Can an Embodied Agent Find Your "Cat-shaped Mug"? LLM-Based Zero-Shot Object Navigation

Vishnu Sashank Dorbala, James F. Mullen, *Graduate Student Member, IEEE*, and Dinesh Manocha

***Abstract*—We present language-guided exploration (LGX), a novel algorithm for *Language-Driven Zero-Shot Object Goal Navigation* (L-ZSON), where an embodied agent navigates to an *uniquely described* target object in a *previously unseen* environment. Our approach makes use of large language models (LLMs) for this task by leveraging the LLM's commonsense-reasoning capabilities for making sequential navigational decisions. Simultaneously, we perform generalized target object detection using a pre-trained Vision-Language grounding model. We achieve state-of-the-art zero-shot object navigation results on RoboTHOR with a success rate (SR) improvement of over 27% over the current baseline of the OWL-ViT CLIP on Wheels (OWL CoW). Furthermore, we study the usage of LLMs for robot navigation and present an analysis of various prompting strategies affecting the model output. Finally, we showcase the benefits of our approach via *real-world* experiments that indicate the superior performance of LGX in detecting and navigating to visually unique objects.**

***Index Terms*—AI-enabled robotics, autonomous agents, domestic robotics, human-centered robotics.**

I. Introduction

HUMANS do not conform to preset class labels when referring to objects, instead describing them with free-flowing natural language. Robot agents performing *object goal navigation* in household environments must be able to comprehend and efficiently navigate to this seemingly infinite, arbitrary set of objects defined using natural language. For instance, a human may ask the robot agent to find its "cat-shaped mug." An agent trained on rigid class labels may interpret this as the human asking for a "cat" or a "mug" when the human is really referring to a mug in the shape of a cat. These types of unique objects typically lie outside the domain of the object categories commonly found in large image datasets such as ImageNet 21k [1] and OpenImages V4 [2]. Additionally, agents deployed in household environments may be required to navigate to these target objects without explicitly having a map or layout of the house available.

In the literature, this problem is known as the *L-ZSON* task [3]. *L-ZSON* or Language-Driven Zero-Shot Object Navigation involves the agent using a *freeform natural language description* of an object and finding it in a "zero-shot" manner, without ever having seen the environment *nor* the target object beforehand.

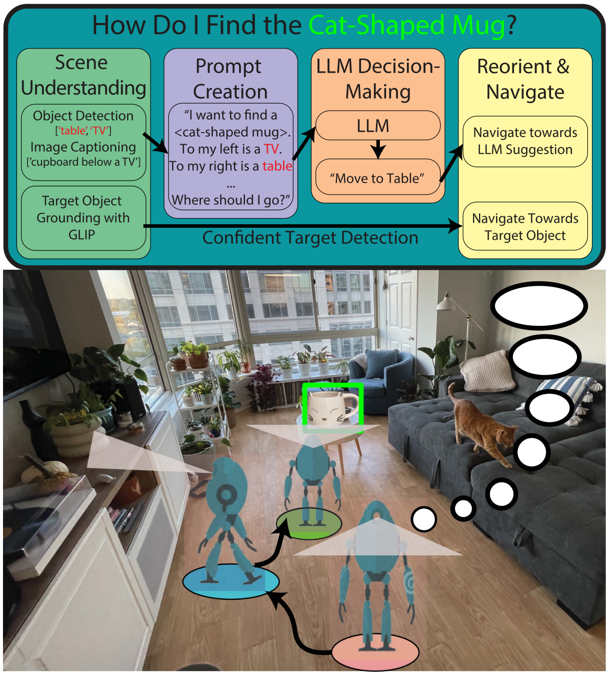
How Do I Find the Cat-Shaped Mug?
Scene Understanding
Object Detection
['table','TV']
Image Captioning
['cupboard below a TV']
Target Object Grounding with GLIP
Prompt Creation
"I want to find a <cat-shaped mug>. To my left is a TV. To my right is a table
...
Where should I go?"
LLM Decision-Making
LLM
"Move to Table"
Reorient & Navigate
Navigate towards LLM Suggestion
Navigate Towards Target Object
Confident Target Detection

Fig. 1. LLM-based navigation: Our method, LGX approaches the problem of language-driven zero-shot object navigation or L-ZSON. To navigate to and detect an unseen, arbitrarily described object class in an unknown environment, we first extract visual semantic information about the environment. This information is utilized to develop a prompt for the large language model (LLM), whose output provides us with either object sub-goals or cartesian directions to guide the embodied agent towards the target. Meanwhile, GLIP searches for the environment for the target object, which in this case is a "*cat-shaped mug*."

L-ZSON, like its parent Object Goal Navigation (ON) task, can be broken down into two key components — *Sequential Decision Making* and *Target Object Grounding*. The former refers to making exploratory decisions on the go, while the latter refers to locating and *grounding* a target object from agent perception. State-of-the-art approaches to solving Object Goal Navigation are based on fully supervised learning [4], [5], [6], which is not practical for an agent that is expected to detect arbitrarily described objects and perform consistently in dynamic real-world environments. Some recent works address this through the ZSON task [7], [8], [9], which enables the sequential decision making component to generalize to new locations, in part by removing environment-specific training. Even fewer recent works address the additional issue of generalizing target grounding to novel objects [10] as required by the L-ZSON task, and none study real-world test cases that contain

Manuscript received 12 July 2023; accepted 27 November 2023. Date of publication 25 December 2023; date of current version 22 March 2024. This letter was recommended for publication by Associate Editor S. Zhang and Editor H. Kurniawati upon evaluation of the reviewers' comments. This work was supported by the National Science Foundation Graduate Research Fellowship Program under Grant DGE 2236417. *(James F. Mullen contributed equally to this work.) (Corresponding author: Vishnu Sashank Dorbala.)*

The authors are with the University of Maryland, College Park, MD 20742 USA (e-mail: vdorbala@umd.edu; mullenj@umd.edu; dmanocha@umd.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3346800, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3346800

an abundance of unconstrained language. These works on the L-ZSON task utilize large-scale pre-trained models such as CLIP [11] and GLIP [12] to perform zero-shot open-vocabulary target grounding in the wild.

The downstream transfer of such 'foundation models' [13] has shown great improvement in various vision and language tasks such as image captioning [14], question answering [15], and the open-vocabulary target grounding [10] required in L-ZSON. However, transferring foundation models for use in the sequential decision making component of L-ZSON is more difficult, as unlike the vision tasks, this involves some form of *experiential* decision-making as the agent continuously interacts with the environment. Exploiting the implicit knowledge contained by these models to compose robot actions presents a unique challenge.

Adding to the challenge of the L-ZSON task, evaluating performance is difficult due to a lack of adequate simulation environments. Those designed for object navigation tasks, including RoboTHOR [16] and AI Habitat [17], only contain common day-to-day household objects described using simple language (eg. Mug, Table, Bed). However, this neglects the freeform natural language descriptions humans tend to use when talking to agents [18], an integral component of the L-ZSON task. As such, real-world experimentation on diverse objects is necessary to evaluate performance on the L-ZSON task.

Main Contributions: Motivated by the challenges above, we present **LGX**, or *Language-Guided Exploration*, a novel approach that leverages the implicit knowledge of large language models (LLMs) and pre-trained vision and language models to tackle the L-ZSON task.

In this work, we make use of large language models (LLMs) and Vision-Language (VL) Models to address generalizability issues that hinder the performance of both the sequential decision making and target object grounding components of L-ZSON. As LLM's rely on the prompts being used [19], we study the influence of prompt formulation via in-context learning [20] and present a case-by-case analysis of the effect of various prompt types. Additionally, we analyze the usage of VL models for Target Object Grounding and show improved performance with unique object references. We make the following contributions:

1) We present LGX, a novel approach to tackle L-ZSON, a language-guided zero-shot object goal navigation task (see Fig. 1). Our approach localizes objects described by unconstrained language by making use of large-scale Vision-Language (VL) models and leverages semantic connections between objects built into large language models (LLMs). Specifically, we study the implicit commonsense-reasoning capabilities of LLMs in assisting the sequential navigational decisions necessary to perform zero-shot object navigation.
2) Our approach utilizes visual scene descriptions of the environment to *formulate prompts* for LLM's, the output of which drives our navigation scheme. We study various types of prompts and provide insights into successfully using these prompts for robot navigation.
3) Our approach shows a 27% improvement on the state-of-the-art zero-shot success rate (SR) and success weighted by path length (SPL) on RoboTHOR.
4) Finally, we also present a transfer of our method onto a real-world robotics platform and study the various complexities involved in this setting. To the best of our knowledge, ours is the first approach to evaluate the performance of L-ZSON methods in the real world.

II. Related Work

A. Language-Guided Robotics

Using language to guide robots is a popular task in literature, with work ranging from using generalized grounding graphs [21] for robot manipulation [22] to performing language-guided navigation [23], [24]. Thomas et al. in [25] presents an approach to parse unconstrained natural language via a systematic probabilistic graph-based approach. More recent work tackling this problem by Jesse et al. [26], [27] and Gao et al. [28] has explored the use of human-robot dialogue to gather relevant information for completing tasks. Parsing unconstrained natural language is very relevant in our work, and we are motivated by the techniques developed by these letters.

B. Language-Driven Zero-Shot Navigation

Recent works have attempted to use CLIP [11] for performing zero-shot embodied navigation. CLIP is a large pre-trained Vision-Language model that is capable of zero-shot object detection. Dorbala et al. in [29] use CLIP to perform Vision-and-Language navigation in a zero-shot manner, while Gadre et.al in [10] have used it to perform object goal navigation. Both these works work under the assumption of unseen environments.

L-ZSON introduced by Gadre et al. in [3] approaches the problem of zero-shot object navigation, using uncommon target objects. They obtain a baseline for this task using OWL-ViT, a finetuned vision transformer for object grounding, and frontier-based exploration (FBE) [30]. In contrast, our approach uses GLIP [12], a pre-trained VL model for zero-shot object grounding. To explore the environment, we incorporate GPT-3 [20], an LLM, to make navigational decisions.

C. LLMs for Language-Guided Navigation

The adaptation of large language models in robotics has recently been garnering interest. Several recent works [31], [32], [33] have also used LLMs specifically for their planning capabilities. [34], [35] both use the LLM as a source of commonsense knowledge for the planning task alongside a classical, knowledge-based task planning system and Monte-Carlo tree search, respectively

Note we are different from these works as they utilize the LLM for *planning objectives* while we utilize the LLM for *environment exploration.* We present more comparisons and related work in our full technical report.[1]

III. Solving L-ZSON Using Language-Guided Exploration

A. Method Overview

We present an overview of our method in Fig. 2. Our approach uses a Large Language Model (LLM) to predict where the agent needs to navigate. To do this, we first extract contextual cues from the scene in the form of object labels or scene captions. Either of these cues are then used to devise a prompt asking the LLM about which how the agent should proceed to explore the

[1] [Online]. Available: https://arxiv.org/abs/2303.03480.

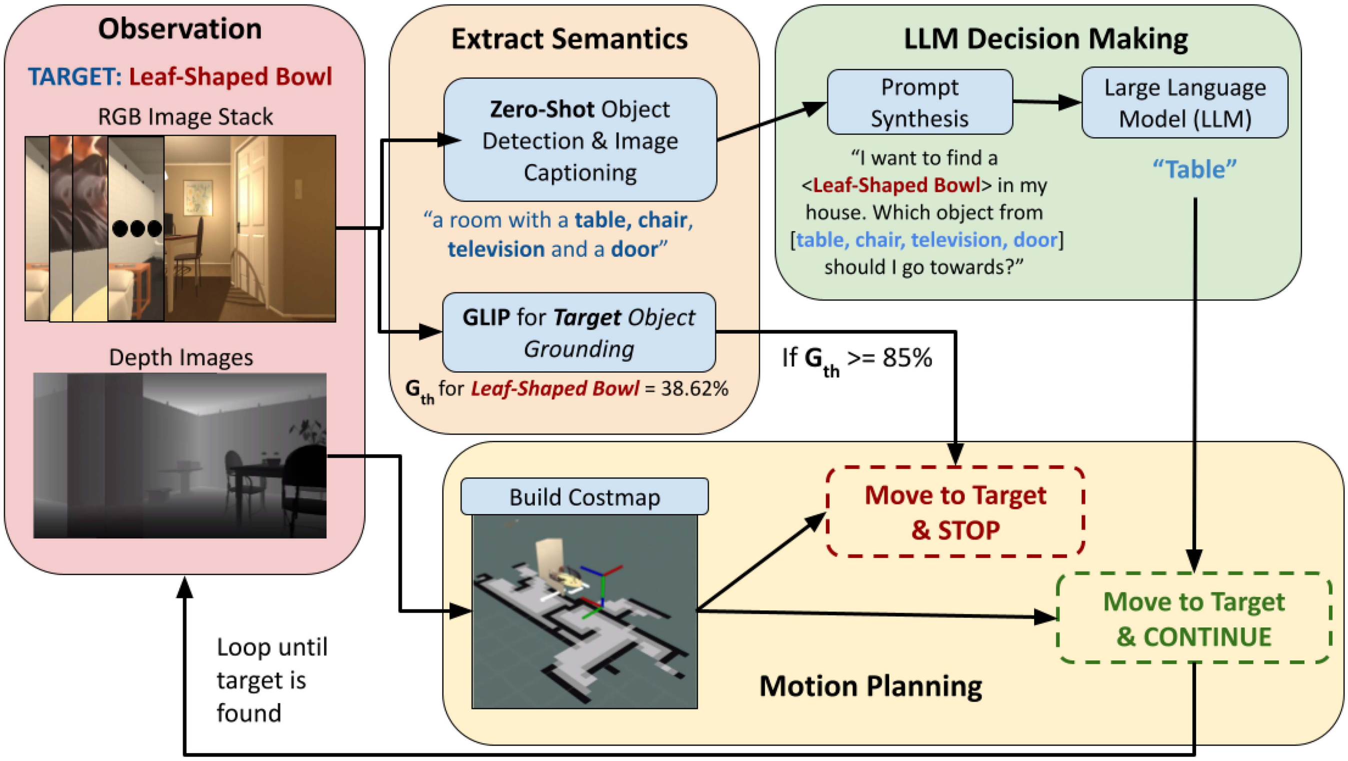
Observation
TARGET: Leaf-Shaped Bowl
RGB Image Stack
Depth Images
Extract Semantics
Zero-Shot Object Detection & Image Captioning
"a room with a table, chair, television and a door"
GLIP for Target Object Grounding
G_th for Leaf-Shaped Bowl = 38.62%
LLM Decision Making
Prompt Synthesis
Large Language Model (LLM)
"I want to find a <Leaf-Shaped Bowl> in my house. Which object from [table, chair, television, door] should I go towards?"
"Table"
If G_th >= 85%
Build Costmap
Move to Target & STOP
Move to Target & CONTINUE
Motion Planning
Loop until target is found

Fig. 2. Overview of our approach. We first gather observational data from the environment by performing a 360° rotation to obtain depth and RGB images around the agent. The RGB images give us semantic information about the objects in the agent's view, while the depth image allows us to create a costmap. We then synthesize prompts for the LLM by utilizing the extracted object labels. Finally, the LLM drives the navigational scheme by producing an output from the object list, which tells the agent which direction to head towards. Simultaneously, we attempt to ground the target object in the scene with GLIP. When the target is found, we exit the decision making loop and navigate directly to it.

environment. The LLM uses its commonsense knowledge about object relationships in the environment to provide the agent with a direction or object for it to move towards.

Simultaneously, we use a Vision-Language model, GLIP [12] to obtain a target object grounding score, which gives us the confidence of the described target being in the scene. Once the confidence meets a threshold G_{th}, the agent assumes the target object to be in its egocentric view. An episode is rendered successful if the target object is in the agent's view while performing rotate-in-place.

B. Scene Understanding

In each run, the agent observes the environment, gathering RGB and depth images for inspection. During each observation, we have the robot rotate in place 360 degrees, taking images at a set resolution r. This leaves us with $360/r$ RGB images, I_r, and depth images I_d. I_d is used to construct a 2D costmap of the environment. Every image in I_r is then fed into either an object detection or an image captioning model. Both these models give us different results, which we discuss in the experimentation section. For object detection, we use YOLO [36], which contains common household classes, while BLIP [37] gives us image captions. BLIP produces descriptive captions C of the environment, while YOLO gives us a list of objects O around the agent that it can potentially navigate towards. We chose either C or O to as part of our prompt to the LLM.

The rotate-in-place at each step allows the agent to fully observe its surroundings, giving the LLM enough information to make a fully informed navigation decision from the agent's current position in the environment. Without it, the agent would proceed toward seen objects over unknown space, even if none of the seen objects were related to the goal object, o_g. For example, if o_g is a "blue pillow," but it is initialized facing a kitchen and we see objects such as "microwave," "mug," and "table," the robot will proceed to explore near those objects because it does not know that directly behind it is a "bed" or a "couch", which is potentially where the pillow might be.

Simultaneously, while performing the full circle rotation, the agent uses I_d to construct a costmap of the environment. We use RTABMAP [38] that uses visual correspondences along with depth information from the standard costmap_2 d ROS package [39], to compute the costmap. Once a navigational decision is made by the LLM by providing either an object or a direction (depending on if C or O is passed to the input), we reorient the agent accordingly and randomly choose a point in the cost map along the agent's egocentric field of view. The costmap allows us to avoid obstacles while exploring the environment.

We use GLIP for target object grounding. During each rotation step the agent takes, the collected RGB images are passed through GLIP along with the target object as a prompt. When the grounding accuracy of GLIP is beyond a threshold G_{th}, we assume that the target object is in view of the agent, which triggers a STOP signal. If not, the agent continues exploring till n_r number of rotate-in-place turns. The episode is rendered successful if the ground truth target object lies in the view.

G_{th} and n_r are hyperparameters that are empirically chosen from ablation experiments. For selecting G_{th}, we ablate with

TABLE I
ABLATIONS ON G_{th} ON THE ROBOTHOR VALIDATION SET: G_{th} IS THRESHOLDED USING EMPIRICAL EVIDENCE FROM ABLATIONS

G_{th}	SR (%)	SPL (%)
0.6	13.8	7.2
0.75	20.3	10.8
0.8	32.5	18.7
0.85	**35.0**	**21.9**
0.95	18.0	11.3

A low value produces many false positives, leading to poor performance. Conversely, a high value rejects many potentially successful cases.

various threshold values in an environment, picking the one with the highest success rate (refer Table I). n_r is chosen based on the size of the environment.

C. Intelligent Exploration With large language models

We utilize the extracted semantics to devise a prompt for our LLM, GPT-3. There are two scenarios we explore,

1) **YOLO → LLM** : In this case, we utilize the list of objects O that YOLO detects around the agent to synthesize a prompt. For improving object detection, we set the rotate resolution r to a lower value here.
2) **BLIP → LLM**: Here, we utilize image captions C generated from the previous step to create a prompt. The rotate resolution r is set to 90 here, referring to either of the 4 directions (Left, Right, Front, Back) that the agent can take while exploring.

The LLM output upon using the YOLO + LLM approach gives us an object from O to navigate towards. The agent then reorients itself towards this object. While using BLIP + LLM, the output gives us a direction towards which the agent reorients itself. When the LLM output does not follow these expected outcomes, the agent chooses a random direction.

D. Goal Detection and Motion Planning

While completing the 360° rotation-in-place, we also run GLIP with the goal object o_g as its target label. Should GLIP find the target object o_g with a high enough confidence G_{th}, we terminate the exploration loop with the target in front of the agent. We then use the simulator as a ground truth to identify if the target actually lies in front of the agent, thus determining our success case. If GLIP does not find the target, we continue the LLM-based exploration.

At each step, we first orient the agent based on the LLM's decision. We then use the constructed cost map (refer Section III-B) to pick a point at a fixed exploratory distance e_d in the direction that the agent is facing. We then use the standard ROS *move_base* package to avoid obstacles.

IV. Analyzing Our Approach

A. Using GLIP for Zero-Shot Detection

Open-vocabulary (OV) grounding models have demonstrated strong zero-shot performance to object-level recognition tasks and proved to generalize well across numerous data sources. In our approach we use GLIP for grounding target objects of interest. It gives us a bounding box that allows us to localize a direction for navigation in the agent's field of view. GLIP outputs can be defined as

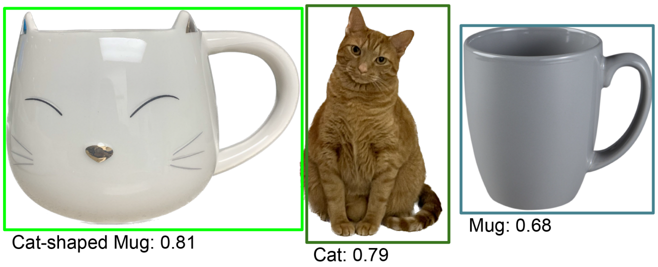

Fig. 3. An example of GLIP output when fed with the input string "Cat-shaped mug. Cat. Mug" on the image given. GLIP can successfully locate a unique object, like a "cat-shaped mug" and differentiate between it and related objects like a cat or a mug.

$$\{o_{t,i}, b_{t,i}\} = GLIP(I_t, P_o) \tag{1}$$

where $o_{t,i}$ and $b_{t,i}$ are the object detections and bounding boxes respectively. I_t and P_o represent the input image and the input prompt, defining the objects of interest, respectively.

We chose GLIP as our grounding model, gauging its outcome to be superior after ablation experiments with other state-of-the-art OV detection models including OWL-ViT [40] and Object-Centric OVD [41] and Detectron2 [42]. An example of the usefulness of GLIP for L-ZSON can be found in Fig. 3. It can not only identify the object defined using natural language, the "Cat-shaped mug", but differentiate it from related objects. Because of this behavior, running GLIP during our rotate-in-place procedure allows us to confidently detect our goal objects o_g irrespective of how they are described.

B. Examining LLM Prompts for Exploration

The outcome of GPT-3 is greatly influenced by the prompts that it is given. Since we directly use its commonsense reasoning capabilities (a *cat-shaped mug* is more likely to be near a table than a bed) for navigation, it is important for us to consider various prompting strategies. Should the LLM not provide us with a valid response, the agent moves in a random direction. We compare seven different LLM prompts which are variations of the following template :

> "*You are controlling a home robot. The robot wants to find a o_g in my house. Which object from $\{O\}$ should the robot go towards? Reply with ONE object from the list of objects.*"

We first explore how different points of view LLM feedback. These prompts are below:

- **Robot-Prompt**: "*You are controlling a home robot. The robot wants to find a o_g in my house. Which object from $\{O\}$ should the robot go towards? Reply with ONE object from the list of objects.*"
- **I-Prompt**: "*I want to find a o_g in my house. Which object from $\{O\}$ should I go towards? Reply in ONE word.*"
- **Third-Person-Prompt**: "*A o_g is in a house. Which object from $\{O\}$ is likely closest to o_g? Reply with ONE object from the list of objects.*"

Second, we vary the order of the information given to the prompt.

- **$\{O\}$-First-Prompt**: "*You are controlling a home robot. You must select one object from $\{O\}$ that the robot should go towards to try to find o_g in my house. Reply with ONE object from the list of objects.*"

- **Get-Closest-Prompt**: "*You are controlling a home robot. The robot wants to find a o_g in my house. Which object from $\{O\}$ is probably the closest to o_g? Reply with ONE object from the list of objects.*"
- **"ONE word"-First-Prompt**: "*Reply with ONE word. You are controlling a home robot. The robot wants to find a o_g in my house. Which object from $\{O\}$ should the robot go towards?*"

Last, we create prompts with natural language captions of the scene.

- **BLIP-Prompt**: "*I want to find a o_g in my house. In Front of you there is <caption >. To your Right, there is < caption>. Behind you there is <caption >. To your Left there is < caption>. Which direction from Front, Right, Behind, Left should I go towards? Reply in ONE word.*"

V. Experiments and Results

A. Experiment Setup

Simulation Setup: We use the RoboTHOR [16] validation set as a simulation environment for our experiments. It contains 1800 validation episodes with 15 validation environments. 12 different goal object categories are present. Each exploratory turn carries out the scene understanding procedure described earlier in Sections III-B and III-C. For each episode, we run LGX for n_r exploratory turns or until it detects the target object above the G_{th} threshold. These constraints form the *STOP* condition. n_r is set to 5, given the small environments in RoboTHOR, where the target object is usually within 10 meters of the spawning point. G_{th} is set to 0.85 for RoboTHOR after ablation experiments described in Table I. The exploratory distance e_d described in III-D is set to 5 m, ensuring significant changes to the scenery in RoboTHOR after motion.

Prompt Selection Setup: We run each of the 7 prompts described in the previous section on a subset of the 500 best and worst performing episodes.

Metrics: We report and compare Success Rate (SR) and Success Rate weighted by inverse path length (SPL) [43]. SR and SPL are the primary metrics used in both the Habitat and RoboTHOR challenges. For our prompt ablations, we define a new metric, **Prompt Success Rate (PSR)** as:

$$PSR = \frac{p_{suc}}{p_{total}} \tag{2}$$

where p_{suc} denotes the number of instances where the LLM chooses a valid response, and p_{total} denotes the total number of times the agent prompts the LLM. A valid LLM response is when it chooses either an object detected by the agent or a direction for navigation, depending on the semantic extraction scheme used.

Real World Setup: We conduct experiments with a TurtleBot 2 to validate two facets of LGX in the real world — i) the **LLM's Exploration Capability** and ii) the **GLIP-based open-vocabulary Grounding**.

To validate i), we look at a **two-phase approach** where the agent is required to travel from one room through a 'hallway' to reach a room containing the target object. In **Phase 1**, the agent performs rotate-in-place in the spawned room gathering information about objects around it. Since the target object is not present in this room, the LLM-output is expected to be *'hallway'*. In **Phase 2**, the agent is present in the hallway and is expected to choose the correct room to navigate to, given a set of common objects found in them (Refer Table II). The LLM-output is now expected to point towards the room that is most likely to contain the target object, based on commonsense knowledge ('remote control' near to 'couch'). This is explained in detail in Fig. 4.

To validate ii), we examine GLIP's accuracy in classifying unique household object classes. In order to do this, we first define and pick unique target objects, as well as common objects belonging to different household rooms. These are shown in Table II. A success case is defined by a successful GLIP detection of the target object.

TABLE II
Object Setup for Validating LLM Exploration

Room	Target Objects	Common Objects
Kitchen	Red Bull can, Stevia sugar packets	sink, fridge
Living Room	remote control, coffee table	couch, tv
Bedroom	bust, olive-colored jacket	bed, blanket
Office	silver pen, whiteboard	desk, computer

We define four household rooms populated with common and target objects that are likely to be found in them. Common objects are regular household items, while target objects are uniquely described with free-form language.

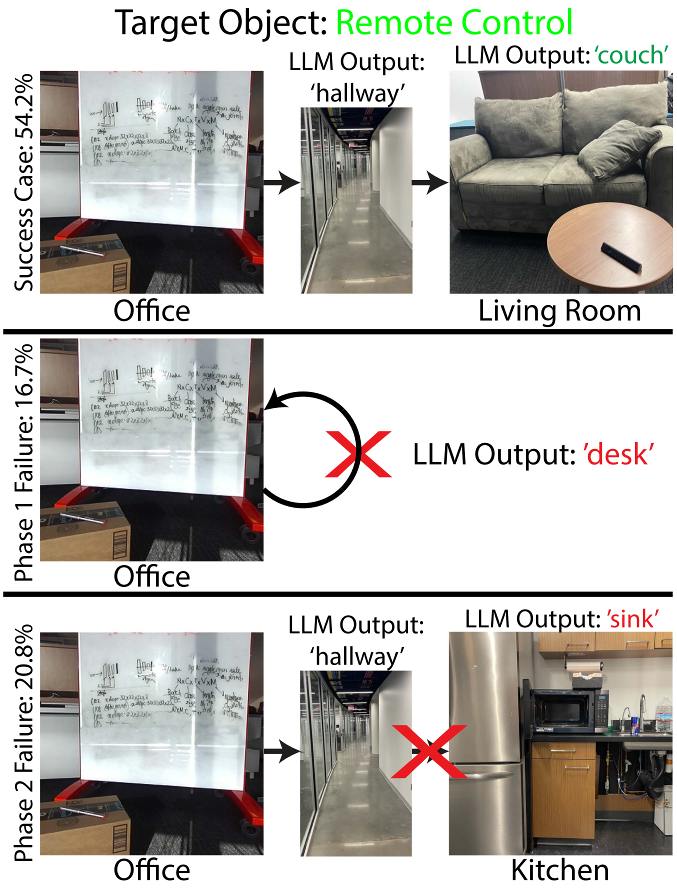

Fig. 4. To validate the LLM's exploration capability, we define a two-phase process. The target object is present in a different room, requiring the agent to navigate out of the current room into a "*hallway*." The LLM in LGX takes objects in the current room along with the hallway as input to the LLM. Not reaching the "hallway" is a *Phase 1* failure. For *Phase 2*, four possible rooms are visible and the agent must navigate to the room with the goal object. We pass a set of common objects for each room as shown in Table II as input to the LLM in LGX. Not choosing the correct room is considered a *Phase 2* failure.

TABLE III
ROBOTHOR RESULTS: OBSERVE THE IMPROVEMENT IN SUCCESS RATE (SR) AND SPL ON BOTH OUR APPROACHES OVER THE CURRENT SOTAS, COW, AND OWL

	RoboTHOR	
Model	SR (%) ↑	SPL (%) ↑
CoW (FBE + CLIP) [3]	15.2	9.7
OWL-CoW (FBE + OWL-VIT) [3]	27.5	17.2
GLIP on Wheels (FBE + GLIP)	33.2	20.3
LGX (BLIP → LLM + GLIP)	28.46	13.5
LGX (YOLO → LLM + GLIP)	**35.0**	**21.9**

This experimental setup validates our system against two main complicating factors of the real world, *free-form natural language*, and *partially-observable environments*. The *free-form natural language* problem is addressed through unique descriptions of each target object, which are not common visual class labels. The *partially-observable environments* component is addressed by conducting a two-phase exploration experiment, where only a few objects are visible in each room, replicating real-world homes.

B. Baselines and Ablations

We compare our method, LGX with two state-of-the-art methods and an ablative method:

CLIP-on-Wheels (CoW): [3] use Grad-CAM, a gradient-based visualization technique with CLIP [11] to localize a goal object in the egocentric view. CoW employs a Frontier-based Exploration technique for zero-shot object navigation.

OWL CoW: [3] utilizes the OWL-ViT transformer, in place of a CLIP model for target object grounding. This detector then replaces CLIP in the CoW method.

GLIP on Wheels (GoW): Where [3] utilizes the OWL-ViT transformer for its visual object grounding, we replace it with our GLIP based grounding system. This is also an ablation of our method without our LLM-based exploration mechanism.

Random with GLIP: As a baseline for our real-world analysis of LGX we also choose a random direction selector for exploration. The agent takes random decisions, replicating the behavior of an *'uninformed'* exploration method.

C. Comparison With Baselines in Simulation

We compare the performance of our method with other models set up for the L-ZSON task in Table III. Our method significantly outperforms the OWL CoW and the original CoW with an improvement in the success rate (SR) and SPL on both. LGX also showcases an improvement in the SR over GoW. This can be attributed to our improved LLM-based exploration scheme on top of using GLIP for target grounding.

In Fig. 5 we compare directly with CoW and OWL across the different target objects in RoboTHOR. Our method outperformed than both baselines across smaller objects like 'bowl' and 'vase.' Our performance was similar to OWL for larger objects like 'television.' These results showcase the performance deficit of CoW that is likely due to the inability of CLIP to localize the target object in the image effectively.

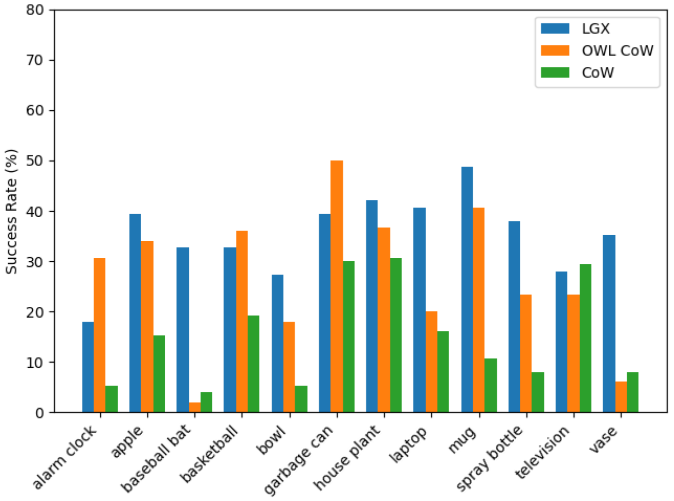

Fig. 5. Class breakdown of LGX versus the OWL CoW and original CoW on RoboTHOR. LGX provides a strong improvement in localizing the baseball bat, bowl, laptop, spray bottle, and vase classes. Similar performance is noted on larger classes like television and garbage can.

TABLE IV
COMPARISON OF SEVEN DIFFERENT PROMPTS ACROSS THREE AXIS OF CHANGE ON ROBOTHOR

	RoboTHOR	
Model	Success-Rate (%) ↑	PSR (%) ↑
BLIP-Prompt	29.2	**100**
I-Prompt	33.3	87.7
Robot-Prompt	**33.8**	71.1
Third-Person	31.3	99.4
$\{O\}$first	**33.8**	95.3
Get-Closest-Object	32.1	95.7
”ONE word” first	28.1	52.3

The object-based prompts (middle and bottom) outperform than natural language-based prompts.

D. Influence of Prompt Tuning Strategies

As seen in Table IV, the natural language-based prompts from BLIP perform worse relative to the object-based prompts despite a perfect PSR. We believe this is due to the limited action space when under the BLIP-based prompting scheme. The object-based prompts gave the LLM many different pathing options while the BLIP-based prompts were by definition associated with the four cardinal directions - potentially leading the agent towards a dead end. Additionally, we noted episodes where the LLM caused the agent to hop in a loop, continuously picking opposite directions.

No significant difference in task SR was captured over our second axis of prompt tuning denoting the perspective of the LLM relative to the robot. This is despite a wide array of PSRs for the different perspectives. The robot-perspective prompt exhibited the highest SR, but also the lowest PSR of the perspectives explored. Notably, when using the robot-perspective prompt, the LLM responded with 'no' or 'nothing' more frequently over the empty responses more commonly seen in the other prompts.

Across our changes to the structure of the prompt, there was no significant difference in SR for the object-set-first prompt or the get-closest-object prompt. However, the “ONE word” first prompt, denoting the placement of the “reply with ONE word”

TABLE V
In Our Real-World Experimentation, Our Model Significantly Outperformed Both Random and GLIP-on-Wheels Baselines

Approach (Navigation Decision + Grounding)	Success-Rate (%)↑
Random + GLIP	6.9
GLIP-on-Wheels (GOW)	27.8
LGX (YOLO → LLM + GLIP)	**54.2**

While all three methods utilize GLIP for target object detection, neither of the baselines integrates the scene context into the exploration phases of the task.

phrase before the rest of the prompt exhibited significantly worse SR and PSR. We believe this is due to the LLM no longer heeding this instruction when placed *before* the remainder of the prompt. The high PSR of the get-closest-prompt indicates that picking the likely closest object may be a simpler problem for the LLM to approach. Similarly, the high PSR of the object-set-first prompt indicates that the LLM could better reference the object-set when it was placed at the beginning of the prompt.

We believe that the insignificant performance differences in task SR, despite large changes in PSR, is another indicator of RoboTHOR providing a skewed basis for this type of context dependent, intelligent exploration of the scene.

E. Comparison With Baselines in the Real World

In our real-world experiments, we consider a two-phase approach as described in the setup earlier. As Table V indicates, LGX significantly outperforms chosen baselines, improving upon the SR of GoW by 26.4% and the SR of Random with GLIP by 47.3%. This resulted in GoW navigating to the correct room 33% of the time while Random with GLIP explored the objects in the starting scene with the same frequency as exploring the hallway. All of the success rates were also effected by the failure cases of GLIP, specifically false negatives when attempting to detect the ‘stevia sugar packets’ and false positives for the ‘Red Bull can’ and the ‘stevia sugar packets’ (see Fig. 6).

The LLM behavior in our method during our real world experimentation is characterized by three potential cases as shown in Fig. 4. The success case occurs 54.2% of the time and is a result of the robot agent successfully navigating from the starting room into the hallway, then into the room that contains the target, before detecting the target with GLIP.

In a Phase 1 failure case, the agent does not enter the hallway as the LLM believes one of the objects in the starting scene likely will lead to the target. One example of this we noted was when the target object is ‘*Red Bull can*’, the LLM would output ‘*desk*’ when the starting scene was the office. Although the target can is actually in the kitchen, it is plausible that it would lie on a ‘desk,’ explaining this output from the LLM. In the Phase 2 failure case, the agent enters a room that does not include the target object. This occurred 20.8% of the time with our method. This case is associated with the LLM poorly relating the target object other objects in the target room. One example of this case is the ‘olive-colored jacket’ which the LLM typically believed would be found near the ’desk.’ A breakdown of the system performance for each target object is found in Fig. 7. Our method failed to localize the ‘bust’ believing it to be associated with the ’desk.’ However, the relative success of the baselines indicates that GLIP succeeded in detecting the ’bust’ once inside the correct room.

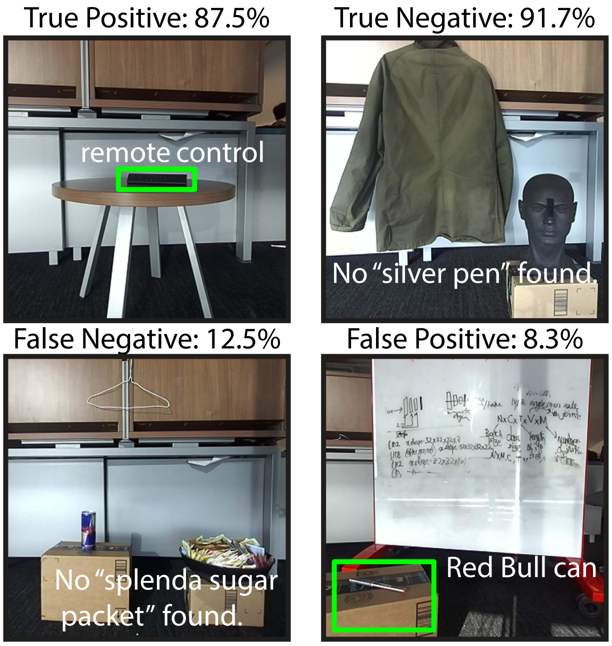

Fig. 6. Sampling of GLIP success and failure cases in our real-world experimentation. When the goal object was present in the scene, GLIP accurately detected it 87.5% of the time. Conversely, when the goal object was not present in the scene, GLIP falsely detected it 8.3% of the time.

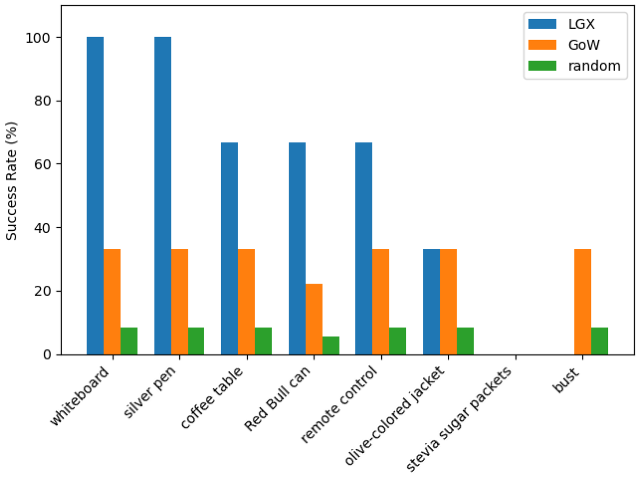

Fig. 7. Comparison of individual object success rates for our method and the baselines in our real-world study. Our method outperformed the baselines across the majority of targets, but notably failed to localize the “bust” object. None of the methods could localize the “stevia sugar packets” as GLIP failed to detect them.

More details and ablations about our experimentation can be found in the full technical report.[2]

VI. Limitations, Conclusions, and Future Work

In this letter, we present a novel algorithm for language-based zero-shot object goal navigation. Our method leverages the capabilities of large language models (LLMs) for making navigational decisions and open-vocabulary grounding models for detecting objects described using natural language. We showcase state of the art results on the RoboTHOR baseline, study

[2] [Online]. Available: https://arxiv.org/abs/2303.03480

the structure and phrasing of the LLM prompts that power our exploration, and validate our approach with real-world experiments.

Our method still includes a number of failure cases, especially when the LLM incorrectly localizes the target object. Future work should explore varying the context fed to the LLM by filtering the list of objects detected or providing a history of visited objects. Similarly, exploration of which objects produced an outsized effect would be useful. Future work should also look into improving the SR and SPL metrics such that they may be more informative for zero-shot navigation tasks.

Acknowledgment

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[2] A. Kuznetsova et al., "The open images dataset V4: Unified image classification, object detection, and visual relationship detection at scale," *Int. J. Comput. Vis.*, vol. 128, no. 7, pp. 1956–1981, 2020.

[3] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "CoWs on pasture: Baselines and benchmarks for language-driven zero-shot object navigation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 23171–23181.

[4] S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "PONI: Potential functions for objectgoal navigation with interaction-free learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18890–18900.

[5] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, "Think global, act local: Dual-scale graph transformer for vision-and-language navigation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 16537–16547.

[6] D. Batra et al., "ObjectNav revisited: On evaluation of embodied agents navigating to objects," 2020, *arXiv:2006.13171*.

[7] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, "ZSON: Zero-shot object-goal navigation using multimodal goal embeddings," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 32340–32352.

[8] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 4247–4258.

[9] S. Y. Min et al., "Object goal navigation with end-to-end self-supervision," 2022, *arXiv:2212.05923*.

[10] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Clip on wheels: Zero-shot object navigation as object localization and exploration," 2022, *arXiv:2203.10421*.

[11] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[12] L. H. Li et al., "Grounded language-image pre-training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10965–10975.

[13] R. Bommasani et al., "On the opportunities and risks of foundation models," 2021, *arXiv:2108.07258*.

[14] R. Mokady, A. Hertz, and A. H. Bermano, "ClipCap: Clip prefix for image captioning," 2021, *arXiv:2111.09734*.

[15] R. Dale, "GPT-3: What's it good for?," *Natural Lang. Eng.*, vol. 27, no. 1, pp. 113–118, 2021.

[16] M. Deitke et al., "RoboTHOR: An open simulation-to-real embodied AI platform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3164–3174.

[17] M. Savva et al., "Habitat: A platform for embodied AI research," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9339–9347.

[18] J. Hatori et al., "Interactively picking real-world objects with unconstrained spoken language instructions," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 3774–3781.

[19] Y. Zhou et al., "Large language models are human-level prompt engineers," 2022, *arXiv:2211.01910*.

[20] T. Brown et al., "Language models are few-shot learners," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1877–1901.

[21] S. Tellex et al., "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proc. AAAI Conf. Artif. Intell.*, 2011, vol. 25, pp. 1507–1514. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/7979

[22] J. S. Park, B. Jia, M. Bansal, and D. Manocha, "Efficient generation of motion plans from attribute-based natural language instructions using dynamic constraint mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 6964–6971.

[23] Z. Hu, J. Pan, T. Fan, R. Yang, and D. Manocha, "Safe navigation with human instructions in complex scenes," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 753–760, Apr. 2019.

[24] V. S. Dorbala, A. Srinivasan, and A. Bera, "Can a robot trust you?: A DRL-based approach to trust-driven human-guided navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 3538–3545.

[25] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions," in *Proc. IEEE/ACM 5th Int. Conf. Hum.-Robot Interact.*, 2010, pp. 259–266.

[26] J. Thomason et al., "Improving grounded natural language understanding through human–robot dialog," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 6934–6941.

[27] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, "Vision-and-dialog navigation," in *Proc. Conf. Robot Learn.*, 2020, pp. 394–406.

[28] X. Gao, Q. Gao, R. Gong, K. Lin, G. Thattai, and G. S. Sukhatme, "DialFRED: Dialogue-enabled agents for embodied instruction following," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 10049–10056, Oct. 2022.

[29] V. S. Dorbala, G. A. Sigurdsson, J. Thomason, R. Piramuthu, and G. S. Sukhatme, "Clip-NAV: Using clip for zero-shot vision-and-language navigation," in *Proc. Workshop Lang. Robot.*, 2022.

[30] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Automat., Towards New Comput. Princ. Robot. Automat.*, 1997, pp. 146–151.

[31] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 9118–9147.

[32] M. Ahn et al., "Do as I can, not as I say: Grounding language in robotic affordances," 2022, *arXiv:2204.01691*.

[33] W. Huang et al., "Inner monologue: Embodied reasoning through planning with language models," 2022, *arXiv:2207.05608*.

[34] Y. Ding et al., "Integrating action knowledge and LLMs for task planning and situation handling in open worlds," 2023, *arXiv:2305.17590*.

[35] Z. Zhao, W. S. Lee, and D. Hsu, "Large language models as commonsense knowledge for large-scale task planning," 2023, *arXiv:2305.14078*.

[36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[37] J. Li, D. Li, C. Xiong, and S. Hoi, "BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 12888–12900.

[38] M. Labbé and F. Michaud, "RTAB-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *J. Field Robot.*, vol. 36, no. 2, pp. 416–446, 2019.

[39] M. Quigley et al., "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, 2009, vol. 3, Art. no. 5.

[40] M. Minderer et al., "Simple open-vocabulary object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 728–755.

[41] H. Bangalath, M. Maaz, M. U. Khattak, S. H. Khan, and F. Shahbaz Khan, "Bridging the gap between object and image-level representations for open-vocabulary detection," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 33781–33794, 2022.

[42] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron 2," 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[43] P. Anderson et al., "On evaluation of embodied navigation agents," 2018, *arXiv:1807.06757*.