Embodied Symbiotic Assistants that See, Act, Infer and Chat

Yuchen Cao[†]**, Nilay Pande**[†]**, Ayush Jain, Shikhar Sharma, Gabriel Sarch,**
Nikolaos Gkanatsios, Xian Zhou, Katerina Fragkiadaki
Carnegie Mellon University
{yuchenca, nmpande, ayushj2, shikhar2, gsarch, ngkanats, xianz1}@andrew.cmu.edu,
katef@cs.cmu.edu

Abstract

We present Symbiote, an embodied home assistant that maps images from its camera into objects and rooms, builds geometric semantic maps, parses human instructions and conversations into user intents and their arguments, explores in a goal-directed way to find relevant objects (if not present in the map) and executes the inferred actions plans using its navigation and manipulation policies, and/or ask questions to clarify intents and arguments. Our main contribution is a hybrid approach to the semantic parsing of user instructions and their mapping to suitable action routines. We propose a text-to-text neural encoder-decoder language parsing model that maps user instructions to sequences of simplified utterances. The generated utterances are then mapped to parameterized action primitives to execute by a rule-based parser. Our neural parser benefits from large-scale text-to-text unsupervised language pre-training, and our rule-based parser effectively covers the domain of simplified single-step instructions that our neural model generates. Training our neural parser to map language utterances directly to parameterized action programs would not work as the output space would be much outside the text domain that the neural model has been pre-trained on. We present ablations and evaluations of different modules of our agent. We discuss our failure models which are mostly related to a lack of accurate referential object instance grounding, instruction parsing, and perception failures. We outline current and future experiments and research directions in the realms of open-vocabulary spatio-temporal 2D and 3D perception, memory-augmented vision-language parsing networks to handle continual learning without forgetting, and fast and few-shot learning during deployment and interaction with human users. We also discuss our present conversational strategies and how we plan to make them more creative and engaging for the user.

1 Introduction

Our SimBot, called Symbiote, is an embodied home assistant that moves in the environment, parses images from its camera into objects, parts, and rooms, builds geometric semantic maps, parses human instructions and conversations into user intents and their arguments, explores in a goal-directed way to find relevant objects (if not present in the map), execute the inferred actions plans using its navigation and manipulation policies, and/or ask questions to clarify intents and arguments. Our agent consists of four main modules: a **semantic parsing module** whose aim is to map human instructions and directions into programs over parametrized action primitives, a **perception module** responsible for detecting objects in the scene and differentiating between different instances of the same object class using information from the instruction (like object color), **a mapping and planning module** which

[†] Equal Contribution

builds an explicit semantic map of objects that the agent sees during its navigation and **an object search policy module** which is responsible for utilizing the perception and mapping modules to find instruction-relevant objects in the environment.

Prior methods in language semantic parsing typically rely on either rule-based parsers (66; 69; 37; 65) or end-to-end deep learning-based parsers(8; 22; 68). Rule-based parsers typically achieve great performance in a narrow domain of structured language utterances but fail when faced with *natural* language instructions. Neural parsers effectively handle natural language instructions in their training distribution but their performance suffers outside the training distribution. We propose a hybrid approach to language parsing that combines the capabilities of rule-based and learning-based parsers, exploits large-scale unsupervised language pre-training, and achieves reliable and generalizable instruction-to-action mapping.

We equip our agent with spatial object memory and maintain a 2D semantic map of the environment that keeps track of all objects that the agent has detected during task executions in the environment. This enables faster object search, especially in cases when the agent needs to interact with an object that it may have seen earlier while executing a different task. Our search policy is hierarchical and efficient – it iteratively searches for objects within its current panoramic view and semantic object memory. It resorts to an exhaustive search of the environment for objects it has not seen before.

We test our method on the recently proposed Alexa Arena Benchmark (27). We show that our model achieves strong task execution accuracy. We further ablate its design choices, namely the hybrid parsing model and semantic object memory, and show each one contributes to performance.

Contributions In summary, the contributions of our work are as follows:

- We propose a modular instruction following agent architecture that can execute language instructions efficiently and robustly in its environment.
- We propose a hybrid approach to semantic parsing of language instructions that combines the strengths of rule-based parsers and large-scale text-to-text language models.
- We design a mapping and planning component that imparts a spatial object memory to facilitate and accelerate object search.
- We perform various ablations and experiments to study the effect of each of our proposed design components.

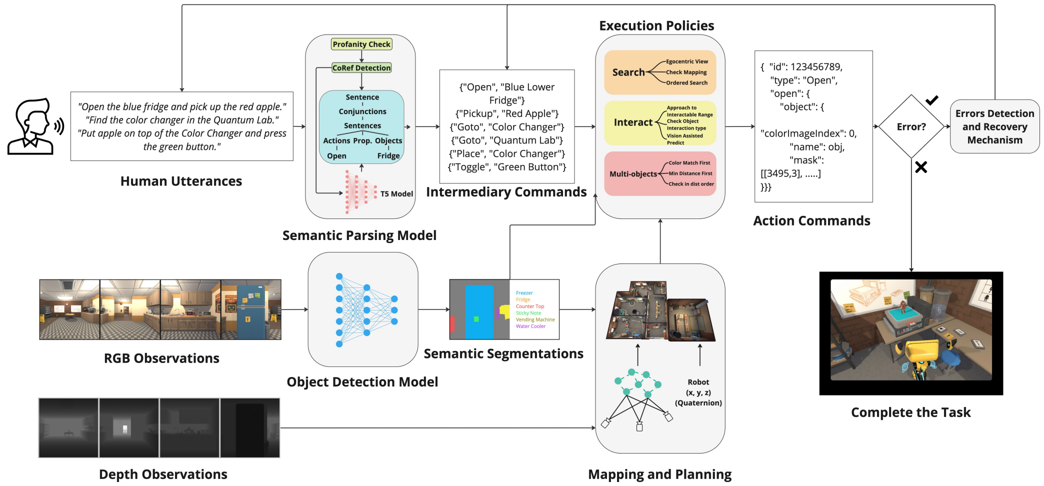
2 Model design and architecture

Symbiote's architecture is shown in Figure-1. Given a verbal instruction from the user, First, we convert verbal user instructions to language utterances using Automatic Speech Recognition from Amazon Alexa. Then, our semantic parser maps the language utterance to a sequence of low-level paramertized actions, which include searching for an object in the environment, interacting with an object in the environment, or interacting with a user in natural language, to ask for help. We describe each component in more detail right below.

2.1 Semantic parsing using rules and neural networks

Our semantic parser maps language instructions to instantiations of various low-level actions and their arguments (for eg. *GoTo, Move Forward, Move Backward, Rotate Right, Rotate Left, Look Down, Look Up, Look Around*). Our parser is a hybrid between a rule-based approach and an end-to-end trained parsing approach. The rule-based method provides a set of predefined rules for interpreting the language input, while the neural parsing method learns from examples to map language utterances to sequences of simple structured utterances.

Rule-Based instruction parsing: We define a set of rules that map a language utterance to a set of low-level actions. The rule-based parser should only retain the relevant instructions and discard any irrelevant or inappropriate content, such as profanity or advice on topics outside the scope of our robot (for example advice on investing). Given a language utterance, we first use a profanity and relevancy checker from AllenNLP (28) to filter out only the relevant instructions. Next, we break down a composite instruction sentence into simpler utterances by breaking the sentence on

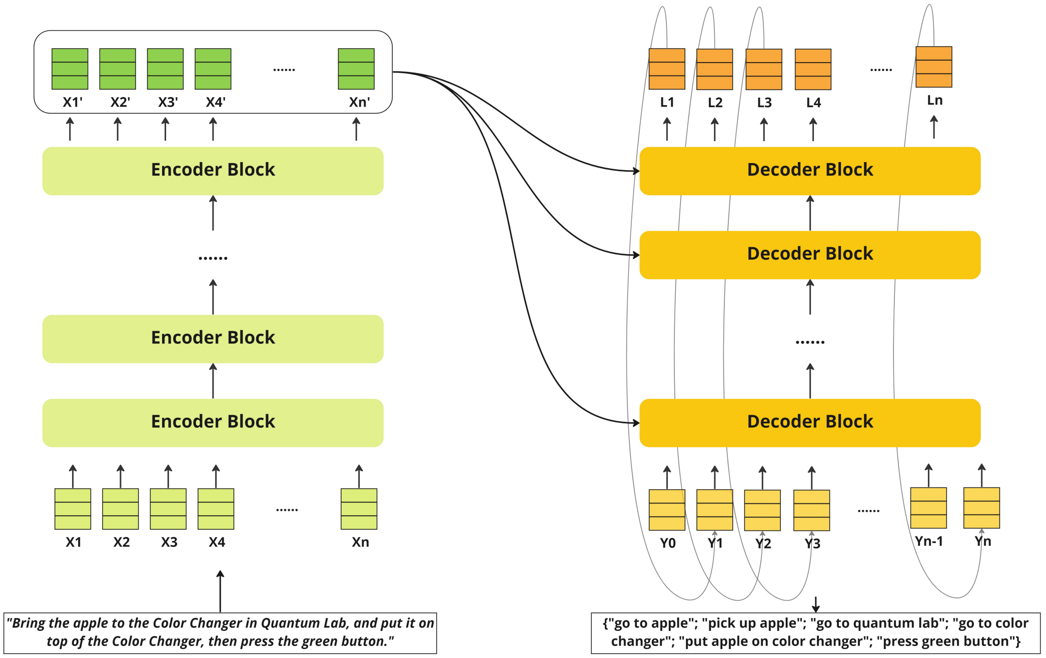


```
{"Open", "Blue Lower
Fridge"}
{"Pickup", "Red Apple"}
{"Goto", "Color Changer"}
{"Goto", "Quantum Lab"}
{"Place", "Color Changer"}
{"Toggle", "Green Button"}
```

```
{  "id": 123456789,
   "type": "Open",
   "open": {
      "object": {

"colorImageIndex": 0,
         "name": obj,
         "mask":
[[3495,3], .....]
}}}
```

Figure 1: **Embodied Symbiotic Assistants that See, Act, Infer and Chat.** Our system predicts low-level navigation and interaction routines given a panoramic image around the agent and a language instruction. We use a hybrid instruction parser, that uses a large-scale pretrained and finetuned text-to-text model (54) to map a complex utterance to a sequence of simplified utterances. Then, a rule based parser maps each one to a parametrized action that the agent can execute. The agent maintains an object memory map and uses search policies to locate the object of interest in the environment. In case of ambiguity or failure during execution, our system seek assistance from the user by asking relevant questions.



"Bring the apple to the Color Changer in Quantum Lab, and put it on top of the Color Changer, then press the green button."

{"go to apple"; "pick up apple"; "go to quantum lab"; "go to color changer"; "put apple on color changer"; "press green button"}

Figure 2: **Our neural semantic parser architecture.** Our neural parser maps complex human instructions to a sequence of structured utterances, that are converted into action commands using our rule-based parser. Our neural parser is an encoder decoder architecture, that first featurizes words of the input utterance with multiple self-attention layers over word tokens, and then generates the output sentence by attending to the encoded input and the already generated output, token by token.

conjunctions like “and", “then" etc. We apply co-reference resolution following AllenNLP (28) to resolve the pronoun co-references in the utterances. Finally, we do token pattern matching which map each simple sentence to a sequence of low-level parametrized actions.

Visual conditioning for instruction interpretation The utterances used as input for parsing are obtained from Amazon Alexa's ASR techniques and can be very noisy. Furthermore, human users oftentimes provide incomplete or ambiguous instructions, e.g., *"Place the cup"* which misses the information about the receptacle to place the object on. To deal with such cases, we utilize the robot's visual scene context. Concretely, we consider all objects detected in the agent's egocentric view and use affordability common sense reasoning to complete the missing information and suggest plausible arguments for our action routines. In the above example, if there is a table in our agent's view, the parser would generate a low-level command to place the cup on the table. If a given instruction does not syntactically match with any of the rules, we resort to our neural semantic parser.

Neural instruction parsing: Our neural semantic parser is an encoder-decoder model based on the T5 architecture (54), we show its architecture in Figure 2. It takes as input a language utterance which can be either a simple or multi-task complex instruction, encodes it via its language encoder, and predicts a sequence of simpler utterances that can be easily parsed by our rule-based parser, as shown in Figure 2. We start from the publically available weights from HugginFace (75) and fine-tune it on the instruction-annotation pairs from Amazon Alexa arena benchmark (27). To enable the model to parse multiple instructions simultaneously, we use augmentations such as randomly selecting two or more instructions and combining them using punctuations such as full stop, comma, or words like 'and' and 'then'. Furthermore, we enhance the model's robustness by occasionally replacing words with their synonyms. Our neural parser benefits from large-scale text-to-text unsupervised language pre-training. Training our neural parser to map language utterances directly to parameterized action programs would not work as the output space would be much outside the text domain that the neural model has been pre-trained on.

2.2 Semantic mapping and planning

Our agent navigates in the home environment and builds semantic geometrically-consistent spatial maps of the environment in 2D (overhead view) (15; 56). The maps keep track of previously seen objects and guide exploration to objects of interest. Specifically, we maintain a spatial visual map of the environment that is updated at each time step from the input RGB-D stream, similar to previous works (57). Within the map, we maintain an object memory as a list of 3D object detection centroids and their predicted semantic category labels $= \{[(X,Y,Z)_i, \ell_i \in \{1 \dots M\}], i = 1 \dots N\}$, where N is the number of objects detected thus far and M is the number of semantic classes ($M = 85$ in our case). We detect objects from semantic object categories in each input RGB image using the Mask-RCNN object detector (35), pre-trained on the MS-COCO datasets (47) and finetuned on images from the training scenes. We obtain 3D object centroids by masking the depth image using detected 2D segmentation masks and orienting the centroid to the coordinate of the map using agent ego-motion. We use non-maximal suppression via Euclidean distance thresholding to remove duplicate objects in our memory.

The Alexa Arena platform (27) we evaluate our system on simplifies navigation by allowing the robot to navigate directly to a viewpoint in a room by specifying the viewpoint or room name, or to an object by specifying the object mask. Alternatively, the robot can perform step-by-step navigation using local primitive actions such as MoveForward, MoveBackward, and Rotate. These actions can take granular inputs as arguments to enable fine-grained control. Additionally, the platform supports a special lookAround action that provides panoramic images to help the robot perceive its surroundings and navigate to objects in its vicinity. While our system does not currently use its full mapping and planning module for indoor navigation due to the simplified navigation setting, our prior work (56) demonstrates its potential for indoor navigation in more realistic settings.

2.3 Object Search Policy

We utilize the semantic map that our agent builds for efficient object search. Concretely, when the user instructs us to manipulate a specific object, for instance "cup", we first check if it is visible in our panoramic view. If it is visible, we directly navigate to it. However, if it is not visible, we check whether any object with label "cup" exists in our semantic map – which is likely to be there if the agent saw it in some previous time step. When the object exists in our semantic map, we directly navigate to it. The users also have the choice to ask us to locate a different instance of the object, if they wish to do so. If the object is not even present in our map, we perform a frontier-based exploration in the current room and finally in the whole environment until we find the thing (or max step exceeds). This design makes our search very reliable and time-efficient. When we come across multiple instances of an object, we distinguish between them by examining certain visual characteristics such as color, if they are specified in the language instructions. In addition, we employ the "highlight" feature to repeatedly ask the user and confirm whether the instance they wish to engage with has been identified.

2.4 Error Detection and Dialogue Strategies

Our model can detect and recover from failed interaction or navigation actions. We employ various strategies to achieve this goal, including automated recovery and human intervention via dialogue. Below, we outline different failure and recovery modes.

- **Object too far for interaction**: One example of a failure-detection-and-recovery strategy is for cases where the target object is too far away from the agent to interact with it. Our agent must be within 2 meters of the object to interact with it successfully. If a user attempts to interact with an object that is out of range, our system detects this and initiates navigation actions to move the agent closer to the object.
- **Proactive Correction Strategy for Out of Vocabulary Object Names in Language Instruction**: Occasionally, the language instruction may contain object names that are not recognized by our object detector. This can occur when a user mentions a new object or when the Automatic Speech Recognition (ASR) system has made a mistake. We then check the objects in the current view and use affordability reasoning to identify objects that could match the user's intended interaction. For instance, if the instruction is to pour, we filter out only pourable objects such as milk cans and glasses. Next, we ask the user for clarification by saying, *"I didn't fully understand. Could you please specify which object you wanted to pour from? The milk can or the glass?"* To perform affordability reasoning, we maintain a catalog of affordable actions for every object in our vocabulary. We further plan to address the out-of-vocabulary error modes by considering an open-vocabulary object detector, such as the one developed in our previous work (40), as we describe in Section 5.
- **Leveraging Sticky Notes for Task Completion in the Alexa Arena Platform**: The Alexa Arena Platform (27) employs "sticky notes" placed throughout the environment to aid the user in completing the task. In case of command failures, such as encountering an obstacle preventing our agent from approaching the target object, we prompt the user to read the sticky notes, if any are visible within our agent's egocentric view. The prompt would be, "Something is blocking my way. Let's check what the sticky notes say." This approach helps even novice users who may be unfamiliar with sticky note usage to benefit from the hints and improve their chances of accomplishing the task.
- **Handling Preconditions for Successful Interactions:** Sometimes, users interact with objects without fulfilling the necessary preconditions for successful interaction. For instance, a user may attempt to pick up an object while the agent is already holding another object. In such cases, we prompt the user with a message such as, "I am already holding something. Please, let's put it down somewhere before picking up a new object." Other instances include trying to place an object inside a closed receptacle without first opening it or attempting to place an object without first picking it up. In such cases, we prompt the user to perform the necessary actions before continuing with the interaction.

3 Datasets

We utilize publicly available data from Alexa Arena Benchmark (27) to train our neural semantic parser and object detection models. The benchmark includes ground-truth action trajectories for over 3.5k game missions, each paired with robot view images and three sets of language instructions. For each set of instructions, there are also two sets of questions and answers collected. Additionally, synthetic language instructions are provided for each action trajectory.

To train our neural parser, we extract the language instruction-template instruction pair and fine-tune a pre-trained T5 (54) checkpoint on this data, mapping human language instructions to synthetic language instructions. At test time, the neural parser maps natural language instructions to structured language utterances, which are then parsed by our rule-based parser into parametrized action commands that can be executed in the environment.

In addition to the language annotations, we also use the labeled object detection dataset from Alexa-Arena to train our Mask-RCNN (35) model for object detection.

For our submission to the eval.ai server, we use the latest Alexa Arena build.

4 Evaluation results and ablations

Our experiments aim to answer the following questions:

- How does our hybrid parser compare to a rule-based alone parser or a neural learning-based alone parser?
- Does our mapping and planning module make object search more efficient?

Metrics We evaluate our model and its variants of our model on the following metrics: a) **Goal Completion,** (higher is better) which measures the model's ability to complete all subtasks of a game; b) **Execution Time,** (lower is better) which refers to the time taken by the agent to complete the game; and c) **Execution Steps,** (lower is better) which refers to the number of steps taken by the agent to complete the game.

Analysis: Our analysis of various model variants on 10 game trials in the online play of Amazon Alexa Arena Benchmark (27) is presented in Table-1. We observe that the performance of our model drops to 80% without the rule-based parser and to 90% without the end-to-end parser. This suggests that our hybrid design of rule-based and end-to-end parser yields the best performance. While removing mapping and planning does not decrease performance, it significantly increases execution time and execution steps. This is because without mapping and planning, the agent would resort to exhaustive object search, which would eventually find the object but would result in longer execution times.

Method	Goal Completion	Execution Time	Execution Steps
Symbiote	**100%**	**268**	**14.6**
Symbiote w/o rule-based parser	80%	470	22.2
Symbiote w/o end-to-end parser	90%	272	14.8
Symbiote w/o mapping and planning	**100%**	284	15.8

Table 1: **Ablations of Symbiote on 10 trials**

We further evaluate our neural parser alone on the validation set of Alexa Arena Benchmark (27) for the exact match metric i.e. if our predicted program matches token by token with the ground truth program. Our neural parser achieves 87% accuracy in the validation set.

4.1 Failure modes - Limitations

We visualize some of our failure modes in Figure 3. Our current system has the following main error modes:

- **Object instance referential resolution.** One of the key challenges our model faces is resolving references to multiple instances of the same object category in a scene. Currently, we rely solely on color hints from the instruction or the use of the *highlight* function to address this issue. To handle more complex referential grounding instructions, we plan to explore the incorporation of a referential language vision-language grounding model, such as our previous work (40), in combination with our semantic parser. This is an important area for future research that we intend to pursue.
- **Perception failures.** Our perception relies on a per-frame R-CNN visual detector, which has the following limitations:

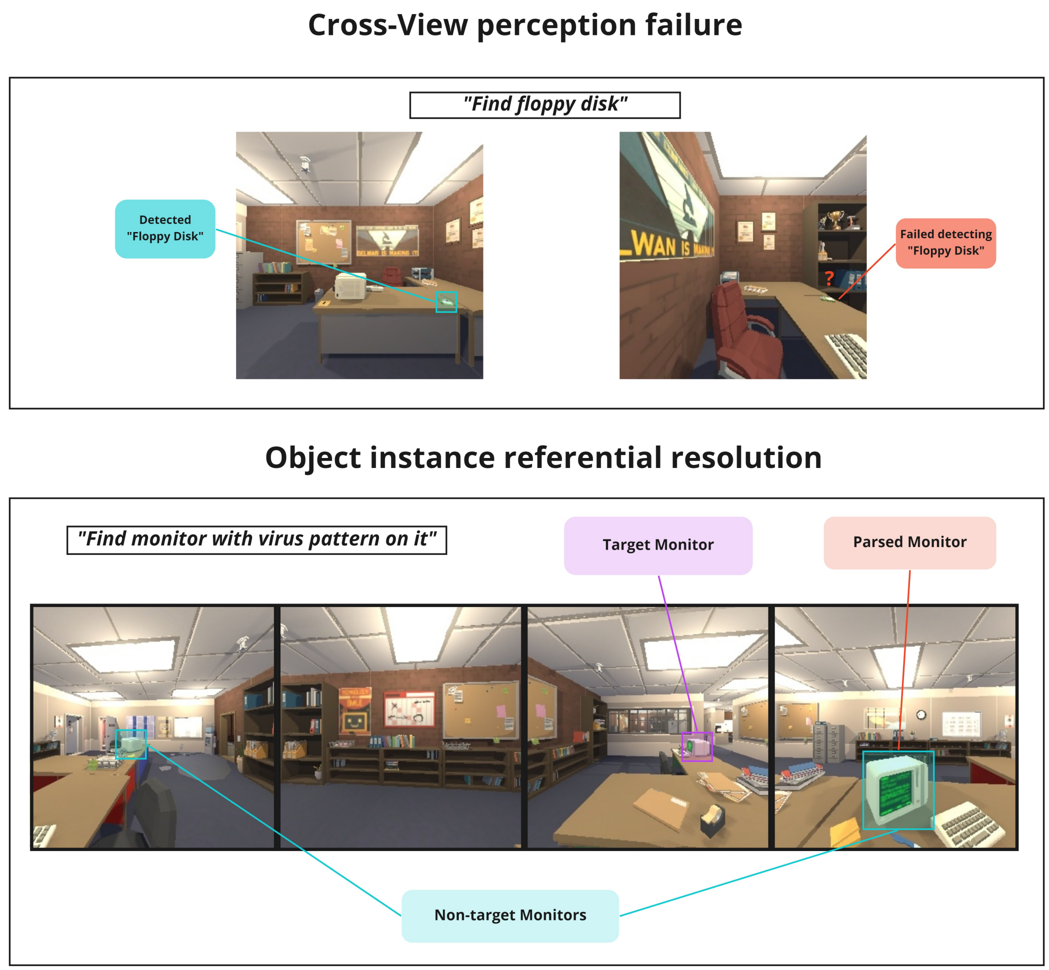

Cross-View perception failure
"Find floppy disk"
Detected "Floppy Disk"
Failed detecting "Floppy Disk"
Object instance referential resolution
"Find monitor with virus pattern on it"
Target Monitor
Parsed Monitor
Non-target Monitors

Figure 3: **Failure modes of Symbiote.** *Top:* The output of our per-frame object detector varies across different viewpoints, which results in failure to detect an object in certain views. *Bottom:* We cannot presently distinguish among object instances of the same category (in this case, a computer) that are not referred to based on their color. Using an open vocabulary referential expression grounding, such as the one presented in our recent work (40), would address this limitation.

  - Detections jitter across frames (the detector fails to detect or erroneously detects objects in some frames) due to lack of spatio-temporal information integration.
  - The detector only provides semantics at a specific level of granularity, and cannot provide finer perceptual details required in some tasks, e.g., chair, pot, drawer or door handles, centers of coffee tables, keys of keyboards, etc.. Each task has a different set of perceptual entities, that our perception system should be able to localize in the scene, as well as flexibly expand its vocabulary. Right now, a fixed set of object category labels are being used.

- **Semantic parsing failures.** Our hybrid semantic parser sometimes fails to parse the input utterances, due to utterance complexity. More training data as well as more flexible API from our parser to our executor are important avenues of future experimentation.
- **Rule-based conversational interaction.** Our conducted conversations are not very interesting or creative, they are dictated by a rule-based logic. Interfacing our embodied perceptual agent with language models for more creative generation or paraphrasing of structured grounded questions or responses, would be useful avenues of future work.
- **Coarse "common sense" core knowledge in language parsing and perception**. Our system has some basic common sense regarding what objects can be paired with what

actions, e.g., a cup can be placed on a table. To improve our embodied agent further, we need to go beyond symbolic common sense of object categories, and incorporate common sense regarding fine-grained object affordances. This would also help our semantic parser to fill in missing information in interpreting ambiguous user instructions.

5 Future Work and Planned Experiments

Our future work is directly related to the failure modes we outlined in Section 4.1. Our vision for our Symbiote agent is the incorporation of referential object grounding and open vocabulary spatio-temporal perception, as well as the development of continual learning and few-shot learning capabilities, detailed in the following pillars:

1. Our SimBot should continually improve its ability to see and parse the visual environment during deployment time, guided by self-supervised learning and active perception, as well as interactions with and guidance from human users.
2. Our agent should continually expand the language domain it can effectively parse, understand and execute by building concept, action, and model abstractions, that can be indexed later with natural language, as well as using supervised instruction and scene-to-program pairs when needed. In this way, the human user can directly refer to previously taught procedures, while abstracting away from the details, e.g., “prepare the dinner table as I taught you yesterday", or “these are my favorite tiny marble collection". Macro-actions will be built over other macro-actions, and in this way, our agent can plan and reason in coarser granularities, which is critical for effective search and reasoning over longer courses of action. Continual language understanding requires common sense knowledge acquisition regarding the environment, objects, and affordances. Our agent should use fine-grained models of how the world works to fill in missing information during semantic parsing of human instructions and conversations, and check user’s commands that may cause dangerous or undesired outcomes. For example, when SimBot is asked to “empty the soup bowl in the pot", the SimBot should be able to predict using its world models possible problems and raise relevant questions, e.g., if appropriate “Are you sure? It may overflow".
3. Our SimBot should default to the generic “I do not know what to do" as rarely as possible, and instead display to the human user an informative set of alternatives to choose from, alongside being open to a completely new course of actions. E.g., “would you like espresso, latte, americano, or something else?". This helps create a transparent and engaging interactive experience.

To be able to develop the above capabilities, we have been exploring two main research thrusts:

- Open-vocabulary spatio-temporal 2D and 3D perception and open vocabulary language grounding.
- Memory-augmented neural network architectures for fine-grained commons sense learning grounded language understanding.

We detail these research thrusts right below.

Open vocabulary 2D-3D and spatio-temporal perception The effectiveness of home assistants critically depends on their ability to accurately parse the visual scene. There has been tremendous progress on single image visual understanding, fueled by large-scale 2D image-caption datasets and object annotation datasets (48; 59). An embodied agent needs to understand a scene from a sequence of frames some of which may have better visibility and object detectability than others. We are exploring methods and architectures that build upon state-of-the-art open-vocabulary single frame detectors (83) and state-of-the-art label propagation methods (19) to generate stable spatio-temporal trackelts for any detectable object in the video, at any frame. We are experimenting with the estimation of camera motion and whether it helps with such temporal propagation of detection responses. We are further building upon our previous works (34; 33; 67) to propose neural architectures that can seamlessly process single-frame, multi-frame posed or unposed sets of images, to fuse features across them and predict object detections that are more accurate than using each frame in isolation. We believe these efforts will result in open-vocabulary detections and referential grounding, stable in space and time, that will be very useful for our Symbiote agent.

Memory retrieval and analogy as knowledge representation The dominant paradigm in today's deep visual and language learning is to train high-capacity networks to map visual and language input to output language or visual target labels (13; 12). There has been a trend towards pushing the capacity of these models to the limit (79), and training them in very large-scale datasets, with the recent Microsoft multi-modal learner (70) having 1.9 billion parameters. Scaling up tremendously improves numbers on established benchmarks. These models have been criticized for acquiring a superficial understanding of language, predicting non-consistent statements, and showing brittle performance, especially out of the training distribution (10). We conjecture that the brittleness of modern deep networks is due to their lack of explicit representations of world common sense knowledge regarding stereotypical objects, object arrangements, scenes, rooms, actions, and events. Indeed, all domain knowledge is implicitly encoded in the model weights. As a result, networks are not conscious of missing entities in the sensory stream, for example, they do not get surprised about a chair with two legs, a car without doors, or a bottle without an opening, because they only build implicit priors about how the world looks and works. **In order to infer variations or anomalies, the stereotypical situation needs to be explicitly modeled.** We believe that the lack of explicit representation of the visuomotor structure of the world in a way that can be easily retrieved and used to perceive, act, and predict the future is a central missing piece in today's deep learning and embodied AI research.

To address the above limitation, and support the learning capabilities of the next generation of intelligent vision-language embodied learning systems, we are exploring an analogical framework for knowledge representation, perception, and language grounding that **encodes domain knowledge explicitly, in a collection of structured sensory experiences at different levels of spatial and temporal abstraction,** in addition to implicitly, as network parameters. In our recent work TIDEE (56) we presented an agent equipped with a memory of object arrangements that it uses to predict out-of-place objects and plausible object re-locations, for tidying up novel scenes, relying on its own perception and action routines. In our recent work Analogy-forming transformers (29), we build network architectures equipped with external memory that learn mainly self-supervised to predict alignment between two 3D scenes. We will build and extensively innovate over these works to develop an analogical framework for grounded language understanding that can support continual and few-shot learning. Memories are perceptual experiences in space and time, labeled with related symbols of objects, object parts, object trajectories, state changes, descriptions, and captions.Each memory experience is encoded as a spatio-temporal graph of perceptual entities alongside their symbols (roles, attributes, objects, and action labels). Each entity in each level is represented by a learnable latent feature embedding, produced by memory encoding. During retrieval, the network retrieves complete entity graphs from incomplete partial sensory observations, and uses them to modulate perceptual inference in order to localize objects and actions, predict possible future and past completions, evaluate counterfactuals, ground referential, answer questions, and follow instructions. Each memory or set of memories operates as an "expert" model abstraction that modulates model inference in order to explain and complete a particular family of sensory inputs. In this way, the world state permits a multitude of representations, depending on retrieved memories and their structure. The system learns "fast" from a single example (42; 53) by simply storing it, and thus can learn continually without requiring i.i.d. examples shown all at once. Knowledge will be updated over time by updating the memory experience graphs. Each memory is further annotated with a relevant instruction or description. This means that language features will be used for retrieving related situations and better resolving to parse users' instructions. RETRO (11) already has shown that explicit memory is beneficial for text-to-text, language-related tasks. We wish to explore this capability for grounded language understanding and instruction following visually grounded agents.

6 Related work

Semantic Parsing of natural language instructions Semantic parsers map language utterances to formal representations of their meaning (81; 82; 46). Existing learning algorithms have primarily focused on building *actionable* meaning representations, e.g., for querying a knowledge base (KB) (9), instructing a robot to navigate in its environment (49), or programming a new functionality on a personal agent or device (4). Semantic parsers have mostly been highly domain-specific, and heavily utilize domain restrictions to resolve ambiguities (71). A variety of models for semantic parsing have been proposed, such as query-graph construction that learns to anchor to the right entity in a KB and guide parsing by proposing constraints and predicates (78), sequence-to-sequence models with attention (5; 41), key-variable memory networks that learn to save and re-use intermediate results (45), tree-structured models that condition on syntactic structure (64) or jointly predict it along with semantic parsing (63; 44), and recurrent neural network features for labeling semantic roles of verbs (84).

The main bottleneck in scaling up semantic parsing is annotating ground-truth logical forms that represent the meaning of utterances (39). Due to their end-to-end nature, parsing models must be relearned for each new target application (71). Many works have developed methods to reduce such annotation efforts. For example, some works use reinforcement learning guided by the result of the execution of the predicted logical form (45; 3). However, recent fully supervised methods (38; 41) greatly outperform their unsupervised equivalents. Other works use binary (correct/incorrect) human feedback signals (20), grammars for sampling canonical utterances and their paired logical forms, and paraphrasing utterances through crowdsourcing (71), active learning for selectively annotating difficult or incorrect examples (39), or NL question-answering for extracting the basic semantic roles by non-expert workers (36) in place of logical form annotations that require expertise. The meaning of an NL utterance can often be understood as a small program to execute in a particular context (3). In that respect, it is similar to program induction the problem of specifying a program using supervision from input/output pairs of its desired behavior. Models for program induction and semantic parsing are divided between neural models that learn to output the program structure directly (51), and approaches that use neural features to better guide a domain-specific search strategy (6).

The seminal T5 (54) work cast many tasks, including semantic parsing, as a text-to-text mapping. Since then, many neural approaches, including neural semantic parsers, follow a pre-train-then-finetune strategy, in which the model is first pre-trained on self-supervised tasks like language modeling or masked autoencoding, and then fine-tuned for the downstream task in a supervised fashion. End-to-end models tend to be less brittle than rule-based parsing modules and generalize better to variations in natural language. However, rule-based parsers still work extremely well in narrow domains, especially if engineered well. In this work, we propose a hybrid approach that combines rule-based and end-to-end parsing models to achieve the best of both worlds.

Embodied AI. The development of learning-based embodied AI agents has made significant progress across a wide variety of tasks, including: scene rearrangement (26; 72; 7), object-goal navigation (1; 77; 76; 16; 31; 14), point-goal navigation (1; 58; 74; 55; 31), scene exploration (18; 15), embodied question answering (30; 21), instructional navigation (2; 61), object manipulation (23; 80), home task completion with explicit instructions (61; 50; 62), active visual learning (17; 24; 32; 73), and collaborative task completion with agent-human conversations (52). While these works have driven much progress in embodied AI, ours is the first agent to tackle the task of tidying up rooms, which requires commonsense reasoning about whether or not an object is out of place, and inferring where it belongs in the context of the room. Progress in embodied AI has been accelerated tremendously through the availability of high visual fidelity simulators, such as, Habitat (58), GibsonWorld (60), ThreeDWorld (25), AI2THOR (43) and recently released Alexa Arena Benchmark (27). Our work builds upon Alexa Arena by relying on the (approximate) dynamic manipulation the simulator enables for indoor scenes.

References

[1] 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.

[2] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018.

[3] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2016.

[4] A. Azaria, J. Krishnamurthy, and T. Mitchell. Instructable intelligent personal agent, 2016.

[5] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[6] M. Balog, A. L. Gaunt, M. Brockschmidt, S. Nowozin, and D. Tarlow. Deepcoder: Learning to write programs. *CoRR*, abs/1611.01989, 2016.

[7] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi, M. Savva, and H. Su. Rearrangement: A challenge for embodied ai. *ArXiv*, abs/2011.01975, 2020.

[8] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544, 2013.

[9] J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, pages 1533–1544. ACL, 2013.

[10] 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.

[11] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Millican, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR, 2022.

[12] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020.

[13] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. *CoRR*, abs/2005.12872, 2020.

[14] M. Chang, A. Gupta, and S. Gupta. Semantic visual navigation by watching youtube videos. *Advances in Neural Information Processing Systems*, 33:4283–4294, 2020.

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

[16] 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*, 33, 2020.

[17] D. S. Chaplot, H. Jiang, S. Gupta, and A. Gupta. Semantic curiosity for active visual learning. In *European Conference on Computer Vision*, pages 309–326. Springer, 2020.

[18] T. Chen, S. Gupta, and A. Gupta. Learning exploration policies for navigation. In *International Conference on Learning Representations*, 2019.

[19] H. K. Cheng, Y.-W. Tai, and C.-K. Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *NeurIPS*, 2021.

[20] J. Clarke, D. Goldwasser, M.-W. Chang, and D. Roth. Driving semantic parsing from the world's response. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, CoNLL '10, pages 18–27, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[21] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2018.

[22] L. Dong and M. Lapata. Language to logical form with neural attention. *arXiv preprint arXiv:1601.01280*, 2016.

[23] L. Fan, Y. Zhu, J. Zhu, Z. Liu, O. Zeng, A. Gupta, J. Creus-Costa, S. Savarese, and L. Fei-Fei. Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In *Conference on Robot Learning*, pages 767–782. PMLR, 2018.

[24] Z. Fang, A. Jain, G. Sarch, A. W. Harley, and K. Fragkiadaki. Move to see better: Self-improving embodied object detection. *arXiv preprint arXiv:2012.00057*, 2020.

[25] C. Gan, J. Schwartz, S. Alter, M. Schrimpf, J. Traer, J. De Freitas, J. Kubilius, A. Bhandwaldar, N. Haber, M. Sano, et al. Threedworld: A platform for interactive multi-modal physical simulation. *arXiv preprint arXiv:2007.04954*, 2020.

[26] C. Gan, S. Zhou, J. Schwartz, S. Alter, A. Bhandwaldar, D. Gutfreund, D. L. Yamins, J. J. DiCarlo, J. McDermott, A. Torralba, and J. B. Tenenbaum. The threedworld transport challenge: A visually guided task-and-motion planning benchmark towards physically realistic embodied ai. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 8847–8854, 2022.

[27] Q. Gao, G. Thattai, X. Gao, S. Shakiah, S. Pansare, V. Sharma, G. Sukhatme, H. Shi, B. Yang, D. Zheng, et al. Alexa arena: A user-centric interactive platform for embodied ai. *arXiv preprint*

arXiv:2303.01586, 2023.

[28] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. S. Zettlemoyer. Allennlp: A deep semantic natural language processing platform. 2017.

[29] N. Gkanatsios, M. Singh, Z. Fang, S. Tulsiani, and K. Fragkiadaki. Analogy-forming transformers for few-shot 3d parsing. In *The Eleventh International Conference on Learning Representations*, 2023.

[30] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4089–4098, 2018.

[31] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[32] N. Haber, D. Mrowca, L. Fei-Fei, and D. L. Yamins. Learning to play with intrinsically-motivated self-aware agents. *32nd Conference on Neural Information Processing Systems*, 2018.

[33] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki. Simple-bev: What really matters for multi-sensor bev perception? *arXiv preprint arXiv:2206.07959*, 2022.

[34] A. W. Harley, F. Li, S. K. Lakshmikanth, X. Zhou, H.-Y. F. Tung, and K. Fragkiadaki. Learning from unlabelled videos using contrastive predictive neural 3d mapping. In *ICLR*, 2019.

[35] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[36] L. He, M. Lewis, and L. Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 643–653, 2015.

[37] G. G. Hendrix, E. D. Sacerdoti, D. Sagalowicz, and J. Slocum. Developing a natural language interface to complex data. *ACM Transactions on Database Systems (TODS)*, 3(2):105–147, 1978.

[38] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[39] S. Iyer, I. Konstas, A. Cheung, J. Krishnamurthy, and L. Zettlemoyer. Learning a neural semantic parser from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 963–973, 2017.

[40] A. Jain, N. Gkanatsios, I. Mediratta, and K. Fragkiadaki. Bottom up top down detection transformers for language grounding in images and point clouds. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVI*, pages 417–433. Springer, 2022.

[41] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, F. Li, C. L. Zitnick, and R. B. Girshick. Inferring and executing programs for visual reasoning. *CoRR*, abs/1705.03633, 2017.

[42] D. Kahneman. *Thinking, fast and slow*. macmillan, 2011.

[43] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

[44] J. Krishnamurthy and T. M. Mitchell. Joint syntactic and semantic parsing with combinatory categorial grammar. In *ACL*, 2014.

[45] C. Liang, J. Berant, Q. Le, K. D. Forbus, and N. Lao. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. *CoRR*, abs/1611.00020, 2016.

[46] P. Liang. Learning executable semantic parsers for natural language understanding. *Commun. ACM*, 59(9):68–76, Aug. 2016.

[47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[48] J. Liu, L. Wang, and M.-H. Yang. Referring expression generation and comprehension via attributes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4856–4864, 2017.

[49] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. *Learning to Parse Natural Language Commands to a Robot Control System*, pages 403–415. Springer International Publishing, Heidelberg, 2013.

[50] S. Y. Min, D. S. Chaplot, P. Ravikumar, Y. Bisk, and R. Salakhutdinov. Film: Following instructions in language with modular methods, 2021.

[51] A. Neelakantan, Q. V. Le, M. Abadi, A. McCallum, and D. Amodei. Learning a natural language interface with neural programmer. *CoRR*, abs/1611.08945, 2016.

[52] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur. Teach: Task-driven embodied agents that chat, 2021.

[53] A. Pritzel, B. Uria, S. Srinivasan, A. P. Badia, O. Vinyals, D. Hassabis, D. Wierstra, and C. Blundell. Neural episodic control. *CoRR*, abs/1703.01988, 2017.

[54] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[55] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman. Occupancy anticipation for efficient exploration and navigation. In *European Conference on Computer Vision*, pages 400–418. Springer, 2020.

[56] G. Sarch, Z. Fang, A. W. Harley, P. Schydlo, M. J. Tarr, S. Gupta, and K. Fragkiadaki. Tidee: Tidying up novel rooms using visuo-semantic commonsense priors. In *European Conference on Computer Vision*, 2022.

[57] G. Sarch, Z. Fang, A. W. Harley, P. Schydlo, M. J. Tarr, S. Gupta, and K. Fragkiadaki. Tidee: Tidying up novel rooms using visuo-semantic commonsense priors. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 480–496. Springer, 2022.

[58] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019.

[59] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022.

[60] B. Shen, F. Xia, C. Li, R. Martín-Martín, L. Fan, G. Wang, C. Pérez-D'Arpino, S. Buch, S. Srivastava, L. P. Tchapmi, M. E. Tchapmi, K. Vainio, J. Wong, L. Fei-Fei, and S. Savarese. igibson 1.0: a simulation environment for interactive tasks in large realistic scenes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, page accepted. IEEE, 2021.

[61] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020.

[62] A. Suglia, Q. Gao, J. Thomason, G. Thattai, and G. S. Sukhatme. Embodied bert: A transformer model for embodied, language-guided visual task completion. In *EMNLP 2021 Workshop on Novel Ideas in Learning-to-Learn through Interaction*, 2021.

[63] S. Swayamdipta, M. Ballesteros, C. Dyer, and N. A. Smith. Greedy, joint syntactic-semantic parsing with stack lstms. *CoRR*, abs/1606.08954, 2016.

[64] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, abs/1503.00075, 2015.

[65] M. Templeton and J. F. Burger. Problems in natural-language interface to dbms with examples from eufid. In *First Conference on Applied Natural Language Processing*, pages 3–16, 1983.

[66] F. B. Thompson, P. C. Lockemann, B. Dostert, and R. Deverill. Rel: A rapidly extensible language system. In *Proceedings of the 1969 24th national conference*, pages 399–417, 1969.

[67] H.-Y. F. Tung, R. Cheng, and K. Fragkiadaki. Learning spatial common sense with geometry-aware recurrent networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2595–2603, 2019.

[68] O. Vinyals, M. Fortunato, and N. Jaitly. Pointer networks. *Advances in neural information processing systems*, 28, 2015.

[69] D. L. Waltz. An english language question answering system for a large relational database. *Communications of the ACM*, 21(7):526–539, 1978.

[70] W. Wang, H. Bao, L. Dong, J. Bjorck, Z. Peng, Q. Liu, K. Aggarwal, O. K. Mohammed, S. Singhal, S. Som, and F. Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks, 2022.

[71] Y. Wang, J. Berant, and P. Liang. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages

1332–1342, 2015.

[72] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi. Visual room rearrangement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[73] L. Weihs, A. Kembhavi, K. Ehsani, S. M. Pratt, W. Han, A. Herrasti, E. Kolve, D. Schwenk, R. Mottaghi, and A. Farhadi. Learning generalizable visual representations via interactive gameplay. *International Conference on Learning Representations*, 2021.

[74] E. Wijmans, A. Kadian, A. S. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *ICLR*, 2020.

[75] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[76] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6750–6759, 2019.

[77] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi. Visual semantic navigation using scene priors. In *Proceedings of (ICLR) International Conference on Learning Representations*, May 2019.

[78] S. W.-t. Yih, M.-W. Chang, X. He, and J. Gao. Semantic parsing via staged query graph generation: Question answering with knowledge base. ACL – Association for Computational Linguistics, July 2015.

[79] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu. Coca: Contrastive captioners are image-text foundation models, 2022.

[80] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pages 1094–1100. PMLR, 2020.

[81] J. M. Zelle and R. J. Mooney. Learning to parse database queries using inductive logic programming. In *AAAI/IAAI*, pages 1050–1055, Portland, OR, August 1996. AAAI Press/MIT Press.

[82] L. S. Zettlemoyer and M. Collins. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *CoRR*, abs/1207.1420, 2012.

[83] H. Zhang, P. Zhang, X. Hu, Y.-C. Chen, L. H. Li, X. Dai, L. Wang, L. Yuan, J.-N. Hwang, and J. Gao. Glipv2: Unifying localization and vision-language understanding. In *Advances in Neural Information Processing Systems*, 2022.

[84] J. Zhou and W. Xu. End-to-end learning of semantic role labeling using recurrent neural networks. In *ACL*, 2015.