Following Natural Language Instructions for Household Tasks with Landmark Guided Search and Reinforced Pose Adjustment

Michael Murray and Maya Cakmak

***Abstract*— We study the challenging problem of following natural language instructions on a mobile manipulator robot. This task is challenging because it requires the robot to integrate the semantics of the unconstrained natural language instructions with the robot's egocentric visual observations of the environment which are typically incomplete and noisy. To address these challenges, we propose a method that is able to use visible landmarks to more efficiently explore the environment in search of the objects described by the natural language instructions. Additionally, we propose using a pose adjustment policy during manipulation planning to help the robot recover from noisy visual observations. We show that this policy can be trained through experience with reinforcement learning as well as with human-in-the-loop feedback. We evaluate our approach on the popular ALFRED instruction following benchmark and show that these methods achieve state-of-the-art performance (35.41%) with a substantial (8.92% absolute) gap from prior work.**

I. INTRODUCTION

For a robot deployed in human-centric environments, natural language provides an intuitive interface enabling non-expert humans to communicate with the robot and instruct it to complete useful tasks. However, executing unconstrained natural language instructions is a challenging problem that requires the robot to understand the semantics of the instructions, ground the semantics to real-world objects through egocentric visual observations, and satisfy the described task by navigating through the environment and manipulating the target objects. Despite recent progress, the problem remains challenging in part because the robot's perception of the environment is often incomplete and noisy. Adaptive decision making is crucial because the robot's perception of the environment is imperfect. Efficient exploration is also necessary because often the target objects that are required to satisfy the language instructions are not immediately observable. For example, the target objects may be in an unseen part of the environment, they may be inside closed receptacles such as refrigerators and cupboards, or they may simply be too small to see from a distance.

In this paper, we propose an approach that more efficiently explores the environment by predicting visible landmarks that are likely to lead to the target objects. For humans, landmarks play an important role in guiding navigational behavior [1]. The use of landmarks as beacons for goal localization is a strategy that develops early in childhood, and may be used in preference to other navigational strategies due to its computational simplicity and high reliability [2]. Our approach aims to utilize landmarks for the navigation phase of the instruction following problem. Towards this goal, we develop a predictive model of landmarks that are likely to lead to a given target object and a procedure for exploring the environment called Landmark Guided Search (LGS). Additionally, we propose using a pose adjustment policy during manipulation planning to help the robot recover from imperfect visual perception. This policy can make minor local pose adjustments that might be required to successfully interact with the target objects. For example, to open a refrigerator, the agent must assume a pose that is close enough to grasp the refrigerator door handle while far enough away that the refrigerator door is not blocked. We call this Reinforced Pose Adjustment (RPA) and we show that this behavior can be learned both through unsupervised experience and through human-in-the-loop feedback.

We evaluate our approach on a popular robot instruction following benchmark, ALFRED [3], and show that these methods achieve state-of-the-art performance (35.41%) with

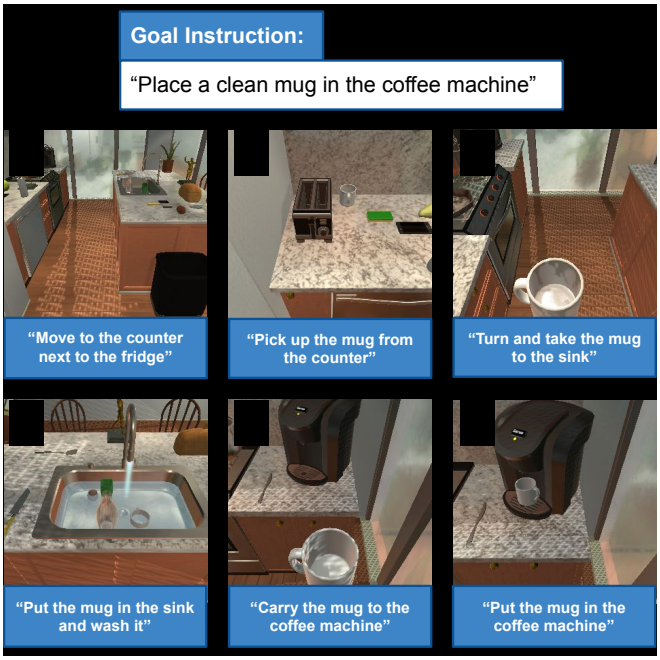

Fig. 1. An example instruction following task from the ALFRED challenge. Efficient exploration is crucial because the objects referenced by the instructions are not immediately visible to the robot. The robot has an imperfect representation of the environment constructed from egocentric RGB images, highlighting the importance planning and manipulation behaviors that can adapt when the robot's perception is inaccurate.

[1]The authors are with the Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA 98195, USA. {mmurr, mcakmak}@cs.washington.edu

a large margin (8.92%) from the previous SOTA [4].

II. RELATED WORK

Natural language is an active area of robotics research [5] that has been studied in contexts such as instruction following [6]–[9], knowledge transfer [10]–[12], and dialog [13]–[15].

Instruction following for robots has been explored in domains ranging from coaching RoboCup soccer robots [16], to executing recipes on cooking robots [17], to routing robotic forklifts [8] and quadcopter drones [9]. Data-driven approaches have shown promising results by learning to directly map language and observations to actions [9], [18]–[20]. Several benchmarks using simulated physical environments have been developed to facilitate progress on learned instruction following for domains such as navigation [21]–[25] and embodied question answering [26]. ALFRED [3] is a benchmark situated in the AI2-THOR simulator [27] that involves following natural language instructions with a mobile manipulator to complete complex tasks in realistic indoor household environments. Some early approaches involve training large end-to-end models that directly translate natural language instructions to low-level actions [3], [28]. Later approaches have decoupled the various aspects of the problem into a hierarchy of individually learned or programmed modules that are easier to interpret for humans and generalize better in unseen environments [4], [29]–[34]. Researchers have developed increasingly sophisticated modules, most recently the authors of [33] and [4] reconstruct a 3D map of the environment with semantic labels encoded in the map. This representation is useful for planning capabilities because it provides a direct mapping from the semantics of the natural language to the spatial properties of the environment. However, in practice, the semantic map is often noisy (due to inaccurate perception and semantic inference) and incomplete (due to insufficient exploration of the environment).

To tackle the exploration problem, [33] proposes doing a simple random search. The authors of [4] improve on this with a search policy that predicts the spatial location of target objects based on the observed semantic map. In our work, we predict landmarks that are likely to lead to the target object and utilize those landmarks to more efficiently explore the environment. Landmarks have been successfully used in related problems such as mechanical search [35]. Because our work is in the context of instruction following, our approach crucially must consider both the language instructions (which may contain references to relevant landmarks) and prior experience (which informs relevant landmarks in the absence of language cues).

To mitigate the inaccuracy problem, we propose using reinforcement learning and human-in-the-loop feedback. Reinforcement learning has been successfully used for other instruction following tasks such as navigation [36]–[39]. However, it is difficult to apply reinforcement learning to the wide variety of long-horizon mobile manipulation tasks evaluated by the ALFRED challenge. In our work, we apply reinforcement learning to a small sub-task which has a short horizon, making reinforcement learning more tractable while still being useful to all of the tasks in ALFRED. Specifically, we teach our mobile manipulator how to adjust its pose to successfully interact with a target object after navigating to that object in the semantic map. By learning how to make small pose adjustments locally, our mobile manipulator can adapt and recover from inaccuracies in the 3D semantic map. Additionally, we show that this approach provides a framework for human-in-the-loop feedback which has been used extensively to train autonomous systems [40]–[43] but to our knowledge has not yet been utilized for instruction following on a mobile manipulator. By correcting the mobile manipulator with small pose adjustments, human experts can give feedback that improves overall performance on the tasks.

III. PROBLEM STATEMENT

A. Task Background

Our work is done in the context of the ALFRED challenge [3], where a mobile manipulator robot is required to follow natural language instructions to complete long-horizon household tasks in realistic indoor home environments (situated in the AI2-THOR simulator [27]). The challenge spans seven types of tasks in 120 indoor scenes. The tasks range from simple pick-and-place tasks to more complex tasks that require heating, cooling, cleaning, or slicing target objects.

B. Task Description

At each timestep t, the robotic agent receives a new observation image o_t and must choose an action a_t. The agent chooses from 13 possible actions including 5 navigation actions: `RotateRight`, `RotateLeft`, `MoveAhead`, `LookUp`, `LookDown` and 7 object manipulation actions: `PickupObject`, `PutObject`, `OpenObject`, `CloseObject`, `ToggleObjectOn`, `ToggleObjectOff`, `SliceObject`. Navigation actions are parameter-free while object manipulation actions are parameterized by a pixelwise mask to denote the target object in the robot's first person view. The agent is given unconstrained natural language instructions L and to succeed it must generate a sequence of actions that satisfy the goal conditions described in L followed by a special `Stop` action to end the episode.

IV. METHOD

Our approach consists of five modules: (1) *language processing*, (2) *perception*, (3) *navigation*, (4) *search*, and (5) *manipulation*. We illustrate the high-level end-to-end architecture in Figure 2.

The language processing module predicts the next sub-goal g_k given the natural language instructions L and the sequence of previous sub-goals $\langle g_i, \rangle_{i<k}$. A sub-goal g is a tuple $(type, target)$ where $type$ is the sub-goal type (e.g. *GotoLocation*, *PickupObject*) and $target$ is the target object of the sub-goal (e.g. *Sink*, *Apple*).

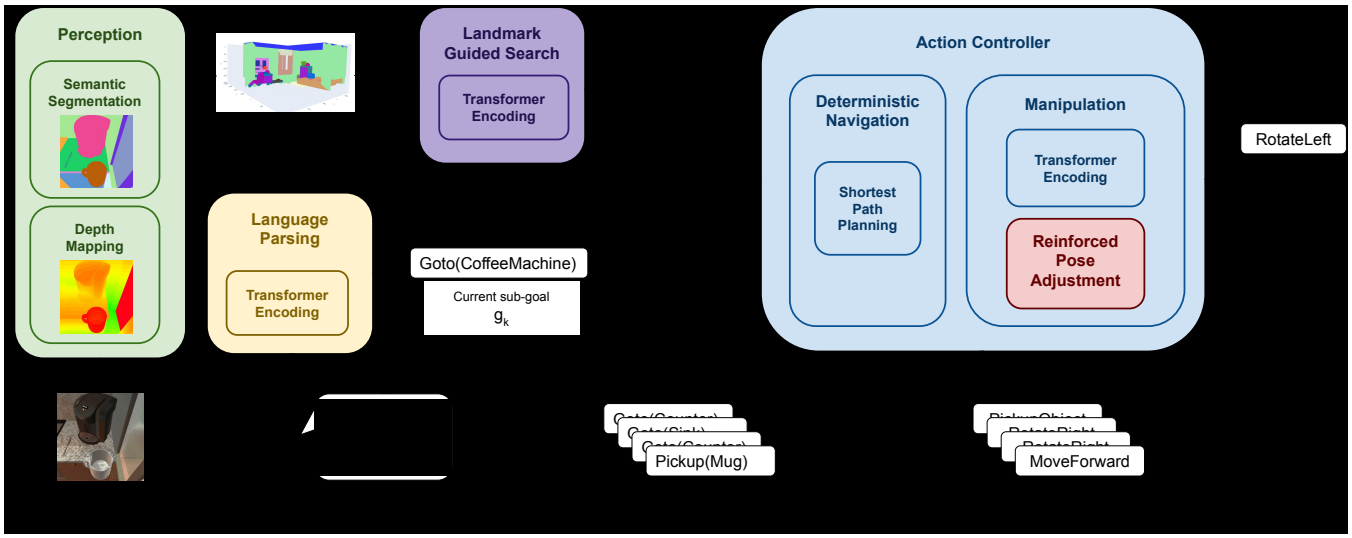

Fig. 2. The proposed architecture is a composition of five modules. The language parsing module transforms the language instructions into parameterized sub-goals. The perception module persists visual observations to a 3D semantic voxel map. The search module determines the next target waypoint. The navigation module uses deterministic planning to generate navigation actions. The manipulation module generates manipulation actions with assistance from reinforced pose adjustment when necessary.

At each timestep t, the perception module updates a persistent spatial semantic map using the current egocentric RGB observation image o_t. The search module performs a landmark guided search over the observed semantic map to update a priority queue of relevant spatial positions and determine a target position. The navigation module uses a deterministic policy to navigate to the target position. After reaching the target position, the sub-goal counter k is incremented and we predict the next sub-goal.

For manipulation sub-goals, the manipulation module predicts the next action a_t given the natural language instruction L, the sequence of previous sub-goals $\langle g_i, \rangle_{i<k}$, and the sequence of previous manipulation actions $\langle a_i, \rangle_{i<n}$. If the manipulation action succeeds, we continue sampling manipulation actions until the action a_{END} is sampled at which point we increment the sub-goal counter k and predict the next sub-goal. If the manipulation action fails, we attempt to use reinforced pose adjustment (RPA) to recover from the failure. If the manipulation action still fails after pose adjustment we discard the current sub-goal g_k and the current target position.

The episode ends when the sub-goal g_{STOP} is sampled or the maximum horizon is exceeded.

A. Language Processing

The language processing module is responsible for parsing the natural language instructions into parameterized sub-goals that can be used for planning. The module consists of two pre-trained BERT [44] transformer models that have been fine-tuned on the ALFRED training dataset. The first transformer predicts the next sub-goal type, $type_k$, and the second transformer predicts the target object of the next sub-goal, $target_k$. The input to both models is the natural language instructions L, and the sequence of past sub-goals $\langle g_i, \rangle_{i<k}$. The sequence of past sub-goals is converted to natural language phrases. For example, the sub-goal `GotoLocation(CounterTop)` would be converted to *"go to counter top"* and the sub-goal `PickupObject(Apple)` would be converted to *"pick up apple"*. The intuition behind this conversion is that it allows us to take advantage of semantic relationships between the sub-goals and instructions that have already been learned by the pre-trained language models. The phrases are joined to produce the sub-goal history H_k. The inputs L and H_k are concatenated together and encoded by the two pre-trained BERT transformers. The first model predicts a distribution over the next sub-goal type $P(type_k|L, H_k)$ and the second model predicts a distribution over the next sub-goal target object $P(target_k|L, H_k)$. We sample from these distributions to select the next sub-goal type $type_k$ and the next target object $target_k$.

B. Perception

At timestep t, the input to the perception module is the current egocentric RGB image observation o_t. Following prior works [33], we use two neural networks based on the U-Net [45] architecture to predict the semantic segmentation o_t^S and depth map o_t^D from the current RGB image observation. Both neural networks are trained on images from the ALFRED training dataset. The depth map o_t^D is then transformed to a point cloud using a pinhole camera model and each point is assigned a semantic label based on o_t^S. We persist these observations in a 3D semantic voxel map $V_t \in \{0,1\}^{X \times Y \times Z \times C}$ where the value at each voxel is the element-wise maximum of the class distributions across all points that land within the voxel. The voxel map is updated at each timestep t and aggregated over time.

C. Navigation

When the current sub-goal involves navigation ($type_k \in \{GotoLocation\}$), the navigation module is responsible for generating actions to navigate toward the goal location. To facilitate navigation, we compute a top-down 2D map of the environment, $M_t \in \{0,1\}^{X \times Y \times C}$, by summing over the height of the voxel map V_t. Target locations on the map are specified as waypoints. A waypoint is a tuple (x, y, ω_y) where (x, y) is a position in the map M_t and ω_y is a yaw angle. Once a target location is specified, the navigation module uses a Djikstra-based deterministic planner to generate navigation actions.

D. Landmark Guided Search

The search module is responsible for finding target locations in the map M_t. Our approach is to prioritize observed locations of the target object $target_k$, and to use predicted landmarks to efficiently explore the environment when $target_k$ is unobserved.

To track relevant waypoints and facilitate efficient exploration, we maintain two queues of waypoints: a high priority queue Q_{high} and a low priority queue Q_{low}. At each timestep t the queue is updated based on the latest semantic voxel map. We first search the voxel map for any voxels containing the current target object $target_k$, which we add to the high priority queue Q_{high}. Next, we use a landmark prediction model to predict an object class $landmark_k$ that is likely to lead us to the target object. We then search the voxel map for any voxels containing $landmark_k$ and add those voxels to the low priority queue Q_{low}.

To select the next target waypoint w, the navigation module chooses the next waypoint from Q_{high}. If Q_{high} is empty then the next waypoint from Q_{low} is selected. If we have exhausted all of the waypoints in both Q_{high} and Q_{low}, then navigation module randomly samples from the observed reachable floor voxels in M_t.

To compute waypoints for a given object class, we search the voxel map for any voxels containing the object. We cluster all of the resulting voxels into groups of connected voxels and sort by descending group size to prioritize the larger clusters of voxels. Within each group we sort voxels by their distance to the centroid of the cluster to promote voxels that are centered on their respective clusters. For each relevant voxel, we compute a waypoint by selecting the closest navigable position in map M_t and a yaw angle such that the robot is facing the target voxel.

We train a landmark prediction model to predict the object class $landmark_k$ that is most likely to lead us to the target object class $target_k$. Because the language instructions can sometimes contain references to relevant landmarks, the landmark prediction model is conditioned on the natural language instructions L and the action history H_k in addition to the target object class $target_k$. All three inputs, L, H_k, and $target_k$ are concatenated together and encoded by a BERT transformer model that predicts a distribution over the landmark object class $P(landmark_k | target_k, L, H_k)$, from which we sample $landmark_k$. To train the model, we process a new dataset from the ALFRED training split containing only examples of navigation actions where the expert policy navigated to a large receptacle object immediately before interacting with a target object. Because the model is conditioned on both the language instructions and the target object $target_k$, it can utilize clues in the natural language instructions in addition to prior experience of large receptacles that are likely to lead to $target_k$.

E. Manipulation with Reinforced Pose Adjustment

When the current sub-goal type is manipulation ($type_k \notin \{GotoLocation\}$), the manipulation module is responsible for generating actions to satisfy $type_k$. The manipulation module takes as input the natural language instructions L, the history of sub-goals H_k, and the sequence of previous actions $\langle a_i, \rangle_{i<t}$. Like the sub-goal history, the past actions are converted to natural languages phrases. For example, the manipulation action $CloseObject(Cabinet)$ is converted to *"close cabinet"*. The past actions are then joined to produce the action history H_k^A. The three inputs L, H_k, and H_k^A are concatenated together and encoded by a BERT transformer model which is used to predict a distribution over the actions $P(a_t | L, H_k, H_k^A)$, from which we sample the next action a_t. The resulting action a_t is then converted to natural language, concatenated to the other inputs, and encoded by a second BERT transformer model which is used to predict the distribution over argument object classes $P(obj_t | a_t, L, H_k, H_k^A)$, from which we sample the argument object obj_t. In ALFRED, manipulation actions are parameterized by a pixelwise mask to denote the target object in the egocentric RGB image observation. To select this mask, we use the semantic segmentation o_t^S from the perception module and select all pixels from the channel corresponding to the object class obj_t. Because the semantic segmentation image can be noisy, we threshold the selected masks based on statistics from the training dataset. Specifically, we consider a mask valid only if it is above the 5th-percentile of mask sizes used to interact with the target object in the training set. This mitigates hallucinated objects and ensures that the agent has a clear view of the target object before trying to manipulate it.

Often the agent is unable to immediately manipulate the target object obj_t from the current waypoint w. This is typically due to the robot being unable to detect the object or due to some environmental constraints (for example, to open a cabinet door the robot must be close enough to reach the handle while far enough away that the robot doesn't block the door). To handle these cases, our manipulation module includes a pose adjustment policy π_p used when the agent is unable to immediately manipulate the target object class obj_t from the current waypoint w. The policy must select from a set of predefined pose adjustment action sequences: `StepBack`, `LookUp`, `LookDown`, `StepBackAndLookUp`, `StepBackAndLookDown`. After executing the pose adjustment sequence the agent will retry the manipulation action $action_k$. Additionally, the agent can choose to `Renavigate`, which forces the agent to

Method	Test Seen				Test Unseen			
	SR	**GC**	**PLWSR**	**PLWGC**	**SR**	**GC**	**PLWSR**	**PLWGC**
Low-level step-by-step instructions + High-level goal instructions								
Seq2Seq [3]	3.98	9.42	2.02	6.27	3.9	7.03	0.08	4.26
MOCA [29]	22.05	28.29	15.10	22.05	5.30	14.28	2.72	9.99
E.T. [28]	38.42	45.44	27.78	34.93	8.57	18.56	4.10	11.46
LWIT [32]	30.92	40.53	**43.10**	**36.76**	9.42	20.91	5.60	16.34
HiTUT [30]	21.27	29.97	11.10	17.41	13.87	20.31	5.86	11.51
ABP [31]	**44.55**	**51.13**	3.88	4.92	15.43	24.76	1.08	2.22
FILM [4]	27.67	38.51	11.23	15.06	26.49	36.37	10.55	14.30
LGS-RPA (ours)	40.05	48.66	21.28	28.97	**35.41**	**45.24**	**15.68**	**22.76**
High-level goal instructions only								
LAV [46]	13.35	23.21	6.31	13.18	6.38	17.27	3.12	10.47
HLSM [33]	25.11	35.79	6.69	11.53	16.29	27.24	4.34	8.45
FILM [4]	25.77	36.15	10.39	14.17	24.46	34.75	9.67	13.13
LGS-RPA (ours)	**33.01**	**41.71**	**16.65**	**24.49**	**27.80**	**38.55**	**12.92**	**20.01**

TABLE I

TEST RESULTS OF THE ALFRED CHALLENGE INCLUDING SEVERAL METRICS: SUCCESS RATE (SR), GOAL CONDITIONED SUCCESS (GC), PATH LENGTH WEIGHTED SUCCESS RATE (PLWSR), AND PATH LENGTH WEIGHTED GOAL CONDITIONED SUCCESS (PLWGC).

select a new target waypoint from which to attempt the manipulation action. We train the initial pose correction policy using offline reinforcement learning in a contextual bandit setting. First we iterate over each training example and perform end-to-end rollouts. Every timestep at which the robot fails to manipulate an object is recorded and aggregated into a dataset of failed states. Next we iterate over the failed states and from each example failed state the agent explores each of the possible action sequences. The reward for each action sequence is computed from the world state. Specifically, the agent receives a positive reward if the agent is able to successfully manipulate the target object after invoking an action sequence. We model the policy as a neural network that receives as input the current target object $target_k$, the attempted manipulation action $action_k$, the robot's egocentric distance to the target waypoint d, the maximum height h of the target object in the voxel map, and the type of failure $f \in \{UndetectedObj, EnvConstraint\}$. After iterating over all of the starting states, we update the parameters of π_p using AdamW stochastic optimization [47].

Inspired by the DAgger algorithm [48], we further show that the pose correction model can incorporate human-in-the-loop feedback. Using an initial pose adjustment policy π_p, we again iterate over each training example and perform end-to-end rollouts to record a dataset of failed states. We then sample failed states and present them to a human expert for annotation. The human expert is shown the robot's egocentric RGB observation at the failed state, the target object $target_k$, and the attempted manipulation action $action_k$. The human expert is asked to assist the robot by choosing the best action sequence to invoke from the set of predefined pose adjustment action sequences. We aggregate these annotations to produce an annotated pose adjustment dataset $\mathcal{D}_P$ which we use to update the parameters of π_p using AdamW stochastic optimization.

V. EXPERIMENTS

A. Experimental Setup

Dataset: The ALFRED dataset contains 8,055 expert trajectories averaging 50 steps each, resulting in 428,322 image-action pairs. Each expert trajectory is annotated with multiple language directives, which consist of a high-level goal statement and a set of low-level instructions, for a total of 25k language annotations. The expert trajectories are grouped into sub-goals, with each sub-goal corresponding to one of the low-level instructions. Each sub-goal is parameterized with a target object and an optional receptacle object. The dataset is split into training, validation, and test folds. The validation and test folds are further split into two conditions: seen and unseen environments.

Metrics: Success Rate (SR) is the fraction of rollouts for which the object positions and state changes completely satisfy the task goal-conditions at the end of the action sequence. Goal Condition Success Rate (GC) is the fraction of goal-conditions completed at the end of an episode to those necessary to have finished a task. Path Length Weighted Success Rate (PLWSR) is the success rate weighted by rollout length. Path Length Weighted Goal Condition Success Rate (PLWGC) is the goal-condition success weighted by rollout length.

Baselines: There are two types of baselines on the ALFRED benchmark: those that use the low-level step-by-step instructions [3], [28], [30]–[32] and those that use only the high level instructions [4], [33], [46]. Successfully completing tasks based only on the high-level instructions is desirable because the low-level step-by-step instructions can be too tedious and unrealistic for a non-expert human to provide. However, this presents additional difficulties because the high-level instructions can be ambiguous and under-specified. Our approach can be evaluated in either context so we compare our results to both types of baselines.

Training Details: The BERT classification models used by the language processing, landmark guided search, and manipulation planning modules use pre-trained "distilbert-base-uncased" weights from the `Transformers` package [49] fine-tuned on examples processed from the ALFRED training set using AdamW with a learning rate of 5e-5. The U-Net models used for perception are trained by prior work [33]. The pose adjustment models used by the manipulation planning module were trained using AdamW and a learning rate of 1e-3.

B. Quantitative Results

In Table I we compare our approach with state-of-the-art methods for the two test sets of the ALFRED challenge. When the low-level step-by-step instructions are included, our approach achieves state-of-the-art performance in unseen environments and competitive performance in seen environments. ABP [31] and LWIT [32] perform better on the seen environments, potentially reflecting more overfitting to the environments and directives seen during training. In the setting with only high-level instructions our approach achieves state-of-the-art performance across both seen and unseen environments. Notably, in unseen environments our approach with only high-level instructions outperforms even the previous state-of-the-art methods that utilize the low-level step-by-step instructions.

Using the validation split as a development dataset, we show additional insights in Table II by comparing our base method to agents that use ground truth perception (+ gt perception), ground truth language parsing (+ gt lang), or both (+ gt perception, lang). We find that ground truth language parsing only provides negligible benefits (1.83% absolute improvement on unseen and 0.65% on seen) indicating that the learned language parsing model is already very strong. Ground truth perception provides a significant performance increase (21.5% absolute improvement on unseen and 5.65% on seen) indicating that there is large room for improvement in the agent's perception capabilities.

In Table III we compare the success rate of different learning strategies for the pose adjustment policy: Reinforcement Learning (RL), Human-in-the-Loop Feedback (HITL), and a combination of both (RL + HITL). We find that all three strategies have competitive performance and we get the highest success rate from a combination of reinforcement learning and human feedback.

In Table IV we compare the success rate of Landmark Guided Search with a random search and we find that Landmark Guided Search increases performance by 4.34% in unseen environments and 3.72% in previously seen environments.

C. Qualitative Results

The qualitative results of Landmark Guided Search are illustrated in Figure 3. In the first example the robot is searching for a mug that is initially too far away to detect in the cluttered scene. Our landmark prediction model infers that the counter-top is a likely landmark and by following the counter-top the robot quickly finds the mug. The random search eventually finds the mug but it wastes many timesteps exploring unnecessary areas of the environment. The benchmark tasks are made up of several steps over a long-horizon and we must complete the task in a limited amount of time so it's important to be as efficient as possible for every step. In the second example the robot is searching for a desk lamp that is initially occluded around a corner. The landmark guided search model determines that a desk is a good landmark candidate and quickly finds the lamp after searching the visible desks in the scene. The random search never finds the lamp because it ends up failing due to collision while exploring random areas of the scene. This illustrates that inefficient exploration also exposes the robot to more opportunities for collision and other failure modes.

Figure 4 contains examples illustrating the use of Reinforced Pose Adjustment. In the first example sequence the robot has planned to open a microwave, but the microwave is mounted above the stove so it is not immediately visible to the robot at the target waypoint. Through experience, the policy has learned to adjust the robot heading upward which allows the robot to successfully interact with the microwave. In the second example sequence the robot needs to open a refrigerator door but initially the robot's body is blocking the door. The policy has learned to adjust it's pose by moving backward in order to successfully open the refrigerator door.

Method	Val Seen		Val Unseen	
	SR	GC	SR	GC
Base Method	43.86	52.51	33.18	44.68
+ gt perception	49.51	58.02	54.68	60.80
+ gt lang	44.51	53.91	35.01	44.71
+ gt perception, lang	51.77	61.32	61.76	64.49

TABLE II

DEVELOPMENT RESULTS ON THE VALIDATION SPLITS WITH GROUND-TRUTH PERCEPTION AND LANGUAGE PARSING ORACLES.

Method	Val Seen		Val Unseen	
	SR	GC	SR	GC
RL	41.56	50.12	30.41	42.79
HITL	42.98	51.64	31.72	44.16
RL + HITL	43.86	52.51	33.18	44.68

TABLE III

DEVELOPMENT RESULTS ON THE VALIDATION SPLITS SHOWING THE EFFECT OF POSE ADJUSTMENT TRAINED WITH REINFORCEMENT LEARNING (RL), HUMAN-IN-THE-LOOP FEEDBACK (HITL), OR BOTH.

Method	Val Seen		Val Unseen	
	SR	GC	SR	GC
Random search	38.12	49.71	29.02	40.34
Landmark Guided Search	43.86	52.51	33.18	44.68

TABLE IV

DEVELOPMENT RESULTS ON THE VALIDATION SPLITS SHOWING THE EFFECT OF LANDMARK GUIDED SEARCH COMPARED TO RANDOM SEARCH.

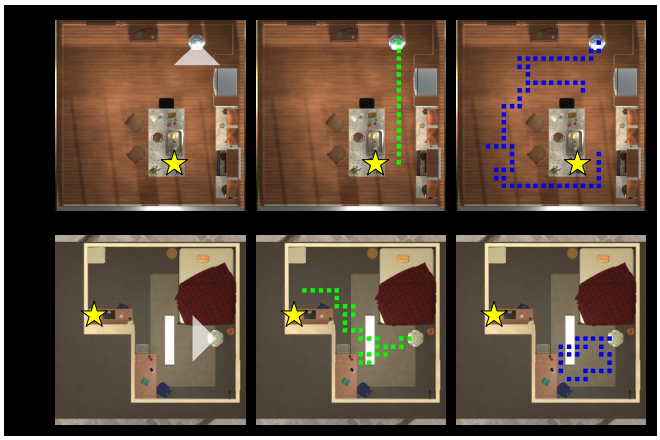

Fig. 3. Two examples depicting top-down views of unseen environments qualitatively illustrating the use of landmarks as beacons for navigation. In each row we first show the robot's field of view (left), followed by the landmark based search route depicted in green (center), and finally a random search route depicted in blue (right). In all images the target object is depicted by a star.

Fig. 4. Two example sequences from unseen environments illustrating the use of learned pose adjustment. In sequence (a) the robot has planned to interact with a microwave but the microwave is mounted above the robot's heading at this waypoint. Through experience, the robot has learned to adjust it's heading to successfully interact with the microwave. In (b) the robot needs to open a refrigerator door but the door is blocked by the robot at this waypoint. The robot has learned to move backward in order to successfully open the refrigerator door.

VI. CONCLUSION

In the context of human-centric robot deployments, we addressed the problem of natural language instruction following on a mobile manipular robot. We showed that our approach using an efficient landmark-based search system combined with an adaptive pose adjustment policy enables state-of-the-art performance on the popular ALFRED instruction following challenge. While effective, our pose adjustment policy is quite simple and limits the robot to a fixed set of adjustment strategies. Future work should explore more flexible recovery policies with more sophisticated training regimes. Additionally, reinforcement learning and human-in-the-loop feedback could potentially be utilized to improve other modules or complete end-to-end models. Finally, while our methods are designed with robot platforms in mind, we have evaluated our approach in simulated environments and we leave exploration of physical deployment challenges to future work.

ETHICS STATEMENT

We study natural language as a means of instructing a mobile manipulator robot. While natural language is an intuitive interface for non-expert humans, other modalities should additionally be considered to develop robot interfaces that are accessible to humans with speech-impairment or other disabilities that prohibit natural language instruction.

Our research is situated in home environments simulated by AI2-THOR [27] which may bias the results towards North American homes. Future work should incorporate a more diverse set of environments.

ACKNOWLEDGMENTS

We thank the authors of ALFRED for their work on the benchmark and the authors and maintainers of AI2-THOR for their work on the simulation environments.

This work was supported by the National Science Foundation, collaborative awards IIS-1924435 and IIS-1925043 "Program Verification and Synthesis for Collaborative Robots."

REFERENCES

[1] E. Chan, O. Baumann, M. Bellgrove, and J. Mattingley, "From objects to landmarks: The function of visual location information in spatial navigation," *Frontiers in Psychology*, vol. 3, 2012.

[2] O. Hardt and L. Nadel, "Cognitive maps and attention," in *Attention*, ser. Progress in Brain Research, N. Srinivasan, Ed., 2009, vol. 176, pp. 181–194.

[3] 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 *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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

[5] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek, "Robots that use language," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 25–55, 2020.

[6] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions," *Def*, vol. 2, no. 6, p. 4, 2006.

[7] 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.

[8] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller, and N. Roy, "Understanding natural language commands for robotic navigation and mobile manipulation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, 2011, pp. 1507–1514.

[9] V. Blukis, D. Misra, R. A. Knepper, and Y. Artzi, "Mapping navigation instructions to continuous control actions with position-visitation prediction," in *Conference on Robot Learning*. PMLR, 2018, pp. 505–518.

[10] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 3515–3522.

[11] M. R. Walter, S. M. Hemachandra, B. S. Homberg, S. Tellex, and S. Teller, "Learning semantic maps from natural language descriptions." Robotics: Science and Systems, 2013.

[12] R. Cantrell, K. Talamadupula, P. Schermerhorn, J. Benton, S. Kambhampati, and M. Scheutz, "Tell me when and why to do it! run-time planner model updates via natural language instruction," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, 2012, pp. 471–478.

[13] D. Bohus and E. Horvitz, "Facilitating multiparty dialog with gaze, gesture, and speech," in *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction*, 2010, pp. 1–8.

[14] S. Tellexll, P. Thakerll, R. Deitsl, D. Simeonovl, T. Kollar, and N. Royl, "Toward information theoretic human-robot dialog," *Robotics*, p. 409, 2013.

[15] J. Thomason, S. Zhang, R. J. Mooney, and P. Stone, "Learning to interpret natural language commands through human-robot dialog," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[16] G. Kuhlmann, P. Stone, R. Mooney, and J. Shavlik, "Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer," in *The AAAI-2004 workshop on supervisory control of learning and adaptive systems*. San Jose, CA, 2004.

[17] M. Bollini, S. Tellex, T. Thompson, N. Roy, and D. Rus, "Interpreting and executing recipes with a cooking robot," in *Experimental Robotics*. Springer, 2013, pp. 481–495.

[18] D. Misra, J. Langford, and Y. Artzi, "Mapping instructions and visual observations to actions with reinforcement learning," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017.

[19] V. Blukis, Y. Terme, E. Niklasson, R. A. Knepper, and Y. Artzi, "Learning to map natural language instructions to physical quadcopter control using simulated flight," in *Conference on Robot Learning (CoRL)*, 2019.

[20] V. Blukis, R. A. Knepper, and Y. Artzi, "Few-shot object grounding for mapping natural language instructions to robot control," in *Conference on Robot Learning (CoRL)*, 2020.

[21] 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*, 2018, pp. 3674–3683.

[22] H. De Vries, K. Shuster, D. Batra, D. Parikh, J. Weston, and D. Kiela, "Talk the walk: Navigating new york city through grounded dialogue," *arXiv preprint arXiv:1807.03367*, 2018.

[23] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer, "Vision-and-dialog navigation," in *Conference on Robot Learning*. PMLR, 2020, pp. 394–406.

[24] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," in *Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 2020.

[25] M. Z. Irshad, C.-Y. Ma, and Z. Kira, "Hierarchical cross-modal agent for robotics vision-and-language navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021. [Online]. Available: https://arxiv.org/abs/2104.10674

[26] 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 (CVPR)*, 2018.

[27] 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.

[28] A. Pashevich, C. Schmid, and C. Sun, "Episodic Transformer for Vision-and-Language Navigation," in *ICCV*, 2021.

[29] K. P. Singh, S. Bhambri, B. Kim, R. Mottaghi, and J. Choi, "Factorizing perception and policy for interactive instruction following," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1888–1897.

[30] Y. Zhang and J. Chai, "Hierarchical task learning from language instructions with unified transformers and self-monitoring," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 4202–4213.

[31] B. Kim, S. Bhambri, K. P. Singh, R. Mottaghi, and J. Choi, "Agent with the big picture: Perceiving surroundings for interactive instruction following," in *Embodied AI Workshop CVPR*, 2021.

[32] V.-Q. Nguyen, M. Suganuma, and T. Okatani, "Look wide and interpret twice: Improving performance on interactive instruction-following tasks," in *IJCAI*, 2021.

[33] V. Blukis, C. Paxton, D. Fox, A. Garg, and Y. Artzi, "A persistent spatial semantic representation for high-level natural language instruction execution," in *Conference on Robot Learning*. PMLR, 2021, pp. 706–717.

[34] Z. Jia, K. Lin, Y. Zhao, Q. Gao, G. Thattai, and G. Sukhatme, "Learning to act with affordance-aware multimodal neural slam," 2021.

[35] A. Kurenkov, R. Martín-Martín, J. Ichnowski, K. Goldberg, and S. Savarese, "Semantic and geometric modeling with neural message passing in 3d scene graphs for hierarchical mechanical search," 2021.

[36] X. Wang, W. Xiong, H. Wang, and W. Y. Wang, "Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation," in *ECCV*, 2018.

[37] X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *CVPR*, 2019.

[38] J. Li, X. Wang, S. Tang, H. Shi, F. Wu, Y. Zhuang, and W. Y. Wang, "Unsupervised reinforcement learning of transferable meta-skills for embodied navigation," in *CVPR*, 2020.

[39] H. Wang, Q. Wu, and C. Shen, "Soft expert reward learning for vision-and-language navigation," in *European Conference on Computer Vision (ECCV'20)*, 2020.

[40] S. Ross, G. J. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning." in *AISTATS*, ser. JMLR Proceedings, G. J. Gordon, D. B. Dunson, and M. Dudík, Eds., vol. 15. JMLR.org, 2011, pp. 627–635.

[41] S. Ross, N. Melik-Barkhudarov, K. S. Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, "Learning monocular reactive uav control in cluttered natural environments," 2012.

[42] V. G. Goecks, G. M. Gremillion, V. J. Lawhern, J. Valasek, and N. R. Waytowich, "Efficiently combining human demonstrations and interventions for safe training of autonomous systems in real-time," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019.

[43] W. Saunders, G. Sastry, A. Stuhlmüller, and O. Evans, "Trial without error: Towards safe reinforcement learning via human intervention," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, ser. AAMAS '18. International Foundation for Autonomous Agents and Multiagent Systems, 2018, p. 2067–2069.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 2019, pp. 4171–4186.

[45] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234–241.

[46] K. Nottingham, L. Liang, D. Shin, C. Fowlkes, R. Fox, and S. Singh, "Lav," 2021. [Online]. Available: https://leaderboard.allenai.org/alfred/submission/c2cm7eranqs9puf9uvjg

[47] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[48] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 15. PMLR, 11–13 Apr 2011, pp. 627–635.

[49] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.