for zero-shot object goal navigation. But it only uses CLIP to localize the goal object, and the exploration decision is not conditioned on the scene context. In contrast, our work leverages a grounded vision-and-language pre-trained model to recognize all common objects and rooms. Our exploration and navigation decisions are made conditioned on these scene contexts via a large language model in a zero-shot manner.

Commonsense Reasoning in Embodied Agents Commonsense reasoning is an essential ability for AI to perform human-level intelligence in various tasks and has been introduced into embodied AI tasks. Chaplot et al. (2020b); Chen et al. (2022c); Chaplot et al. (2020a); Min et al. (2022) used in-domain data to train a navigation policy on a semantic map to help find objects in an environment. External object relation knowledge was also used for object navigation in small environments (Yang et al., 2019; Zeng et al., 2021) and embodied procedural planning (Lu et al., 2022). For household task completion, Zheng et al. (2022) incorporate different levels of commonsense into task completion process, and Sarch et al. (2022) leverage in-domain semantic prior for room rearrangement. In our work, we aim to transfer the commonsense knowledge in large language models into the object goal navigation task in a zero-shot manner by expressing the commonsense knowledge as first-order rules and encoding them into a declarative templating language—Probablistic Soft Logic to help better exploration.

Large Pre-trained Models for Embodied Agents Benefiting from large-scale pre-training, large pre-trained models have been shown to excel in vision and language tasks (Radford et al., 2021; Li* et al., 2022; Khashabi et al., 2022), and recently have been used for embodied AI tasks, including object navigation (Khandelwal et al., 2022; Gadre et al., 2022; Majumdar et al., 2022), task planning (Ahn et al., 2022; Chen et al., 2022a; Huang et al., 2022b;a; Min et al., 2022; Blukis et al., 2022), and vision-and-language navigation (Shen et al., 2021; Shah et al., 2022). Among these works, Shen et al. (2021); Khandelwal et al. (2022) utilize CLIP (Radford et al., 2021) vision embedding to improve the performance of object goal navigation and vision-and-language navigation. Majumdar et al. (2022) leverage CLIP to project image and text into one goal embedding space for zero-shot object navigation. For language models, Min et al. (2022); Blukis et al. (2022); Zheng et al. (2022); Sharma et al. (2022) fine-tune a pre-trained language model on annotated (task, sub-tasks) pairs to teach the language model for sub-task planning. Ahn et al. (2022); Chen et al. (2022a); Huang et al. (2022b;a) feed several examples as prompt to language models to decompose a high-level goal into executable steps for a robot. Our work, instead, focuses on efficiently finding a goal object in unseen environments. And we leverage the pre-trained location-related commonsense in large language models and object and room-level context detected by a pre-trained grounded vision-and-language model to guide exploration in a zero-shot manner.

7. Conclusion and Future Work

In this paper, we propose a zero-shot object navigation framework, ESC, that leverages the pre-trained knowledge of the language-image grounding model and large language model. We introduce commonsense into frontier-based exploration as a soft constraint via PSL. The experiment results illustrate the efficacy and generalizability of our methods from different perspectives.

Our work establishes new state-of-the-art and explores the direction of using pre-trained commonsense knowledge in LLMs for object navigation. Future work can try to acquire more commonsense from LLMs, like the spatial relation between rooms for object navigation, and acquire different knowledge from LLMs for other embodied AI tasks. What is more, ESC uses a fixed strategy to combine commonsense knowledge. Improving the fixed strategy or relaxing the zero-shot constraint to limited finetuning to learn a frontier selection strategy is also a potential direction.

Acknowledgements

We thank Xuehai He, Jialu Wang, and the anonymous reviewers for their valuable feedback on this work.

References

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R. J., Jeffrey, K., Jesmonth, S., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.-H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., Yan, M., and Zeng, A. Do as i can and not as i say: Grounding language in robotic affordances. In *arXiv preprint arXiv:2204.01691*, 2022.

Al-Halah, Z., Ramakrishnan, S. K., and Grauman, K. Zero experience required: Plug & play modular transfer learning for semantic visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17031–17041, June 2022.

Anderson, P., Chang, A. X., Chaplot, D. S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., and Zamir, A. R. On evaluation of embodied navigation agents. *CoRR*, abs/1807.06757, 2018. URL http://arxiv.org/abs/1807.06757.

Bach, S. H., Broecheler, M., Huang, B., and Getoor, L. Hinge-loss markov random fields and probabilistic soft logic. *J. Mach. Learn. Res.*, 18(1):3846–3912, jan 2017. ISSN 1532-4435.

Batra, D., Gokaslan, A., Kembhavi, A., Maksymets, O., Mottaghi, R., Savva, M., Toshev, A., and Wijmans, E. Objectnav revisited: On evaluation of embodied agents navigating to objects. *CoRR*, abs/2006.13171, 2020. URL https://arxiv.org/abs/2006.13171.

Blukis, V., Paxton, C., Fox, D., Garg, A., and Artzi, Y. A persistent spatial semantic representation for high-level natural language instruction execution. In *arXiv preprint arXiv:2107.05612*, 2022.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, jan 2011. ISSN 1935-8237. doi: 10.1561/2200000016. URL https://doi.org/10.1561/2200000016.

Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., and Zhang, Y. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

Chaplot, D. S., Gandhi, D., Gupta, A., and Salakhutdinov, R. Object goal navigation using goal-oriented semantic

exploration. In *In Neural Information Processing Systems (NeurIPS)*, 2020a.

Chaplot, D. S., Gandhi, D., Gupta, S., Gupta, A., and Salakhutdinov, R. Learning to explore using active neural slam. In *International Conference on Learning Representations (ICLR)*, 2020b.

Chen, B., Xia, F., Ichter, B., Rao, K., Gopalakrishnan, K., Ryoo, M. S., Stone, A., and Kappler, D. Open-vocabulary queryable scene representations for real world planning. In *arXiv preprint arXiv:2209.09874*, 2022a.

Chen, P., Ji, D., Lin, K., Hu, W., Huang, W., Li, T. H., Tan, M., and Gan, C. Learning active camera for multi-object navigation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=iH4eyI5A7o.

Chen, P., Ji, D., Lin, K., Zeng, R., Li, T. H., Tan, M., and Gan, C. Weakly-supervised multi-granularity map learning for vision-and-language navigation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022c. URL https://openreview.net/forum?id=gyZMZBiI9Cw.

Datta, S., Maksymets, O., Hoffman, J., Lee, S., Batra, D., and Parikh, D. Integrating egocentric localization for more realistic point-goal navigation agents. *CoRL*, 2020.

Deitke, M., Han, W., Herrasti, A., Kembhavi, A., Kolve, E., Mottaghi, R., Salvador, J., Schwenk, D., VanderBilt, E., Wallingford, M., Weihs, L., Yatskar, M., and Farhadi, A. Robothor: An open simulation-to-real embodied ai platform. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Deitke, M., VanderBilt, E., Herrasti, A., Weihs, L., Salvador, J., Ehsani, K., Han, W., Kolve, E., Farhadi, A., Kembhavi, A., and Mottaghi, R. Procthor: Large-scale embodied ai using procedural generation, 2022. URL https://arxiv.org/abs/2206.06994.

Gadre, S. Y., Wortsman, M., Ilharco, G., Schmidt, L., and Song, S. Clip on wheels: Open-vocabulary models are (almost) zero-shot object navigators. *arXiv*, 2022.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017. doi: 10.1109/ICCV.2017.322.

He, P., Gao, J., and Chen, W. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021.

Huang, W., Abbeel, P., Pathak, D., and Mordatch, I. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *arXiv preprint arXiv:2201.07207*, 2022a.

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., Sermanet, P., Brown, N., Jackson, T., Luu, L., Levine, S., Hausman, K., and Ichter, B. Inner monologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608*, 2022b.

Khandelwal, A., Weihs, L., Mottaghi, R., and Kembhavi, A. Simple but effective: Clip embeddings for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

Khashabi, D., Kordi, Y., and Hajishirzi, H. Unifiedqa-v2: Stronger generalization via broader cross-format training. *arXiv preprint arXiv:2202.12359*, 2022.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022.

Li*, L. H., Zhang*, P., Zhang*, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., Chang, K.-W., and Gao, J. Grounded language-image pre-training. In *CVPR*, 2022.

Lu, Y., Feng, W., Zhu, W., Xu, W., Wang, X. E., Eckstein, M., and Wang, W. Y. Neuro-symbolic causal language planning with commonsense prompting. *ArXiv*, abs/2206.02928, 2022.

Majumdar, A., Aggarwal, G., Devnani, B. S., Hoffman, J., and Batra, D. ZSON: Zero-shot object-goal navigation using multimodal goal embeddings. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=VY1dqOF2RjC.

Maksymets, O., Cartillier, V., Gokaslan, A., Wijmans, E., Galuba, W., Lee, S., and Batra, D. Thda: Treasure hunt data augmentation for semantic navigation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15354–15363, 2021. doi: 10.1109/ICCV48922.2021.01509.

Mezghani, L., Sukhbaatar, S., Lavril, T., Maksymets, O., Batra, D., Bojanowski, P., and Alahari, K. Memory-augmented reinforcement learning for image-goal navigation. *arXiv preprint arXiv:2101.05181*, 2021.

Min, S. Y., Chaplot, D. S., Ravikumar, P. K., Bisk, Y., and Salakhutdinov, R. FILM: Following instructions in language with modular methods. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=qI4542Y2s1D.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Gray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Ramakrishnan, S. K., Gokaslan, A., Wijmans, E., Maksymets, O., Clegg, A., Turner, J. M., Undersander, E., Galuba, W., Westbury, A., Chang, A. X., Savva, M., Zhao, Y., and Batra, D. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=-v4OuqNs5P.

Ramakrishnan, S. K., Chaplot, D. S., Al-Halah, Z., Malik, J., and Grauman, K. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *Computer Vision and Pattern Recognition (CVPR), 2022 IEEE Conference on*. IEEE, 2022.

Ramrakhya, R., Undersander, E., Batra, D., and Das, A. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *CVPR*, 2022.

Sarch, G., Fang, Z., Harley, A. W., Schydlo, P., Tarr, M. J., Gupta, S., and Fragkiadaki, K. Tidee: Tidying up novel rooms using visuo-semantic commonsense priors. In Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., and Hassner, T. (eds.), *Computer Vision – ECCV 2022*, pp. 480–496, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19842-7.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017. doi: 10.1109/ICCV.2017.74.

Shah, D., Osinski, B., Ichter, B., and Levine, S. Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action. 2022. URL https://arxiv.org/abs/2207.04429.

Sharma, P., Torralba, A., and Andreas, J. Skill induction and planning with latent language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1713–1726, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.120. URL https://aclanthology.org/2022.acl-long.120.

Shen, S., Li, L. H., Tan, H., Bansal, M., Rohrbach, A., Chang, K.-W., Yao, Z., and Keutzer, K. How much can clip benefit vision-and-language tasks? *arXiv preprint arXiv:2107.06383*, 2021.

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, 1997. doi: 10.1109/CIRA.1997.613851.

Yang, W., Wang, X., Farhadi, A., Gupta, A., and Mottaghi, R. Visual semantic navigation using scene priors. In *ICLR*, 2019.

Ye, J., Batra, D., Das, A., and Wijmans, E. Auxiliary tasks and exploration enable objectgoal navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16117–16126, October 2021.

Zeng, Z., Röfer, A., and Jenkins, O. C. Semantic linking maps for active visual object search (extended abstract). In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 4864–4868, 8 2021. doi: 10.24963/ijcai.2021/667. URL https://doi.org/10.24963/ijcai.2021/667. Sister Conferences Best Papers.

Zheng, K., Zhou, K., Gu, J., Fan, Y., Wang, J., Li, Z., He, X., and Wang, X. E. Jarvis: A neuro-symbolic commonsense reasoning framework for conversational embodied agents. *arXiv preprint arXiv:2208.13266*, 2022.

A. Implementation Details

A.1. GLIP Implementation Details

We use pre-trained GLIP-L (Li* et al., 2022) for all our experiments. For both object detection and room detection, we use 0.61 as the threshold of GLIP. For object detection in MP3D and HM3D dataset, we define the common indoor objects $\mathcal{O}_c$ as all the goal objects in MP3D dataset, which has included the goal objects in the HM3D dataset. Therefore, the object detection prompt for both datasets will be '*chair. table. picture. cabinet. cushion. sofa. bed. chest_of_drawers. plant. sink. toilet. stool. towel. tv_monitor. shower. bathtub. counter. fireplace. gym_equipment. seating. clothes.*'.

For object detection in the RoboTHOR dataset, we utilize '*Bed, Book, Bottle, Box, Knife, Candle, CD, CellPhone, Chair, Cup, Desk, Table, Drawer, Dresser, Lamp, Fork, Newspaper, Painting, Pencil, Pepper Shaker, Pillow, Plate, Pot, Salt Shaker, Shelf, Sofa, Statue, Tennis Racket, TV Stand, Watch*' as common indoor objects, which are selected from the furniture and object categories defined by RoboTHOR (Deitke et al., 2020) dataset. These objects, combined with the 12 goal objects will constitute the object prompt on the RoboTHOR dataset. For room detection on three datasets, we define the room prompt as 'bedroom. living room. bathroom. kitchen. dining room. office room. gym. lounge. laundry room.'.

A.2. PSL detailed explanation

During PSL inference, each ground rule creates one or more hinge-loss potentials defined over logical rules, which are relaxed using Łukasiewicz continuous valued logical semantics:

$$\phi(Y, X) = [max(0, l(Y, X))]^p \tag{12}$$

where l is a linear penalty function defined by PSL. We explain how the potential function is calculated and help determine unobserved variables through the example rule r in Eq. 4:

$$\begin{aligned} &\text{IsCooccur}(\text{Goal}, \text{Object}) \\ \wedge\ &\text{IsNearObj}(\text{Frontier}, \text{Object}) \\ \longrightarrow\ &\text{ChooseFrontier}(\text{Frontier}) \end{aligned} \tag{13}$$

This can be transformed into the following logic form:

$$\begin{aligned} &\neg(\text{IsCooccur}(\text{Goal}, \text{Object}) \\ \wedge\ &\text{IsNearObj}(\text{Frontier}, \text{Object}) \\ \vee\ &\text{ChooseFrontier}(\text{Frontier}) \end{aligned} \tag{14}$$

Given two grounded atoms $A_1, A_2 \in [0, 1]$, the formulas for the Łukasiewicz relaxation of the logical conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$) are as follows:

$$\begin{aligned} A_1 \tilde{\wedge} A_2 &= \max\{0, A_1 + A_2 - 1\} \\ A_1 \tilde{\vee} A_2 &= \min\{A_1 + A_2, 1\} \\ \tilde{\neg} A_1 &= 1 - A_1 \end{aligned} \tag{15}$$

From Eq. 15, by noting the ground atom as

$$\begin{aligned} \text{IsCooccur}(\text{Goal}, \text{Object}) &= x_1 \\ \text{IsNearObj}(\text{Frontier}, \text{Object}) &= x_2 \\ \text{ChooseFrontier}(\text{Frontier}) &= y_1 \end{aligned} \tag{16}$$

we can calculate the true value of a ground rule as :

$$\begin{aligned} &\min\{1, (1 - (x_1 + x_2 - 1)) + y_1\} \\ &= \min\{1, 2 - (x_1 + x_2) + y_1\} \end{aligned} \tag{17}$$

The *distance to satisfaction* of the rule is defined as

$$\begin{aligned} &\phi(x_1, x_2, y_1) \\ &= 1 - \min\{1, 2 - (x_1 + x_2) + y_1\} \\ &= \max\{0, x_1 + x_2 - y1 - 1\} \end{aligned} \tag{18}$$

For instance, when $x_1 = 0.8$ and $x_2 = 0.8$, $\phi(x_1, x_2, y_1) = 0$ only when $y_1 > 0.6$. And this will force the agent to choose a frontier when it's next to an object that the goal object is likely to be near.

A.3. LLM Details

For commonsense reasoning in object navigation, we choose the following two language models for zero-shot navigation inference.

Deberta v3 (He et al., 2021) (which we will refer to as Deberta for simplicity hereafter) utilizes replaced token detection objective and disentangled attention mechanism for pre-training and achieved SOTA performance on a wide range of natural language understanding tasks. The Deberta v3 is first pre-trained on a commonsense reasoning QA (CSQA) dataset. During pre-training, each 'question+candidate answer' pair in the dataset is fed into the Deberta model and gets the embedding $\mathbf{v} \in \mathcal{R}^d$ of [CLS] token from the output, which is then projected to a score s_i with learned weights.

During inference, we design two questions: 'What is a $GoalObject$ likely to be near?' and 'If you want to find a $GoalObject$, where should you go?' for object reasoning and room-level reasoning. The candidate rooms and objects in $\{o_g\} \bigcup \{o_c\}$ will be the candidate answers. The predicted scores s_i will be linearly normalized into $0, 1$ and be the values of IsCooccur in the rules.

ChatGPT (Ouyang et al., 2022) is a recently release pre-trained conversational LLM. It is capable of answering questions in a reasonable way. For ChatGPT, we input the language prompt as 'Among P_r / P_o, can you give the scores

Algorithm 1 Navigation algorithm

Input: Goal G
Initialize: frontier $F = None$, navigation map M_{nav}, semantic map M_{sem}, map update module MAP, GoalDetected
Object reasoning $\{R^o_{G,i} = \text{LLM}(G, o_i, P^{LLM}_o)\}$
Room reasoning $\{R^r_{G,i} = \text{LLM}(G, r_i, P^{LLM}_r)\}$
Look around and initialize M_{sem}, M_{nav}.
while not GoalDetected **do**
  $\{o_{t,i}, b^o_{t,i}\} = \text{GLIP}(I_t, P_o)$
  $\{r_{t,i}, b^r_{t,i}\} = \text{GLIP}(I_t, P_r)$
  $M_{sem} = \text{MAP}(\{o_{t,i}, b^o_{t,i}\}, \{r_{t,i}, b^r_{t,i}\}, M_{sem})$
  $M_{nav} = \text{MAP}(\{I_{t,i}\}, M_{nav})$
  if G in $\{o_{t,i}\}$ **then**
    GoalDetected = True
    break
  end if
  if reached F or F is $None$ **then**
    F = PSL($\{R^o_{G,i}\}, \{R^r_{G,i}\}, M_{sem}, M_{nav}$)
  else
    Navigate to F
  end if
end while
if GoalDetected **then**
  Navigate to G
end if

of likelihood to find a $GoalObject$ inside / nearby?'. Here P_r, P_o stands for room prompts and object prompts of GLIP respectively. The model will generate a series of scores that can be directly used as the values of IsCooccur.

A.4. Navigation Algorithm and Deterministic Policy

The navigation policy is illustrated in Alg. 1. The agent will first perform object-level and room-level reasoning about the goal object. Then it will look around and initialize the semantic map and navigation map. During the navigation process, it will perform semantic scene understanding with GLIP at each step and update the information on the semantic/navigation map. It will select frontiers using PSL based on commonsense reasoning, semantic map, and navigation map. After the agent detect a goal object, it will directly navigate toward it.

Under the general navigation policy, we also design several local policies to address the specific constraints encountered in different datasets and benchmarks. For MP3D and HM3D, the depth input is limited to 5 meters. Therefore, we design a long-distance goal policy to deal with the situation where a detected object is 5 meters away. If the object is a goal object, the agent will keep navigating in its direction until it's within 5 meters. If it's not a goal object, it will not be recorded in the semantic map.

For RoboTHOR, there is no GPS input to the agent. Therefore, when the agent is facing a wall and takes a move-forward action, it will believe it moves 0.25 meters while it actually stays in the original place. To mitigate this, we calculate the difference of the depth input of the last step and the current step to judge if the agent is moving.

B. Dataset Details

In MP3D (Chang et al., 2017) dataset, there are 21 target objects: `chair, table, picture, cabinet, cushion, sofa, bed, chest_of_drawers, plant, sink, toilet, stool, towel, tv_monitor, shower, bathtub, counter, fireplace, gym equipment, seating, clothes.` In HM3D (Ramakrishnan et al., 2021) dataset, there are 6 target objects: `chair, sofa, plant, bed, toilet, and tv_monitor.` In the RoboTHOR dataset, there are 12 target objects: `AlarmClock, Apple, BaseballBat, BasketBall, Bowl, GarbageCan, HousePlant, Laptop, Mug, SprayBottle, Television, Vase.`

C. More Results

C.1. Results per category

We demonstrate the success rate of each goal category on HM3D and RoboTHOR datasets here and compare our ESC with CoW method in Fig. 4. From the results, we first observe that ESC performs well in all the categories in HM3D dataset. ESC performs better than CoW on most of the object goals on RoboTHOR, except houseplant, basketball and spray bottle. There are mainly two reasons. First, CLIP still has great object localization ability on certain objects. Second, the smaller exploration space of RoboTHOR requires less exploration ability, so our method has less advantage.

C.2. Case study

To give a more intuitive view of how our model work in the object navigation process, we visualize an example in Fig. 5. The agent chooses 3 frontiers during the navigation process as the green points show. First, the agent detects a kitchen and performs commonsense reasoning that the toilet is not likely in the kitchen. Therefore it selects a frontier with a certain distance from the kitchen. When it gets out of the kitchen, it detects a table and several chairs, which are also not likely to be near a toilet. So, the agent selects two frontiers far from them, which avoids useless exploration and helps it find the toilet.

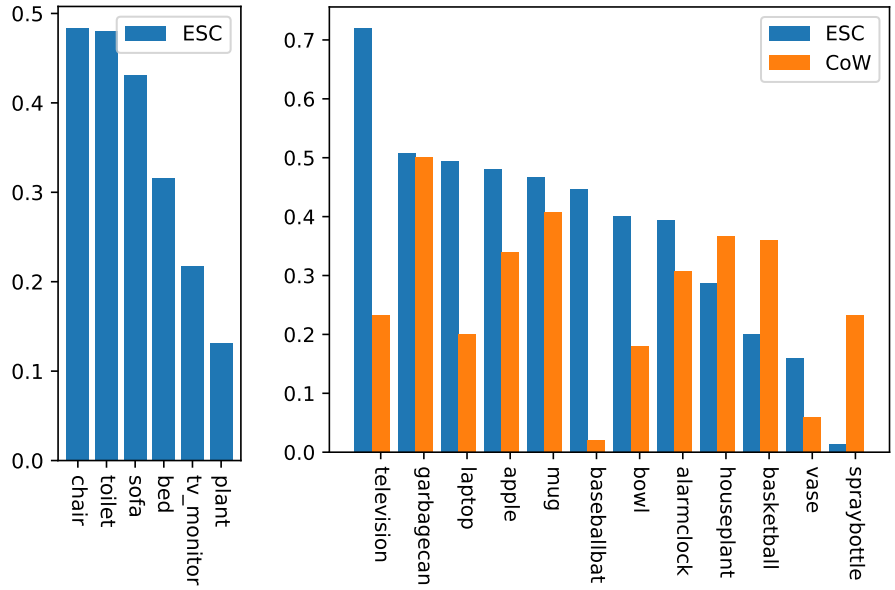

Figure 4. A demonstration of the success rate of each goal category on HM3D (left) and RoboTHOR (right) datasets of ESC method and CoW.

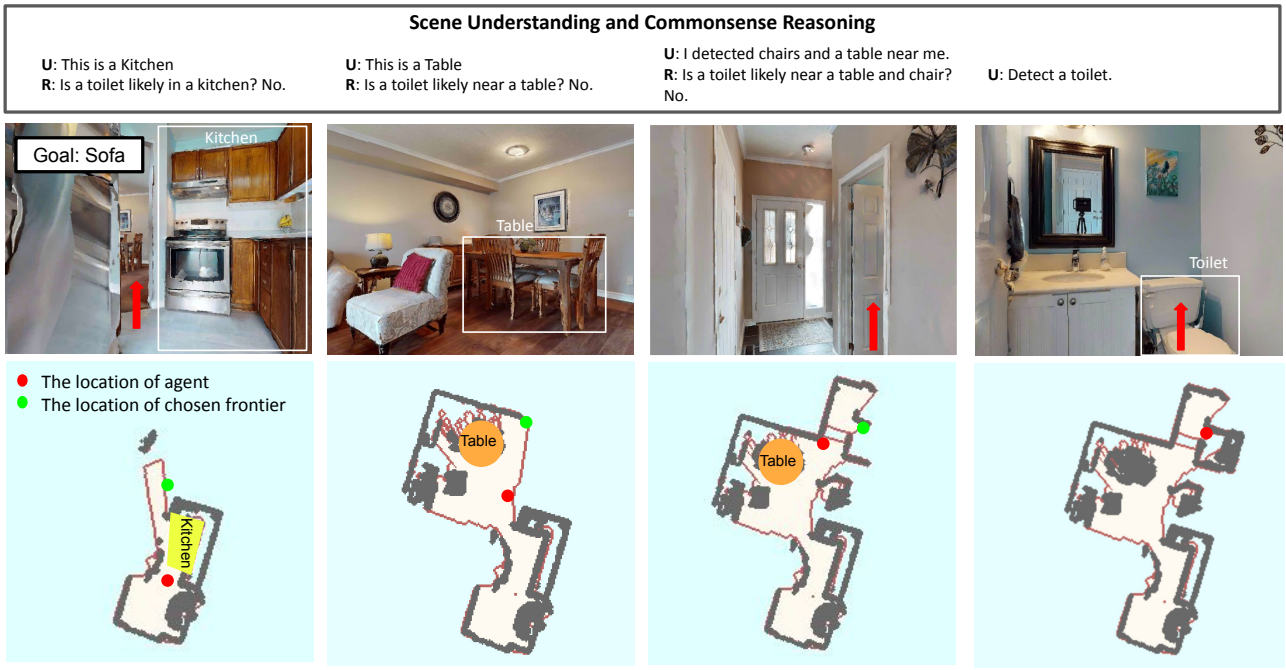

Figure 5. An example shows how commonsense reasoning helps the agent choose better frontiers that lead the agent to the goal 'toilet'. '**U**' means scene understanding and '**R**' means commonsense reasoning.