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

Vishnu Sashank Dorbala¹, James F. Mullen Jr¹, and Dinesh Manocha¹

Supplemental material including Full Technical Report, Code, Video are at https://gamma.umd.edu/LGX/

Abstract—We present LGX (Language-guided Exploration), 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 commonsensereasoning 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.

I. INTRODUCTION

Humans do not conform to preset class labels when referring to objects, instead describing them with freeflowing 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 our work, we aim to address these issues by tackling 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. The conventional Object Goal Navigation task [4], [5] requires the agent to locate an object from a predetermined category set within an unseen environment. Zero-Shot Object Navigation (ZSON) [6], [7] extends this task to identify a target without any environment-specific training. *L-ZSON* further extends ZSON to freeform natural language descriptions.

Simulation environments for object navigation tasks, including RoboTHOR [8] and AI Habitat [9], only contain common day-to-day household objects described using



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*".

simple language (eg. Mug, Table, Bed). However, humans tend to use unconstrained, natural language when talking to agents [10], leading to potential confusion in the agents' interpretation [11]. This problem becomes more apparent in the sim2real transfer of common object navigation models [12] that fail to understand the humans instructions. In this work, we seek to address this issue by carrying out real-world experiments with unique object references (eg. olive-colored jacket).

Common approaches to solving Object Goal Navigation are based on fully supervised learning [4], [13], which is not practical for an agent that is expected to detect arbitrarily described objects and perform consistently in dynamic real-world environments. While some recent works address generalizability to new locations via the ZSON task [14], even fewer address the issue of generalizing to novel objects [15] with the L-ZSON task, and none study real-world test cases that contain an abundance of unconstrained language.

¹The authors are associated with the University of Maryland, College Park, USA vdorbala@umd.edu, mullenj@umd.edu, dmanocha@umd.edu

These works utilize large-scale pre-trained models such as CLIP [16] and GLIP [17] to perform zero-shot openvocabulary object detection in the wild. The downstream transfer of such 'foundation models' [18] has shown great improvement in various vision and language tasks such as image captioning [19] and question answering [20]. This transfer to robotics is non-trivial however, as unlike vision and language, robot tasks usually involve 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.

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.

Object goal navigation including L-ZSON 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 percepts.

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 these components. As LLM's rely on the prompts being used [21], we study the influence of prompt formulation via incontext learning [22] 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:

- We present LGX, a novel approach to tackle L-ZSON, a language-guided zero-shot object goal navigation task. 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 stateof-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 [23] for robot manipulation [24] to performing language-guided navigation [25], [26]. Thomas et. al in [27] 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. [28], [29] and Gao et. al. [30] 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 papers.

B. Language-Driven Zero-Shot Navigation

Recent works have attempted to use CLIP [16] for performing zero-shot embodied navigation. CLIP is a large pre-trained Vision-Language model that is capable of zeroshot object detection. Dorbala et. al. in [31] use CLIP to perform Vision-and-Language navigation in a zero-shot manner, while Gadre et.al in [15] 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) [32]. In contrast, our approach uses GLIP [17], a pre-trained VL model for zero-shot object grounding. To explore the environment, we incorporate GPT-3 [22], an LLM, to make navigational decisions.

C. Language-Guided Scene Manipulation

Language-guided scene manipulation is a popular task, requiring agents to process natural language instructions to manipulate objects in the environment. The authors of [33]–[35] present various semantic-segmentation-based approaches for scene manipulation and object placement using language prompts as input. There also exist a broad range of classical approaches for language-guided manipulation [24], [36]–[39] that utilize different forms of grounding graphs to perform this task. Raman et. al in [40] tackle a unique manipulation task, synthesizing language that the robot can use to *explain* its failure to the user.

More recently, several works have presented novel approaches for language-conditioned manipulation task using pre-trained embeddings [41]–[44]. Our work focuses on language-guided exploration or LGX, as opposed to these manipulation tasks.

D. LLMs for High-Level Task Planning

Recent works have established LLM's as a tool for creating plans for the agent to execute, in breaking down high level instructions (such as "*Make me breakfast*") to lower-level tasks for the agent to perform (such as "*Make eggs*" and "*Bring juice*") [45]–[47]. Note that these approaches utilize it for generating and planning actions for the robot, but not explicitly for *exploration* in the wild. Information about the environment is already assumed to be known, and the LLM is used for breaking down complex tasks for the robot to execute.

Huang et al. in [45] looks at using an LLM to simplify complex tasks into "actionable sub-tasks", for a robot then act upon. Their task involves using pre-trained LLM's at a *planning level* in an interactible environment to determine



Fig. 2: An overview of our approach. We first gather observational data from the environment by performing a 360 degree 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.

how to interact with a known VirtualHome environment, and does not use the outcome to explore it.

SayCan in [46] utilizes an LLM in a similar manner to deduce actions that the agent can take, but additionally also uses an affordance function to ground the generated actions to the environment. They assume the object locations in the map to be known (see section D.2 in said paper) and solely tackle the "planning problem", with pick, go to and place actions already available. They do not consider a zero-shot exploration case like LGX to navigate in an unseen environment.

Huang, Xia, and Xiao et al., in [47], also follow a similar *planning* objective and utilize the LLM in tandem with human feedback to accomplish the instruction.

In contrast, our work uses an LLM as a prior not for planning, but rather to explore an unseen environment to predict the direction that the robot needs to take without requiring a map.

E. LLMs for Language-Guided Navigation

The adaptation of Large Language Models in robotics has recently been garnering interest. Several recent works [45]– [47] have also used LLMs specifically for their planning capabilities. Note we are different from these works as they utilize the LLM for *planning objectives* while we utilize the LLM for *environment exploration*.

A few works have looked at using generative models for

navigation, specifically, LM-Nav [48] and VLMaps [49]. Both these works look at solving the Vision-and-Language Navigation (VLN) problem, where the input is the unconstrained language describing a *path to the goal*. The latter uses GPT-3 [22] to obtain "landmarks" or subgoals, while the former focuses on using an LLM for "code-writing" [50]. In contrast, our focus is on translating visual scene semantics into input prompts for the LLM to obtain navigational guidance in the form of actions. We directly incorporate the LLM output into a sequential decision-making pipeline to drive our agent's navigation scheme.

III. SOLVING L-ZSON USING LANGUAGE-GUIDED EXPLORATION (LGX)

A. Method Overview

We present an overview of our method in Figure 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 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 [17] 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 [51], which contains common household classes, while BLIP [52] 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 [53] that uses visual correspondences along with depth information from the standard costmap_2d ROS package [54], 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 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** \rightarrow **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** \rightarrow **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

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

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 [55] and Object-Centric OVD [56] and Detectron2 [57]. An example of the usefulness of GLIP for L-ZSON can be found in figure 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



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.

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 og 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 og 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 og in my house. Which object from {O} is probably the closest to og? 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 og 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 [8] 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 5m, ensuring significant changes to the scenery in RoboTHOR after motion.

SR (%)	SPL (%)
13.8	7.2
20.3	10.8
32.5	18.7
35.0	21.9
18.0	11.3
	SR (%) 13.8 20.3 32.5 35.0 18.0

TABLE I: Ablations on G_{th} on the RoboThor Validation Set: G_{th} is thresholded using empirical evidence from ablations. A low value produces many false positives, leading to poor performance. Conversely, a high value rejects many potentially successful cases.

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) [58]. 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 Turtle-Bot 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 figure 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



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.

shown in Table II. A success case is defined by a successful GLIP detection of the target object.

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

TABLE II: Object Setup for validating LLM Exploration. 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.

This experimental setup validates our system against two main complicating factors of the real world, *free-form natural language*, and *partially-observable environments*. The *freeform 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.

	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 \rightarrow LLM + GLIP)	28.46	13.5
$LGX (YOLO \rightarrow LLM + GLIP)$	35.0	21.9

TABLE III: RoboTHOR Results: Observe the improvement in Success Rate (SR) and SPL on both our approaches over the current SOTAs, CoW and OWL.



Fig. 5: The 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.

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 gradientbased visualization technique with CLIP [16] to localize a goal object in the egocentric view. CoW employs a Frontierbased 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 LLMbased exploration scheme on top of using GLIP for target grounding.

	RoboTHOR	
Model	Success-Rate (%) \uparrow	PSR (%) \uparrow
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

TABLE IV: Comparison of seven different prompts across three axis of change on RoboTHOR. The object-based prompts (middle and bottom) outperform than natural language-based prompts.

In Figure 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.

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 robotperspective 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" 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

True Positive: 87.5%True Negative: 91.7%Image: state of the state

Fig. 6: An sampling of GLIP success and failure cases in our realworld 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.

Approach (Navigation Decision + Grounding) Success-Rate (9)	%)	↑
---	----	---

Random + GLIP	6.9
GLIP-on-Wheels (GOW)	27.8
LGX (YOLO \rightarrow LLM + GLIP)	54.2

TABLE V: In our real-world experimentation, our model significantly outperformed both random and GLIP-on-Wheels baselines. 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.

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 Figure 6).

The LLM behavior in our method during our real world experimentation is characterized by three potential cases as shown in Figure 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



Fig. 7: A 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.

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 Figure 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.

More details and ablations are presented in the discussion section.

VI. LIMITATIONS, CONCLUSIONS, AND FUTURE WORK

In this work we present a novel algorithm for languagebased 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 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 zeroshot navigation tasks.

VII. DISCUSSION

We present some additional experiments and analyses in this section, while also answering some related questions about the real-world applicability of LGX.



Fig. 8: Ablations on Ambiguity: We ablate on various openvocabulary models to assess the ambiguity of natural language when describing the object. Detectron2 [57], OCOVD [56] and OWL-ViT [55] (used in OWL-CoW) show false positives and not detecting the objects sometimes. On the other hand, Detic [59] and GLIP [17] work well for such open-vocabulary grounding. We decided to use GLIP for its ease of implementation, and experiments with using Detic in our framework are part of future work.

A. Real-World Challenges

Free-form natural language descriptions and partially observable environments make ObjectNav a complex task in the real-world. In zero-shot settings performance usually comes down to two main factors; **Exploration** and **Visual Object Grounding**, both of which face challenges in a real-world implementation. These are presented as follows

- Visual Object Grounding: Object grounding in simulation environments is significantly more constrained than in the real world. The simulator has a set number of available classes, and sometimes characteristics of these objects are not available to ground their natural language descriptions. Users in the real world are expected to use free-form language and the agent must adapt accordingly. We attempt to overcome this challenge by using an open-vocabulary grounding model, GLIP. GLIP grounds targets from the language used in the prompt, as opposed to finding all visual correspondences, allowing for the detection of objects described in free-form natural language. We explore GLIP's performance in the second part of our real-world experimentation described in Figures **5** and **6**.
- Exploration: While exploring real-world environments, the agent is bound to a dynamically changing partially observable environment, where the entire map and associated target objects may constantly keep changing. An uninformed exploration approach such as Frontier Based Exploration (FBE) can be inefficient, since the assumption of a static environment does not exist, causing the agent to re-explore regions that it has already seen. We tackle this issue using the LLM for informed decisionmaking, relying on commonsense knowledge to explore the environment. Unlike FBE which relies on selecting frontiers in free space, our LLM-based exploration is mapless, making informed choices about directions that the agent must take based on the environment surrounding it. In our real-world benchmarks, we validate the decision-making capability of the LLM via a two-phase approach presented in Section V.A and Figure 4.

B. Why GLIP?

Our approach uses GLIP [60], which is an openvocabulary object detector, which rather than proposing every object from an open-set, only detects objects mentioned in the text prompt. As an open vocabulary object detector, GLIP is known to take ambiguous text descriptions and ground those descriptions in the environment. LGX's ObjectNav performance is indeed sensitive to the performance of GLIP. Figure 3 in the manuscript highlights the superior capability of GLIP in tackling scenarios which are ambiguous to models without an open-vocabulary. When the user refers to a "catshaped mug", GLIP is successfully able to identify this unique object from a group of images with a cat and a mug. We present an ablation on the performance of other SoTA zero-shot open vocabulary object detectors - Detectron2 [57], OCOVD [56], OWL-ViT [55], Detic [59] and GLIP [60] in Figure 8 below. Figure 8 qualitatively shows a positive inference in that GLIP is successfully able to tackle ambiguous situations with free-form language, reinforcing our usage for open-vocabulary grounding in LGX.

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 2009 IEEE Conference on Computer Vision and Pattern Recognition, Jun. 2009, pp. 248-255.
- [2] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov et al., "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," International Journal of Computer Vision, 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," Dec. 2022.
- S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, and K. Grauman, "Poni: Potential functions for objectgoal navigation with [4] interaction-free learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 18 890–18 900.
- [5] D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, and E. Wijmans, "Objectnav revisited: On evaluation of embodied agents navigating to objects," arXiv preprint arXiv:2006.13171, 2020.
- [6] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," Advances in Neural Information Processing Systems, vol. 33, pp. 4247-4258, 2020.
- S. Y. Min, Y.-H. H. Tsai, W. Ding, A. Farhadi, R. Salakhutdinov, [7] Y. Bisk, and J. Zhang, "Object goal navigation with end-to-end selfsupervision," arXiv preprint arXiv:2212.05923, 2022
- [8] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford, L. Weihs, M. Yatskar, and A. Farhadi, "RoboTHOR: An Open Simulation-to-Real Embodied AI Platform," Apr. 2020.
- M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," Nov. 2019.
- [10] J. Hatori, Y. Kikuchi, S. Kobayashi, K. Takahashi, Y. Tsuboi, Y. Unno, W. Ko, and J. Tan, "Interactively picking real-world objects with un-constrained spoken language instructions," in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 3774-3781.
- [11] D. Whitney, E. Rosen, J. MacGlashan, L. L. Wong, and S. Tellex, "Reducing errors in object-fetching interactions through social feedback,' in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 1006-1013.
- [12] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, and S. Lee, "Sim-to-real transfer for vision-and-language navigation," in Conference on Robot Learning. PMLR, 2021, pp. 671-681.
- [13] 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 Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16537–16547.
- [14] A. Majumdar, G. Aggarwal, B. Devnani, J. Hoffman, and D. Batra, 'Zson: Zero-shot object-goal navigation using multimodal goal embeddings," arXiv preprint arXiv:2206.12403, 2022.
- [15] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "Clip on wheels: Zero-shot object navigation as object localization and exploration," arXiv preprint arXiv:2203.10421, 2022.

- [16] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., "Learning transferable visual models from natural language supervision," in International
- conference on machine learning. PMLR, 2021, pp. 8748–8763. L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, and J. Gao, "Grounded Language-Image Pre-training," Jun. 2022.
- [18] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill et al., "On the opportunities and risks of foundation models," arXiv preprint arXiv:2108.07258, 2021.
- [19] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for
- image captioning," *arXiv preprint arXiv:2111.09734*, 2021. R. Dale, "Gpt-3: What's it good for?" *Natural Language Engineering*, vol. 27, no. 1, pp. 113–118, 2021. [20]
- [21] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba, "Large language models are human-level prompt engineers," arXiv preprint arXiv:2211.01910, 2022.
- [22] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., "Language models are few-shot learners," Advances in neural information processing systems, vol. 33, pp. 1877–1901, 2020.
 S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands
- [23] for robotic navigation and mobile manipulation," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 25, no. 1, pp. 1507-1514, Aug. 2011. [Online]. Available: https: //ojs.aaai.org/index.php/AAAI/article/view/7979
- [24] 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 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 6964–6971.
 [25] Z. Hu, J. Pan, T. Fan, R. Yang, and D. Manocha, "Safe navigation
- with human instructions in complex scenes," IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 753-760, 2019.
- V. S. Dorbala, A. Srinivasan, and A. Bera, "Can a robot trust you?: A [26] drl-based approach to trust-driven human-guided navigation," in 2021 IEEE International Conference on Robotics and Automation (ICRA).
- IEEE, 2021, pp. 3538–3545. T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions," in *2010 5th ACM/IEEE International* Conference on Human-Robot Interaction (HRI). IEEE, 2010, pp. 259-266
- J. Thomason, A. Padmakumar, J. Sinapov, N. Walker, Y. Jiang, H. Yedidsion, J. Hart, P. Stone, and R. J. Mooney, "Improving [28] grounded natural language understanding through human-robot dialog," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 6934–6941.
- [29] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, "Vision-and-dialog navigation," in *Conference on Robot Learning*. PMLR, PMLR. 2020, pp. 394-406.
- X. Gao, Q. Gao, R. Gong, K. Lin, G. Thattai, and G. S. Sukhatme, [30] 'Dialfred: Dialogue-enabled agents for embodied instruction following," IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 10049-10 056, 2022
- [31] 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 Workshop on Language and Robotics at CoRL 2022, 2022
- [32] B. Yamauchi, "A frontier-based approach for autonomous exploration," in Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'. IEEE, 1997, pp. 146-151.
- [33] O. Mees, A. Emek, J. Vertens, and W. Burgard, "Learning object placements for relational instructions by hallucinating scene representations," in 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 94-100.
- [34] O. Mees and W. Burgard, "Composing pick-and-place tasks by grounding language," in *Experimental Robotics: The 17th International Symposium*. Springer, 2021, pp. 491–501. R. Kartmann, D. Liu, and T. Asfour, "Semantic scene manipulation
- [35] based on 3d spatial object relations and language instructions," in 2020 IEEE-RAS 20th International Conference on Humanoid Robots (*Humanoids*). IEEE, 2021, pp. 306–313. [36] A. Boteanu, T. Howard, J. Arkin, and H. Kress-Gazit, "A model for
- verifiable grounding and execution of complex natural language instructions," in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016, pp. 2649–2654.
- T. M. Howard, S. Tellex, and N. Roy, "A natural language planner interface for mobile manipulators," in 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014, pp. 6652-6659.

- [38] A. Boteanu, J. Arkin, S. Patki, T. Howard, and H. Kress-Gazit, "Robotinitiated specification repair through grounded language interaction," arXiv preprint arXiv:1710.01417, 2017.
- [39] H. Kress-Gazit, M. Lahijanian, and V. Raman, "Synthesis for robots: Guarantees and feedback for robot behavior," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 211–236, 2018.
- [40] V. Raman, C. Lignos, C. Finucane, K. C. Lee, M. P. Marcus, and H. Kress-Gazit, "Sorry dave, i'm afraid i can't do that: Explaining unachievable robot tasks using natural language." in *Robotics: science* and systems, vol. 2, no. 1. Citeseer, 2013, pp. 2–1.
- [41] Z. Zhao, W. S. Lee, and D. Hsu, "Differentiable parsing and visual grounding of natural language instructions for object placement," in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 11546–11553.
- [42] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.
- [43] S. Stepputtis, J. Campbell, M. Phielipp, S. Lee, C. Baral, and H. Ben Amor, "Language-conditioned imitation learning for robot manipulation tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13139–13150, 2020.
 [44] C. Lurab, and B. S. Lee, and B. S. Lee, C. Baral, and B. S
- [44] C. Lynch and P. Sermanet, "Language conditioned imitation learning over unstructured data," arXiv preprint arXiv:2005.07648, 2020.
- [45] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International Conference on Machine Learning*. PMLR, 2022, pp. 9118–9147.
- [46] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.
 [47] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng,
- [47] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv* preprint arXiv:2207.05608, 2022.
- [48] D. Shah, B. Osinski, B. Ichter, and S. Levine, "LM-Nav: Robotic Navigation with Large Pre-Trained Models of Language, Vision, and Action," Jul. 2022.
- [49] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual Language Maps for Robot Navigation," Oct. 2022.
- [50] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," arXiv preprint arXiv:2209.11302, 2022.
- [51] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
- [52] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping languageimage pre-training for unified vision-language understanding and generation," in *International Conference on Machine Learning*. PMLR, 2022, pp. 12888–12900.
- [53] 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," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.
- [54] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [55] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen *et al.*, "Simple open-vocabulary object detection," in *European Conference* on Computer Vision. Springer, 2022, pp. 728–755.
 [56] H. Bangalath, M. Maaz, M. U. Khattak, S. H. Khan, and F. Shah-
- [56] 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," *Advances in Neural Information Processing Systems*, vol. 35, pp. 33781–33794, 2022.
- [57] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.
- [58] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.
- [59] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," in ECCV, 2022.
- [60] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang, K.-W. Chang, and J. Gao, "Grounded Language-Image Pre-training," Jun. 2022.