Common sense and Semantically Grounded Understanding for Effective Room Navigation in Embodied Environment

Abstract

Previous multi-modal embodied navigation research focused on completing tasks by aligning image features with natural language instructions. In this paper, we investigate if intrinsic features such as common sense and semantic understanding, which are critical for human navigators but ignored in previous research, also help artificial agents navigate in realistic 3D environments given a high-level linguistic instruction. From our experiments, we observe that common sense helps the agent in long-term planning, while semantic understanding helps the agent in local planning in the room navigation ($RoomNav$) task. We also propose a novel semantically-guided self-supervision mechanism which further improves the performance of the agent on unseen environments. The cross-modal embeddings learned during training suggest that common sense and semantic understanding helps in capturing the structural and positional patterns of the environment, implying that the agent benefits by inherently learning a map of the environment.

Introduction

Most previous embodied agent research focuses on combining language and visual inputs (Das et al. 2017, 2018; Gordon et al. 2017; Manolis Savva et al. 2019; Mirowski et al. 2018; Anderson et al. 2017; Fried et al. 2018; Wang et al. 2018). However, recent research suggests that using language instructions alone can outperform models with visual features (Anand et al. 2018; Hu et al. 2019). It raises the question of what the agent actually learns from both the visual and language inputs, and if there is any underlying features that the agent can benefit from.

Most past research ignores intrinsic features such as common sense of the environment settings and encoded scene-relevant information such as semantic understanding. Thus, previous agents need to rely on step-by-step instructions (Shridhar et al. 2020) to navigate to the target successfully, especially in a new environment. In comparison, humans do not require low-level instructions such as "go straight for five meters, and turn left at the end of the hallway" to navigate to the restroom in a new restaurant. Instead, humans leverage intrinsically embedded features such as scene information and common sense of room layouts to navigate in an unseen environment. Automated agents, however, have difficulties in performing grounded language navigation tasks with abstract, high-level instructions such as "go to the kitchen" in realistic unseen 3D environments (Tangiuchi et al. 2019). We hypothesize that common sense of room layout and semantic understanding of the environment can benefit agents in a similar way as they benefit humans. Specifically, common sense of room layout can assist path planning by setting the general course of the trajectory. For instance, when navigating to the kitchen, it is useful to know that a dining room is usually close to a kitchen. On the other hand, semantic understanding of the room (i.e. objects in each room, etc.) supports better local actions. For example, agents should stop when the target room is reached.

In this work, we explore the role of common sense room layout and room semantic understanding in the concept-driven room navigation task. Figure 1 describes an example of the $RoomNav$ task. An agent is spawn randomly in a realistic 3D house environment with a given instruction (e.g. " go to the restroom"). Then the agent has to navigate to the target room by performing a sequence of actions: turn left, turn right, move forward, or stop. This paper's objective is not to build an agent which outperforms the state-of-the-art in the $RoomNav$ task. Instead, the primary focus of our research is to explore the research question related to grounded language understanding without data bias issues seen in previous research: can intrinsic features such as common sense and semantic grounded understanding of the environment also help the agent navigate with high-level instructions? Our contributions to address the problem are the following: (i) we proposed novel ways to incorporate common sense and semantic understanding within the artificial agents to address a complex task in the multi-modal setting inspired by humans (ii) proposed semantically-guided self-supervised imitation learning (*SIL*) mechanism for grounding to fine-tune the agent on unseen environments for generalization ability (iii) showed that common sense facilitates long-term while semantic grounding facilitates local planning, and (iv) demonstrated that the reason common sense and semantic grounded understanding help with navigation is by mapping learned instruction embeddings to the scenes.

Figure 1: Illustration of the RoomNav task. At each timestep, the agent observes a panoramic view (left, front and right views concatenated) with dining room on the left, living room ahead, wall on the right, and hallway being current. The agent is spawned in a random location and is asked to navigate to the target room with a high-level instruction (”Go to the kitchen”) using four possible actions: turn right, turn left, go forward, and stop.

Related Work

Previous related research in embodied environments lie in comparing vision and language-grounded tasks, exploring potential underlying features of the environment, and making agents more robust towards unseen environments.

Vision and Language Grounded Tasks: Embodied question answering (Das et al. 2017) and instruction following (Anderson et al. 2017; Shridhar et al. 2020) in embodied environments have been popular to study the interaction between language and visual inputs. We choose the $RoomNav$ task to test the research question that whether equipping an agent with similar high-level inputs as humans (common sense and semantic understanding) can help with downstream navigation tasks by eliminating other factors such as data bias seen in instruction following tasks (Hu et al. 2019).

Common Sense and Understanding: Some recent research explores semantic representation and common sense knowledge graph in object navigation tasks in simpler environment settings (Hermann et al. 2017). Mousavian et al. (2018) use pre-trained object detection or segmentation models to represent semantics to navigate to five semantic goals in nine homes. Yang et al. (2019) extract relationships among objects into a knowledge graph with a Graph Convolution (Kipf and Welling 2017) encoding as priors to the navigation model. Gupta et al. (2017) propose a spatial memory map by projecting environment information to a 2D matrix. Recently, Wu et al. (2019) proposed to model relational memory among room types in navigation tasks using a Bayesian probabilistic relational graph. We instead adopt a simple backward language model to model common sense. In our work, we train the agent to learn semantics and common sense together in navigation. Instead of abstracting visual and language representations, we illustrate whether providing these inputs can help with embodied tasks. In addition, we leverage the learned models and further fine-tune the agent in novel environments. We also demonstrate the causality by analyzing what the agent learns.

Robustification: Several studies have analyzed robustification and generalization to unseen environments, using methods such as reinforcement learning and semi-supervised learning. Manolis Savva et al. (2019) apply Proximal Policy Optimization (Schulman et al. 2017) for point-nav task guided by a very strong signal of the relative distance between the agent and the target coordinate. Wang et al. (2018) fine-tune the agent on unseen environments using a cycle-reconstruction loss obtained by reversing the original instruction following problem (Fried et al. 2018). For a similar $RoomNav$ task, Wu et al. (2018a) use Deep Deterministic Policy Gradient (Heess et al. 2015) and Asynchronous Advantage Actor Critic (Mnih et al. 2016) on the semantically rich House3D (Wu et al. 2018b) environment. These learned policies do not leverage any intrinsic common sense and knowledge-grounded semantic information available in the environment. We perform SIL by introducing auxiliary tasks related to semantic understanding to make the model generalize to unseen environments better. Furthermore, we analyze the common sense that the agent learns from SIL and why SIL improves the performance on unseen environments by evaluating the understanding of the agent on the input instructions.

Common Sense and Semantically Grounded Agent

We first introduce the agent architecture, and then the learning process.

Agent Architecture

Our architecture consists of four components: Base Navigation, Common Sense Planning, Semantic Grounding, and Semantic-Grounded Navigator. We also explain how the agent functions can be fine-tuned on unseen environments without annotations. Figure 2 shows the entire architecture framework and Figure 6 in the Appendix depicts detailed architecture with model information along with loss functions.

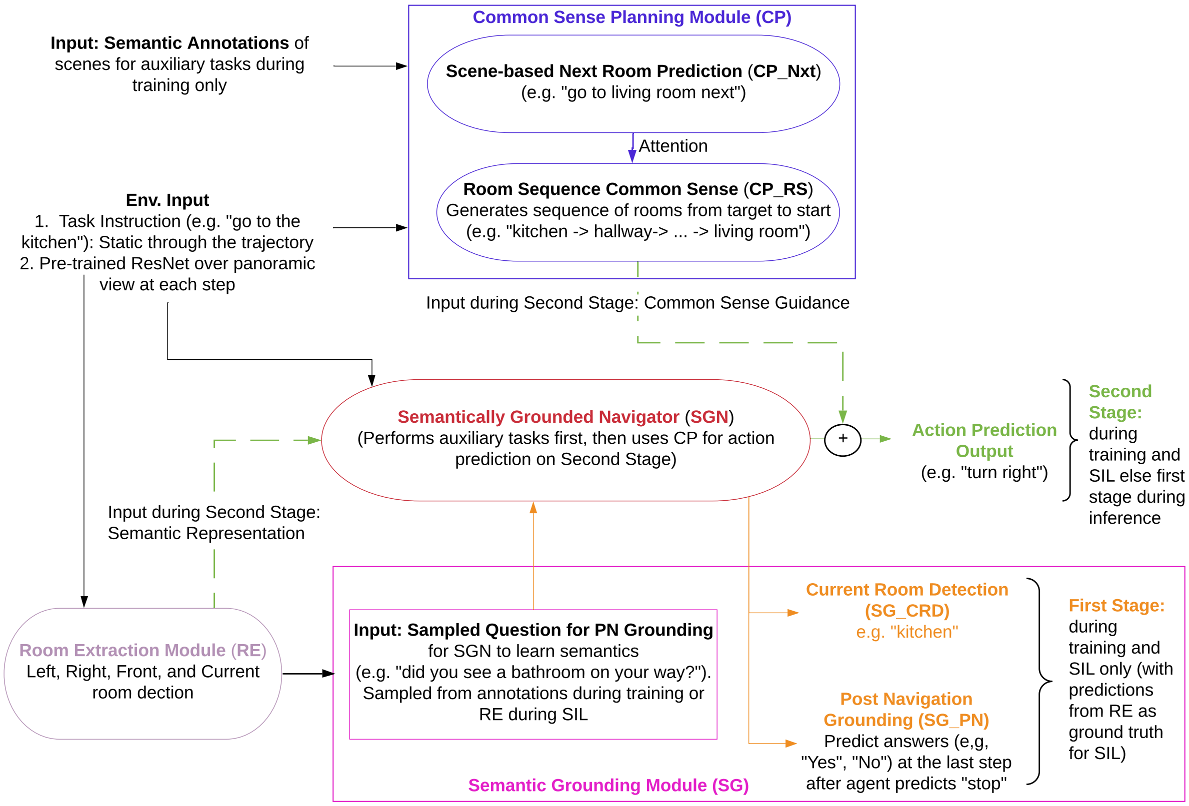

Figure 2: High-level architecture of Common Sense and Semantically Grounded Navigation model. Red components correspond to Base Navigation model. Purple components are introduced to incorporate common sense while pink components are used for semantic understanding. The Semantically Grounded Navigator (SGN) is designed to perform semantic understanding for action prediction, while common sense is fed as guidance for better planning. Dotted lines indicate input in the second pass in the model. A detailed architecture with model information and illustration with all the loss functions can be found in Figure 6 in the Appendix.

Base Navigation We use an LSTM-based (Hochreiter and Schmidhuber 1997) navigation model following the work in Das et al. (2017) to train a navigation agent which predicts an action given the current state. The input to the LSTM is the context of the path, $RoomNav$ instructions (e.g. "go to the kitchen"), a visual representation of the scene, and the previous action following Das et al. (2017). We choose the simple LSTM baseline as opposed to a state-of-the-art navigation model because the components we introduce are orthogonal to the previous contributions which did not utilize this available information from the environment. In addition, using a simple model reduces the influence from interacting with different complex modules, and thus disentangles the contribution of incorporating common sense and semantic understanding.

Common Sense Planning Module (CP) We hypothesize that realistic house environments follow common sense such as structural patterns (e.g. a refrigerator is usually placed in the kitchen) and sequential patterns (e.g. a kitchen is usually near a dining room). To incorporate such common sense information in the agent, we design two auxiliary tasks in the CP module: Scene-based Next Room Prediction (CP_Nxt) and Room Sequence Common Sense (CP_RS) using room sequences observed from training environments. CP_Nxt module enables the agent to predict and navigate to intermediate targets to deal with the fact that the target room in the instruction may be distant and thus hard to interpret. Specifically, we train the agent to predict the next room as an intermediate goal at every time step. For instance, in Figure 1, the agent is in the hallway, and the CP_Nxt module will predict the next room to move to as "dining room", since it is the closest room on the potential path to the target location (the kitchen).

The CP_Nxt module is hinted by the current scene and the target room from the instruction, but it does not take into account what rooms are connected to the target room. To rectify such misalignment, we design CP_RS to generate backward room sequences that start from the target room using an LSTM model, similar to an auto-regressive language model. We then obtain a contextual representation by attending CP_Nxt hidden state to each state representation in the output of the CP_RS module before predicting the next room accordingly.

Semantic Grounding Module (SG) We design two submodules in SG to help the agent to understand the semantics of the environment: (i) Current Room Detection (SG_CRD) for detecting the current room at each timestep to capture local semantics and (ii) Post Navigation Grounding (SG_PN) for understanding and capturing the global semantics along the trajectory. We cast SG_CRD as a multi-label classification problem to detect the room type on top of the hidden states from the base navigation LSTM. In comparison, we cast the SG_PN as a binary classification problem to predict if there is a certain room type on the trajectory using the last hidden state with a sampled question (e.g. "did you see a bathroom on the way") from the environment annotations.

Semantic-grounded Navigator (SGN) SGN incorporates CP and SG into the Base Navigation model. Our SGN first adopts the LSTM baseline model and generates a state at every step that is used in multiple tasks: (i) action prediction to perform one of the four possible actions at each step: go forward (0.25m), turn left (10 degrees), turn right (10 degrees), and stop; (ii) semantic grounding via current room detection (*SG_CRD*); and (iii) semantic grounding via post-navigation grounding (*SG_PN*). The *SGN* takes the $RoomNav$ instruction (which is fixed throughout the task) and the visual representation (which changes at each step) as input. In specific, SGN predicts the current room for SG_CRD using a linear layer on top of each SGN hidden state and predicts post-navigation answer on the last SGN hidden state. For action prediction at each timestep, the SGN hidden state attends to each hidden state of the generic room sequence in CP_RS to obtain a contextual representation. The attention-based representation is then concatenated with the original SGN hidden state to predict the next action. Please refer to Figure 6 in the Appendix for more details.

Two-stage SGN for Self-supervised Learning with Room Extraction (RE) Module In an unseen environment, we aim to build an agent which can update its action prediction by aligning what it has learned already to the unique patterns and semantics observed in the new setting. Humans are capable of doing this because they have semantic understanding: humans fine-tune their action prediction in an unseen environment according to newer semantic observations. Since there is no semantic annotation for unseen environments, we need to simulate the annotations as ground-truth to fine-tune the SGN module in a self-supervised setting. To get extra pseudo labels, we introduce the Room Extraction Module (RE).

The RE module detects the current room and the rooms on the left, right, and front given the panoramic representation, which facilitates the agent to understand semantics from different angles. We implement the RE module with a multi-layer perceptron (MLP) on the image representation and add different heads for different room extraction. Note that the major difference between RE and SG_CRD is the input to the modules: RE takes image features independently from the agent while SG_CRD takes hidden states from SGN as input for room detection. Specifically, RE is a separate model to extract semantics, agnostic to instructions and trajectories. In comparison, SG_CRD encodes the instruction and the trajectory history. More importantly, SG_CRD shares parameters with SGN which predicts actions. Therefore, we take RE predictions as pseudo ground-truth and fine-tune SG_CRD on top of SGN so that we can achieve the goal of fine-tuning action prediction.

Because we need independent features to predict the same objective from RE and SG_CRD, we design a two-stage training process over the SGN at each time step to perform grounding along with navigation. In the first stage, the Current Room Detection task (SG_CRD) on top of SGN is performed without information flowing from RE representations by masking. The reason for masking RE representation is that RE hidden states are optimized for room extractions in different angles with its training objective, which already contains room detection features. Without RE hidden state, SGN is encouraged to capture semantics for SGN_CRD to detect room information independently using raw scene information and previous SGN hidden states. On the other hand, if RE outputs are considered as features, SG_CRD may simply copy the representations without utilizing the learned semantics. Similarly, Post Navigation Grounding task (SG_PN) on top of SGN is performed only at the last state when "stop" action is received in the first stage. In the second stage, we feed output representations from CP and RE modules (depicted via dotted lines in Figure 2) into SGN to perform action prediction. The reason to incorporate RE representation, which extracts features directly from image input, is that abstracted semantics are shown to help with navigation as seen in previous research (Mousavian et al. 2018; Hudson and Manning 2019).

With the two-stage training objectives, we can perform self-supervised learning (SIL) on unseen environments to update the agent for better semantic understanding. In specific, we take the prediction from RE as ground truth labels and fine-tune the SG_CRD and SG_PN heads together with the SGN for action prediction. The agent explores the environment according to the trained SGN for a pre-defined t steps to get familiarized with the new environment, calculates losses between the two room detection models, and finally updates the parameters for the LSTM in SGN. The agent then navigates towards the target room from the starting location using the fine-tuned parameters.

Learning Procedure

We train the agent in two ways: (i) imitation learning (IL) with shortest path trajectories available during training, and (ii) self-supervised imitation learning (SIL) on unseen environments, inspired by the work from (Wang et al. 2018). During IL, apart from the main action prediction task, we perform five auxiliary tasks: 1. next room detection (CP_Nxt) 2. target to source room sequence prediction (CP_RS) 3. current and surrounding rooms extraction (RE) 4. post navigation response generation (SG_PN) and 5. current room predictions on top of SGN for the first stage (SG_CRD). In total, we have six losses during imitation learning including action prediction. The overall loss func-

Module	succ. rate	easy succ. rate	med. succ. rate	hard succ. rate
Baseline	0.25	0.41	0.24	0.17
+ Common Sense (CS)	0.30	0.44	0.29	0.25
+ Semantic Grounding (SG)	0.28	0.53	0.24	0.19
+ SG + SIL	0.36	0.56	**0.40**	0.21
+ CP + SG + SIL	**0.39**	0.56	**0.40**	**0.28**

Table 1: Results on Imitation Learning (IL) and Self-supervised IL (SIL) for easy, medium, and hard trajectories on unseen test environments. For Common Sense Planning (CP), CP_Nxt represents next room prediction while CP_RS utilizes room sequence. For Semantic Ground (SG), room extraction (RE) identifies current and nearby rooms on input images, while SG_CRD and SG_PN performs current room detection at each timestep for local semantics and post navigation for global semantics, respectively, on hidden states.

tion is:

$$\begin{aligned} L_{IL} = \lambda\_a * L\_action + \lambda_{CP\_Nxt} * L_{CP\_Nxt} \\ + \lambda_{CP\_RS} * L_{CP\_RS} + \lambda_{RE} * L_{RE} + \lambda_{SG\_PN} \\ * L_{SG\_PN} + \lambda_{SGN\_CRD} * L_{SGN\_CRD} \end{aligned} \tag{1}$$

where the loss for each task is the cross-entropy loss between the prediction and the annotations in the environment on either the last state (for SG_PN only) or for each state in the SGN (for other modules).

For SIL on unseen environments, we obtain losses from SG_CRD using RE predictions as target labels at each time step and from SG_PN at the end of the exploration. The loss function is represented as:

$$\begin{aligned} L_{SIL} = \lambda_{SG\_CRD} * L^{'}_{SG\_CRD} + \\ \lambda_{SG\_PN} * L^{'}_{SG\_PN} \end{aligned} \tag{2}$$

where $L^{'}_{SG\_CRD}$ and $L^{'}_{SG\_PN}$ indicates the loss using simulated labels on unseen environments, in comparison with L_{SG_CRD} and L_{SG_PN} using true ground-truth labels on annotated training environments.

Experiments

Data and Environment: We use Habitat Simulator and corresponding APIs (Manolis Savva et al. 2019) to render the MatterPort3D environment for all our tasks. One of the key tasks in MatterPort3D dataset is point navigation, wherein an agent needs to navigate from a source coordinate to a target coordinate. We adapt this task to form a $RoomNav$ task by replacing the target coordinates with the corresponding 27 room types annotated in the dataset (excluding "other room"). We remove those trajectories where the target and the source rooms are the same and the ones where the target is at the border of several rooms. There are in total 53 houses and 5020 trajectories in training, 11 houses and 168 trajectories for validation , and 15 houses and 324 trajectories for testing. To measure the complexity of each trajectory, we use the same measure as the point navigation task, which is the ratio of geodisic distance to that of the euclidean distance, where higher ratio indicates harder tasks. The average number of rooms between the source and target is 2.41, 3.01, and 4.06, in easy, medium, and hard trajectories in the training data respectively.

Model Input: The SGN has two types of input: (i) Environment input including task specific instruction (e.g. "go to the kitchen") and RGB values of visual observations in each state, and (ii) semantic information such as room annotations for training RE and sampled questions for PN. Semantic information is used in semantic predictions and question generation during training only, because such information is not available on unseen environments. Following previous embodied navigation work (Fried et al. 2018; Wang et al. 2018), we extract panoramic image features using a fixed pretrained ResNet-152 (He et al. 2015). Specifically, we turn the agent 90 degrees to the left and right to obtain a 270-degree view at each timestep. We extract and concatenate features in the left, front, and right images and then pass through a single feed forward layer to obtain the environment visual representation. In order to evaluate the information gained from semantic understanding instead of memorizing segments or detecting obstructions, we only use RGB features in our model instead of features from other sensors, such as semantic masking features (Wu et al. 2018a) or depth features.

Hyperparameter tuning: We use the validation set to tune the hyperparameters including the weights in each of the tasks in equation 1. In specific, we set the weight of action prediction loss to 1 and do grid search for other weights.

Evaluation Metrics: We use three evaluation metrics: success rate, success per length (SPL) following Wang et al. (2018), and non-stop SPL. Success rate is defined as the percentage of trajectories where the agent enters the target room. Success per length (*SPL*) is defined as the success rate normalized by the shortest path. In particular, *SPL* considers a game successful only if the agent predicts the "stop" action inside the target room, which is infrequently seen (about once every 71 steps) compared to other actions during training. We use *non-stop SPL* to relax this constraint to count the percentage of trajectories in which agent enters the target room during the trajectory. We note that non-stop SPL is a relatively weak metric, but we include this less sensitive metric against the "stop" action to indicate how well the agent can navigate to the target room. In other words, non-stop SPL can indicate the agent's performance on path planning. We also report average steps, which directly determines SPL, to indicate the number of steps the agent explores before predicting "stop" (with the maximum number of steps set to 200, and the average number of steps in the annotated trajectories for training is 82).

Results

We first analyze results for imitation learning and self-supervised imitation learning. Then we interpret why the agent benefits from the proposed model by interpreting the learned embedding alignments.

Imitation learning

We observe in Table 1 that common sense planning and semantic understanding help in the imitation learning setting across the board when compared to the $LSTM$ baseline model that does not incorporate these modules. Note that common sense and semantic information is not fed as features to the agent, rather is learnt via auxiliary tasks.

Common Sense Planning: We incorporate common sense via two sub-tasks: (i) next room guidance (*CP_Nxt*) and (ii) generic room sequence from target to source room (*CP_RS*) as described in section . Results show that *CP* modules improve navigation performances in medium and harder trajectories more than the easy trajectories and hence indicates that they help with long-term planning. Next room prediction alone leads to significant improvement in SPL (80% improvement over the baseline) and the second best success rate in hard tasks (40% improvement over baseline). When we combine with room sequence module, the agent's performance improves in both easy and medium trajectories, but not in hard trajectories. This suggests that room sequence module learns generic patterns, but for hard trajectories where geodesic distance is significantly higher than the euclidean distance, CP_RS does not help much probably due to incorrect long sequence predictions.

Semantic Understanding is incorporated via three different tasks: (i) a separate room identification model which predicts nearby rooms (*RE*) given current views using the shared $ResNet$ input. (ii) current room prediction given the SGN hidden state (SGN_CRD) (iii) post-navigation grounding with sampled question (SGN_PN). Results suggest that incorporating semantic understanding generally improves short-term planning compared to an agent without semantic understanding (baseline). It is observed that the agent fed with RE and SG_CRD tends to achieve higher SPL scores because it usually stops early with less average number of steps to complete the task. At the same time, the early stopping can also explain the low performances on medium and hard trajectories. *SG_PN* does not follow a similar pattern because grounding in this case is performed at the terminal state, hence it does not impact turn-level action prediction directly, leading to longer trajectories. Moreover, SG_PN does better on medium and hard trajectories because it lets the navigator (*SGN*) focus more on intermediate action predictions while ensuring semantic understanding at the terminal state.

Combing the two room detection objectives together ($RE + SG\_CRD$), we get the second best *SPL* score (0.141, 110% better than the baseline) but improvements are mostly on easy trajectories. This indicates that the agent might tend to focus more on the auxiliary task than on the original action prediction task. However, adding SG_PN to step level RE and SG_CRD modules to facilitate semantic understanding from a global perspective leads to significant increase in performance for medium and hard trajectories, while maintaining high SPL scores.

Discussion: Note that the average number of steps is higher than that in the annotated trajectory (82). From our qualitative analysis by evaluating the generated videos along the testing trajectories, the main reason for higher average steps is that the agent can get stuck in front of an object such as a table (by predicting turning and going forward consistently). This indicates that the agent does not achieve the goal by chance roaming around the environment. In addition, different performances in different metrics such as success rate in multiple difficulty levels suggest that our proposed modules are complementary to each other since they are helping navigation in different perspectives.

Self-supervised Imitation Learning

To perform *SIL*, we either use current room prediction (*SG_CRD*) or post navigation grounding (*SG_PN*) or both as the auxiliary task to fine-tune the Semantically Grounded Navigator (*SGN*) to get a loss function against the simulated target label using RE. While performing SIL, we found that by letting the agent explore unseen environments for 20 steps ($t = 20$) before actually executing the instruction and navigating to the target significantly improves the performance (7% with *RE + SG_CRD*, 22% with *RE + SG_PN* and *RE + SG_CRD + SG_PN*). Similar to the pattern observed without *SIL*, using local semantic grounding (SG_CRD) performs better on easier trajectories while using global semantic grounding by post-navigation questions (SG_PN) achieves better performance on harder trajectories. Finally, when we combine all the proposed CP and SG modules and perform SIL, we get the best performance overall with 56% improvement over non-stop SPL and 64% improvement over SPL, with maximum improvements on medium and hard trajectories. Note that the low absolute scores on SPL and non-stop SPL indicates that room navigation with low-level instructions on unseen environment for generalization is a hard task. We also observe that when we further fine-tune the agent in the SIL setting with more steps (with $t = 40, 60$), the performance degrades drastically as the model tends to overfit to noises of the approximate pseudo labels obtained from the *RE* model.

Cross-modal Embeddings

To identify why CP and semantic understanding helps in the navigation task, we analyzed the cross-modal embeddings learnt during training to show how the agent interpret language instructions. Traditional embeddings such as GloVe (Pennington, Socher, and Manning 2014), are functions of words or semantic entities appearing in similar contexts and

Figure 3: Top view of a house layout with dark areas as obstacles

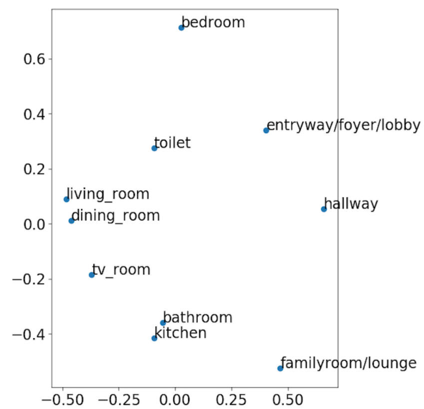

Figure 4: Embeddings of *RE + SG_CRD* model trained on all the training environments mapped to 2D space with PCA

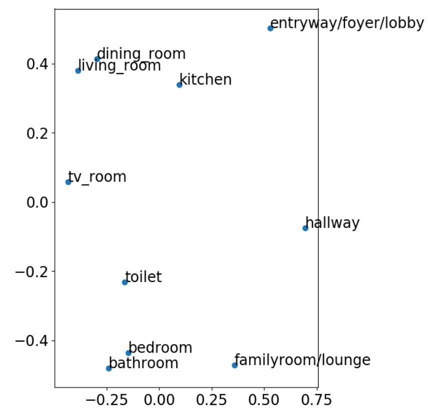

Figure 5: Embeddings of *RE + SG_CRD* model after SIL fine-tuning on the environment in Figure 3 mapped to 2D space.

may not capture the visual and structural properties of entities in realistic 3D worlds. Therefore, we first randomly initialize the cross-modal embeddings and then train them with the proposed modules across multiple trajectories in different environments. We qualitatively analyze these embeddings to explore if they reflect any structural or visual characteristics of the environment. Figure 3 represents the top-view of an environment and Figure 4 visualizes the embeddings trained using the *RE + SG_CRD* model by dimension reduction using PCA (while other models illustate similar patterns). The figure shows that the learned embeddings capture the average structural pattern of rooms across all the environments. We further fine-tune the agent and the embeddings on the example environment shown in Figure 3 using self-supervised imitation learning and visualize it in Figure 5. We observe that the fine-tuned embeddings tend to mimic the structural and positional patterns of the exact environment. This indicates that the proposed models help the agent to understand the instructions better by aligning the instruction encoding with the actual scene information. We conjecture that such alignment, which is learned from the proposed common sense and semantic grounding modules, explains what the model actually learns. The close mapping between the fine-tuned word embeddings of the room types and the structure of the environment draws connections to the SLAM (Durrant-Whyte and Bailey 2006) algorithm, which is one of the most popular mapping algorithms for navigation. However, we can leverage what the agent has already learned as a prior instead of exhaustively exploring each room for SLAM. This alignment also draws connection to recent research on vision-and-language pre-training such as VisualBERT (Li et al. 2019) which is optimized to align text and image regions with self-attention. We leave the detailed comparison to future work.

Conclusion

Humans navigate to rooms on unseen environments leveraging common sense of room layout and semantic understanding of the environment. We propose to simulate human navigation by incorporating these features ignored in previous research. The goal of this paper is not to build a state-of-the-art navigation system, but using the navigation environment to explore if common sense and semantic grounding is useful in visual navigation. We introduced methods to incorporate these features and showed that common sense and semantic grounding help in long-term and short-term planning respectively for effective navigation. We also found out that the agent fine-tuned using self-supervised imitation learning generalizes better to unseen environments. Furthermore, we analyzed the reason for such improvement by inspecting cross-modal embeddings obtained during training, which captures structural and positional patterns of the environment. This suggests that the agent learns a semantic map of the environment in the process of the navigation.

References

Anand, A.; Belilovsky, E.; Kastner, K.; Larochelle, H.; and Courville, A. C. 2018. Blindfold Baselines for Embodied QA. *CoRR* abs/1811.05013. URL http://arxiv.org/abs/1811.05013.

Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I. D.; Gould, S.; and van den Hengel, A. 2017. Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments. *CoRR* abs/1711.07280. URL http://arxiv.org/abs/1711.07280.

Das, A.; Datta, S.; Gkioxari, G.; Lee, S.; Parikh, D.; and Batra, D. 2017. Embodied Question Answering. *CoRR* abs/1711.11543. URL http://arxiv.org/abs/1711.11543.

Das, A.; Gkioxari, G.; Lee, S.; Parikh, D.; and Batra, D. 2018. Neural Modular Control for Embodied Question Answering. *CoRR* abs/1810.11181. URL http://arxiv.org/abs/1810.11181.

Durrant-Whyte, H. F.; and Bailey, T. 2006. Simultaneous localization and mapping: part I. *IEEE Robot. Automat. Mag.* 13(2): 99–110. doi:10.1109/MRA.2006.1638022. URL https://doi.org/10.1109/MRA.2006.1638022.

Fried, D.; Hu, R.; Cirik, V.; Rohrbach, A.; Andreas, J.; Morency, L.; Berg-Kirkpatrick, T.; Saenko, K.; Klein, D.; and Darrell, T. 2018. Speaker-Follower Models for Vision-and-Language Navigation. *CoRR* abs/1806.02724. URL http://arxiv.org/abs/1806.02724.

Gordon, D.; Kembhavi, A.; Rastegari, M.; Redmon, J.; Fox, D.; and Farhadi, A. 2017. IQA: Visual Question Answering in Interactive Environments. *CoRR* abs/1712.03316. URL http://arxiv.org/abs/1712.03316.

Gupta, S.; Davidson, J.; Levine, S.; Sukthankar, R.; and Malik, J. 2017. Cognitive Mapping and Planning for Visual Navigation. *CoRR* abs/1702.03920. URL http://arxiv.org/abs/1702.03920.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385. URL http://arxiv.org/abs/1512.03385.

Heess, N.; Wayne, G.; Silver, D.; Lillicrap, T. P.; Tassa, Y.; and Erez, T. 2015. Learning Continuous Control Policies by Stochastic Value Gradients. *CoRR* abs/1510.09142. URL http://arxiv.org/abs/1510.09142.

Hermann, K. M.; Hill, F.; Green, S.; Wang, F.; Faulkner, R.; Soyer, H.; Szepesvari, D.; Czarnecki, W. M.; Jaderberg, M.; Teplyashin, D.; Wainwright, M.; Apps, C.; Hassabis, D.; and Blunsom, P. 2017. Grounded Language Learning in a Simulated 3D World. *CoRR* abs/1706.06551. URL http://arxiv.org/abs/1706.06551.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Hu, R.; Fried, D.; Rohrbach, A.; Klein, D.; Darrell, T.; and Saenko, K. 2019. Are You Looking? Grounding to Multiple Modalities in Vision-and-Language Navigation. *CoRR* abs/1906.00347. URL http://arxiv.org/abs/1906.00347.

Hudson, D. A.; and Manning, C. D. 2019. Learning by Abstraction: The Neural State Machine. *CoRR* abs/1907.03950. URL http://arxiv.org/abs/1907.03950.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

Li, L. H.; Yatskar, M.; Yin, D.; Hsieh, C.-J.; and Chang, K.-W. 2019. VisualBERT: A Simple and Performant Baseline for Vision and Language.

Manolis Savva; Abhishek Kadian; Oleksandr Maksymets; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; Parikh, D.; and Batra, D. 2019. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Mirowski, P.; Grimes, M. K.; Malinowski, M.; Hermann, K. M.; Anderson, K.; Teplyashin, D.; Simonyan, K.; Kavukcuoglu, K.; Zisserman, A.; and Hadsell, R. 2018. Learning to Navigate in Cities Without a Map. *CoRR* abs/1804.00168. URL http://arxiv.org/abs/1804.00168.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T. P.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous Methods for Deep Reinforcement Learning. *CoRR* abs/1602.01783. URL http://arxiv.org/abs/1602.01783.

Mousavian, A.; Toshev, A.; Fiser, M.; Kosecka, J.; and Davidson, J. 2018. Visual Representations for Semantic Target Driven Navigation. *CoRR* abs/1805.06066. URL http://arxiv.org/abs/1805.06066.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1532–1543. URL https://www.aclweb.org/anthology/D14-1162/.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms. *CoRR* abs/1707.06347. URL http://arxiv.org/abs/1707.06347.

Shridhar, M.; Thomason, J.; Gordon, D.; Bisk, Y.; Han, W.; Mottaghi, R.; Zettlemoyer, L.; and Fox, D. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. URL https://arxiv.org/abs/1912.01734.

Tangiuchi, T.; Mochihashi, D.; Nagai, T.; Uchida, S.; Inoue, N.; Kobayashi, I.; Nakamura, T.; Hagiwara, Y.; Iwahashi, N.; and Inamura, T. 2019. Survey on frontiers of language and robotics. *Advanced Robotics* 1–31. doi:10.1080/01691864.2019.1632223.

Wang, X.; Huang, Q.; Çelikyilmaz, A.; Gao, J.; Shen, D.; Wang, Y.; Wang, W. Y.; and Zhang, L. 2018. Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation. *CoRR* abs/1811.10092. URL http://arxiv.org/abs/1811.10092.

Wu, Y.; Wu, Y.; Gkioxari, G.; and Tian, Y. 2018a. Building generalizable agents with a realistic and rich 3D environment. *arXiv preprint arXiv:1801.02209* .

Wu, Y.; Wu, Y.; Gkioxari, G.; and Tian, Y. 2018b. House3D: A Rich and Realistic 3D Environment. *arXiv preprint arXiv:1801.02209* .

Wu, Y.; Wu, Y.; Tamar, A.; Russell, S.; Gkioxari, G.; and Tian, Y. 2019. Bayesian Relational Memory for Semantic Visual Navigation.

Yang, W.; Wang, X.; Farhadi, A.; Gupta, A.; and Mottaghi, R. 2019. Visual Semantic Navigation using Scene Priors. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. URL https://openreview.net/forum?id=HJeRkh05Km.

674

Appendix

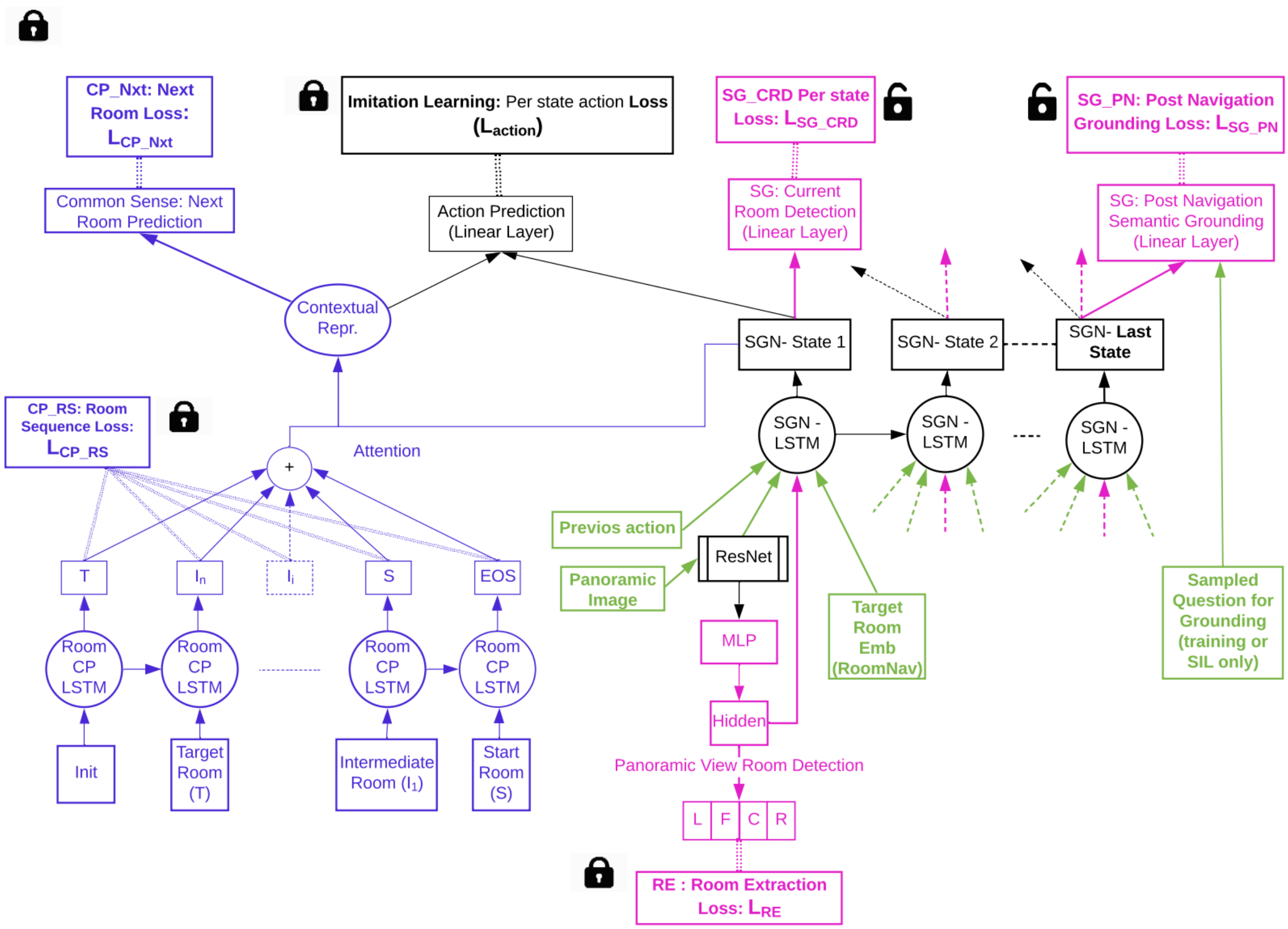

Figure 6: Detailed architecture for Common Sense and Semantically Grounded Navigation model. Black components correspond to the baseline navigator model. Purple components are introduced to incorporate common sense while pink components are for Semantic Understanding. Semantically Grounded Navigator (SGN) is designed to perform semantic understanding for action prediction, while common sense is fed as guidance for better planning. There are six losses, four of them are locked during inference, except L_{SG_CRD} and L_{SG_PN} are unlocked for self-supervision.