FollowNet: Robot Navigation by Following Natural Language Directions with Deep Reinforcement Learning

Pararth Shah[1] Marek Fiser[1] Aleksandra Faust[1] J. Chase Kew[1] Dilek Hakkani-Tur[1]

Abstract— **Understanding and following directions provided by humans can enable robots to navigate effectively in unknown situations. We present FollowNet, an end-to-end differentiable neural architecture for learning multi-modal navigation policies. FollowNet maps natural language instructions as well as visual and depth inputs to locomotion primitives. FollowNet processes instructions using an attention mechanism conditioned on its visual and depth input to focus on the relevant parts of the command while performing the navigation task. Deep reinforcement learning (RL) a sparse reward learns simultaneously the state representation, the attention function, and control policies. We evaluate our agent on a dataset of complex natural language directions that guide the agent through a rich and realistic dataset of simulated homes. We show that the FollowNet agent learns to execute previously unseen instructions described with a similar vocabulary, and successfully navigates along paths not encountered during training. The agent shows 30% improvement over a baseline model without the attention mechanism, with 52% success rate at novel instructions.**

I. Introduction

Humans often navigate unknown environments by observing their surroundings and following directions. These directions consist predominantly of landmarks and directional instructions and other common words. For example, humans can find a kitchen in a home they haven't visited before, by following directions such as: "Turn right at the dining table, then take the second left". This process requires visual observations, e.g. a dining table in the field of view or knowledge of a typical hallway, and execute actions present in the direction: turn left. There are multiple dimensions of complexity: limited field of view, qualifier words like "second", synonyms such as "taking" and "turning", understanding that "take the second left" refers to the door, etc.

In this paper, we apply human-like direction following to robots navigating in 2-dimensional workspaces (Fig. 1). We present robots with example directions similar to the one above, and train a deep reinforcement learning (DRL) agent to follow the directions. The agent is tested on how well it follows new directions when starting from different locations. We accomplish this with a novel deep neural net architecture, FollowNet (Fig. 2), which is trained with Deep Q-Network (DQN) [17]. The observation space consists of natural language instructions and visual and depth observations from the robot's vantage point (Fig. 4b). The policy's output is the next motion primitive to perform. The robot moves along an obstacle-free grid, but the instructions require the robot to move over a variable number of nodes to reach the destination. The instructions we use (Table I) contain implicitly encoded rooms, landmarks, and motion primitives. In the example above, "kitchen" is the room that serves as the goal location. "Dining table" is an example of a landmark, a point at which the agent might change direction. Both rooms and landmarks are mapped to groups of grid points without the agent's knowledge. We use a sparse reward, given to the agent only when it reaches a waypoint.

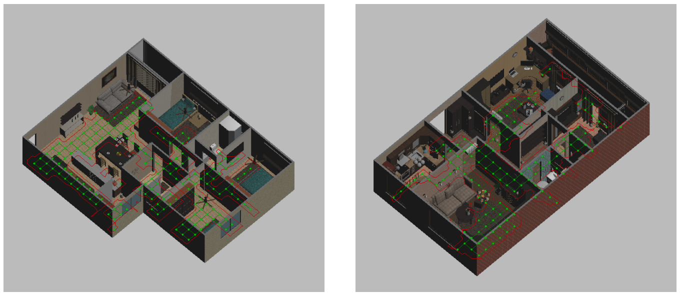

(a) House 1 (b) House 2

Fig. 1. 3-dimensional rendering of the houses used for learning navigation from natural language instructions.

The novel aspect of the FollowNet architecture is a language instruction attention mechanism that is conditioned on the agent's sensory observations. This allows the agent to do two things. First, it keeps track of the instruction command and focuses on different parts as it explores the environment. Second, it associates motion primitives, sensory observations, and sections of the instruction with the reward received, which enables the agent to generalize to new instructions.

We evaluate how well the agent generalizes to new instructions and new motion plans. First, we evaluate the agent on how well it follows previously unseen two-step directions in houses with which it is familiar. The results show that the agent follows 52% directions completely and 61% partially, a 30% increase over a baseline. Second, the same instructions are valid for a set of different starting positions. For example, ”Exit the room” is valid for any start location inside the room, yet the motion plan that the robot needs to execute to complete the task can be very different. To access how well the motion plans generalize to new start locations, we evaluate the agent on the instructions on which it was trained (up to five-step directions), but from new starting positions. The agent completes 70% directions partially and 54% fully. To put that in perspective, multi step directions are challenging for people to perform as well.

II. Related Work

End-to-end navigation methods [21], [6], [30] use deep reinforcement learning on robots' sensory observations and

[1] The authors are with Google, Mountain View, CA, USA {pararth,mfiser,faust,jkew,dilekh}@google.com

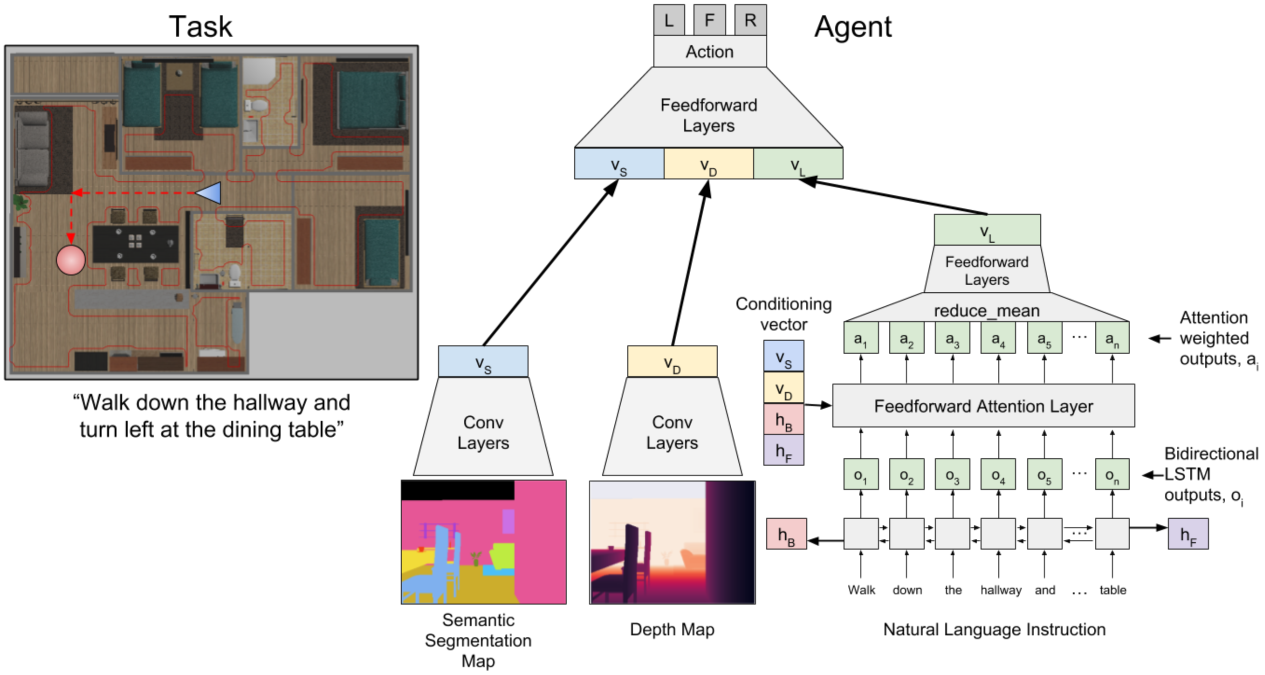

Task
"Walk down the hallway and turn left at the dining table"
Agent
L
F
R
Action
Feedforward Layers
v_S
v_D
v_L
Feedforward Layers
reduce_mean
a_1 a_2 a_3 a_4 a_5 ... a_n
Attention weighted outputs, a_i
Conditioning vector
v_S
v_D
h_B
h_F
Feedforward Attention Layer
o_1 o_2 o_3 o_4 o_5 ... o_n
Bidirectional LSTM outputs, o_i
h_B
h_F
Walk down the hallway and ... table
Conv Layers
Conv Layers
Semantic Segmentation Map
Depth Map
Natural Language Instruction

Fig. 2. Neural model for mapping visual and language inputs to a navigation action. Left: An example task, where the robot starts at the position and orientation specified by the blue triangle, and must reach the goal location specified by the red circle. The robot receives a natural language instruction to follow the path marked in red, listed below the image. Right: the FollowNet architecture. Semantic segmentation map is fed into a 3-layer convolutional net with 3, 8, and 16 outputs, [1,1], [4,4], [3,3] kernels, and 1, 2, 1 strides. The depth image is an input to 2-later convolutional network with 8 and 16 outputs, [4, 4] and [3, 3] kernels, and 2 and 1 strides. The command is an input to a bidirectional GRU with 32 outputs. The Feedforward Attention Layer has soft attention 16 hidden states. Lastly the the Feedforward layers consist of two layers with 16 and 8 hidden units.

relative goal location. In this work, we provide natural language instructions instead of the explicit goal, and the agent must learn to interpret the instructions to complete the task. One challenge in reinforcement learning applied to robotics is the state space representation. Large state spaces slow down the learning, so often different approximation techniques are used. Examples of these are probabilistic roadmaps (PRMs) [6], [11] and simple discretization of the space [19], [18]. Here, we discretize the 2-dimensional workspace and allow the agent to move through the grid from node to node. Essentially, we assume that the robot can avoid obstacles and move safely between two grid points by executing the motion primitive corresponding to the action.

Deep learning has shown great success with learning natural language [14], [15] and vision [10], [8] and even combining visual and language learning [24], [26]. Applied to robot motion planning and navigation, language learning typically requires some level of parsing with formal descriptions [12], semantic parsing [27], a probabilistic graphical model [22], encoding and alignment [13], or task grounded language [3] etc. Learning object labels through natural language, though, has been addressed mainly by learning to parse natural language instructions into a hierarchical structure which can be used during planning and execution of robot actions [23], [28], [2] and active learning [25]. Here, similarly to [1], we aim to implicitly learn the labels for landmarks (objects) and motion primitives (actions) and their interpretation with respect to visual observations. Unlike [1], we use DQN [17] over the proposed FollowNet to learn the navigation policy. Other works [9] have used curriculum to complete several tasks in an environment.

Another recent work that combines 3D navigation, vision, and natural language is learning to answer questions [5]. The questions come from a prescribed set of questions where certain keywords are replaced. In our work, the language instructions given to the agent here are independently created by four people, and presented to the agent without any processing. Several methods learn from unfiltered language [16], [29] and visual input. In these methods that visual input is an image of an entire planning environment. In contrast, FollowNet only receives partial environment observation.

III. Methods

A. Problem formulation

We assume the robot to be a point-mass with three degrees of freedom (x, y, θ), navigating in a 2-dimensional grid overlaid on a 3-dimensional indoor house environment (Fig. 1). To train a DQN [17] agent, we formulate the task as a Partially Observable Markov Decision Process (POMDP): a tuple (O, A, D, R) with observations $\boldsymbol{o} = [\boldsymbol{o}_{NL}\ \boldsymbol{o}_{V}] \in O$, where $\boldsymbol{o}_{NL} = [w_1 w_2 \ldots w_n]$ is a natural language instruction sampled from a set of user-provided directions for reaching a goal. The location of the goal is unknown to the agent. $\boldsymbol{o}_V$ is the visual input available to the agent, which consists of the image that the robot sees (Fig. 4c) at a time-step i. The set of actions $A = \{turn\frac{\pi}{2},\ go\ straight,\ turn\frac{3\pi}{2}\}$ enables the robot to either turn in place or move forward by a step. The system dynamics $D : O \times A \rightarrow O$ are deterministic and apply the action to the robot. The robot either transitions to the next grid cell or changes its orientation. Note, that the agent does not know where it is located in the environment.

The reward $R : O \to \mathbb{R}$ rewards an agent reaching a landmark (waypoint) mentioned in the instruction, with a reward of $+1.0$ if the waypoint is the final goal location, and a smaller reward of $+0.05$ for intermediate waypoints. The agent is rewarded only once for each waypoint in the instruction it reaches, and the episode terminates when the agent reaches the final waypoint, or after a maximum number of steps. Our aim is to learn an action-value function $Q : O \to \mathbb{R}^{\|A\|}$, approximated with a deep neural network and trained with DQN.

Fig. 2 provides an example task, where the robot starts at the position and orientation specified by the blue triangle, and must reach the goal location specified by the red circle. The robot receives a natural language instruction (Table I) to follow the path marked in red.

B. FollowNet

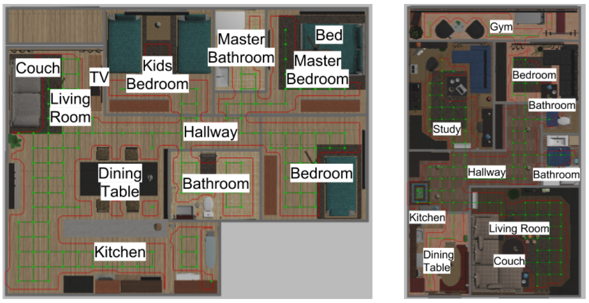
We present FollowNet, a neural architecture for approximating the action value function directly from the language and visual inputs (Fig. 2). To simplify the image processing task, we assume a separate preprocessing step parses the visual input $\boldsymbol{o}_V \in \mathbb{R}^{n \times m}$ to obtain a semantic segmentation $\boldsymbol{o}_S$ which assigns a one-hot semantic class id to each pixel, and a depth map $\boldsymbol{o}_D$ which assigns a real number to each pixel corresponding to the distance from the robot. The agent takes the ground truth $\boldsymbol{o}_S$ and $\boldsymbol{o}_D$ from its current point of view and runs each through a stack of convolutional layers followed by a fully-connected layer. From these it obtains fixed length embedding vectors $v_S \in \mathbb{R}^{d_S}$ and $v_D \in \mathbb{R}^{d_D}$ (where $d_X = length(v_X)$) that encode the visual information available to the agent.

We use a single layer bi-directional GRU network [4] with state size d_L and initial state set to 0, to encode the natural language instruction using the following equations:

$$h_F, \{o_{i,F}\} = GRU_F(\{w_i\})$$
$$h_B, \{o_{i,B}\} = GRU_B(\{w_i\})$$
$$o_i = [o_{i,F}\ o_{i,B}]$$

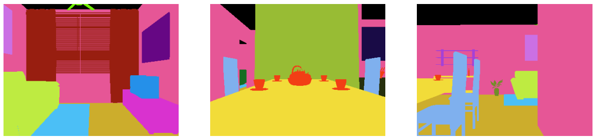
where $h_F, h_B \in \mathbb{R}^{d_L}$ are the final hidden states of the forward and backward GRU cells, respectively, while $o_i \in \mathbb{R}^{2d_L}$ are the concatenated outputs of the forward and backward cells, corresponding to the embedded representation of each token conditioned on the entire utterance. To enable the agent to focus on different parts of the instruction depending on the context, we add a feed-forward attention layer over o_i:

$$v_C = [v_S\ v_D\ h_B\ h_F]$$
$$e_i = FF_A(v_C, o_i)$$
$$\alpha_i = softmax(e_i)$$
$$a_i = \alpha_i\ o_i$$
$$v_A = (1/k)\Sigma_i^k a_i$$
$$v_L = FF_L(v_A)$$
$$Q(\boldsymbol{o}) = FF_Q([v_S, v_D, v_L])$$



(a) House 1 (b) House 2

Fig. 3. Landmarks and grid overlaid over the environments.



(a) Couch in living room (b) Table in front (c) Table to left, couch to the right

Fig. 4. Semantic segmentation map observations for FollowNet agents. Colors correspond to object types (unknown to the agent), and are consistent between the houses and vantage points. Couch is green (a and c), dinning table is yellow (b and c).

We use a feed-forward attention layer FF_A conditioned on v_C, which is the concatenated embeddings of the visual and language inputs, to obtain unnormalized scores e_i for each token w_i. e_i are normalized using the softmax function to obtain the attention scores α_i, which correspond to the relative importance of each token of the instruction for the current time step. We take the attention-weighted mean of the output vectors o_i, and pass it through another feed-forward layer to obtain $v_L \in \mathbb{R}^{d_L}$, which is the final encoding of the natural language instruction.

The Q function is then estimated from the concatenated $[v_S, v_D, v_L]$ passed through a final feed-forward layer. During training, we sample actions from the Q-function using an epsilon-greedy policy to collect experience, and update the Q-network to minimize the Bellman error over batches of transitions using gradient descent. After the Q function is trained, we used the greedy policy $\pi(\boldsymbol{o}) : O \to A$, with respect to learned $\hat{Q}$, $\pi(\boldsymbol{o}) = \pi^{\hat{Q}}(\boldsymbol{o}) = \text{argmax}_{a \in A}\, \hat{Q}(\boldsymbol{o}, a)$, to take the robot to the goal presented in the instruction $\boldsymbol{o}_l$.

IV. Results

In this Section we present the training and evaluation setup, and then evaluate FollowNet against a baseline model without attentional layer. We also look into the effect of the attentional layer and task complexity.

Setup and methodology: We chose two houses from the SUNCG [20] dataset that had many rooms and objects in common (Fig. 1). The size of the grid for House 1 is 23×18 nodes, for House 2 14×20 nodes.

For both houses we chose 7 navigation tasks consisting of starting and ending locations (e.g. "Study to table in the kitchen"). Three people each independently wrote one

TABLE I

Examples of instructions used in training. House # is identity of the house, start and goal determine the valid regions of the instructions. The robot only has access to the instruction and visual observation, without the context of where it is located (house #, start and goal) and the valid regions for the instruction. House #, start and goal are used for the RL reward design.

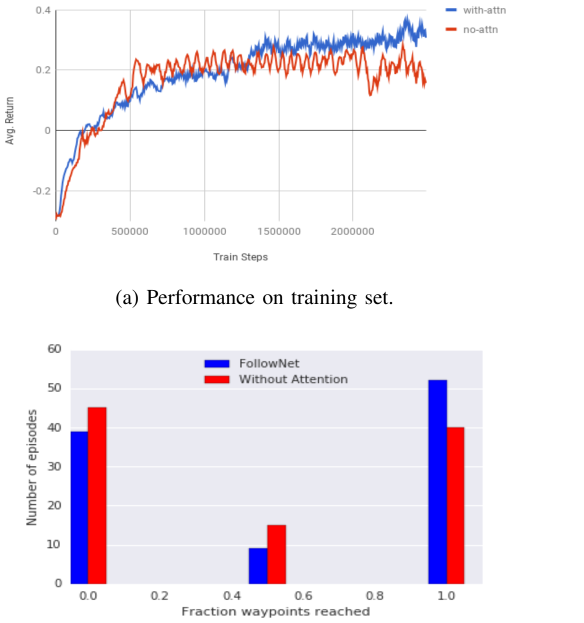
House #	Start	Goal	Instruction
1	Kids Bedroom	Bedroom	Exit the room, turn left and walk through the hall and enter the doorway in front of you
2	Study	Gym	Go out the door and turn left. Go forward until you reach a doorway, then turn left. Go forward and through the door in front of you. Go straight through the bedroom and through the door on the far side.
2	Gym	Kitchen	Go out the door, straight through the bedroom and out the door on the far side. Continue straight until you hit the wall, then turn right. After the corner of the wall, turn left and go through the door ahead.
2	Hallway	Gym	Go out through the bedroom, straight across and through the far door.
2	Dining Table	Bathroom	Go out the door and turn right after the corner of the wall. Go straight ahead and through the door.
2	Kitchen	Gym	Go out the door and straight across the hallway until you reach a doorway, then turn right. Go straight until you reach another doorway, then turn left. Go straight forward until you reach another doorway and go through that one. Go straight across the bedroom and through the door on the far side.

instruction for each task in each house forward, and one instruction for the same task reversed (e.g. "exit the room and take the door on the opposite wall"). After discarding some instructions for containing vocabulary not seen elsewhere, we settled on a set of 58 instructions. Examples of the tasks and instructions in the Table I. Each instruction contains implicitly stated waypoints. In the example above, the set of waypoints might be: study, table, kitchen, and door. The agent has no knowledge of the waypoints, and they are only used for reward computation and evaluation. We overlaid a navigation grid (1 meter edges) onto each house (Fig. 3). For each instruction, we associated grid nodes with valid starting points and waypoints that we expect agent to reach when completing the instruction. For each task, we selected approximately 5-10 starting nodes, where were randomly selected during the training.

For the evaluation, we followed the same methodology to create an additional dataset of 15 instructions which introduce new combinations of instructions and locations not present in training. For example: "Go out the door and straight across the hallway, then through the door in front of you." We made sure the evaluation instruction uses the same landmark and directional vocabulary as the training set without introducing new words. The evaluation instructions consists of two-step instructions, while the training set contains up to five-step instructions. The evaluation dataset consists of 100 queries, created over the 15 evaluation instructions and randomly sampled start and goal location within the start and goal area applicable to the instruction. For example, for Kids Bedroom anywhere in the room is the possible location to start an episode.

We trained FollowNet agent with the learning rate $\alpha = 1.70974 * 10^{-4}$ and discount factor $\gamma = 0.990022$, selected through a hyper-parameter tuning [7]. We stop the training after $2\,500\,000$ steps. We compare FollowNet to a baseline, which is an identical network but without the attention layers. The baseline consists of convolutional layers, RNN, and FCC layers. The baseline was trained and tuned in the same manner as the FollowNet. It is a non-trivial and challenging baseline.

Comparison to baseline with no attention: Fig. 5 shows the learning curve for the FollowNet and baseline over the holdout dataset. Early in the training the model with no attention is showing a slightly better learning curve. With prolonged training, FollowNet outperforms the baseline agent. This is because encoding the instruction with an RNN enables the agent to consider the relative ordering of words in the instruction. The visual inputs and language input are embedded separately and then fused into a single context vector which conditions the final action selection policy. Without attention on the RNN, the agent cannot selectively focus on parts of the instruction relevant to the visual context.



(a) Performance on training set.

(b) Performance on hold-out set.

Fig. 5. Comparing FollowNet (blue) with a baseline agent with no attention (red). Top: Average return on the training set plotted against no. of training steps. Evaluations done every 10,000 steps on the same hold out set. Bottom: Evaluated trained policies of both agents on a hold-out set for 100 episodes, and plotted a histogram of fraction of waypoints reached successfully. Learning to attend over the input instruction shows a 30% relative increase (52% vs. 40%) in fully successful episodes.

Fig. 5b depicts the histogram of fraction of waypoints reached successfully on the evaluation dataset. The FollowNet agent's overall success at following instructions is 52% on the evaluation dataset, while the baseline completes

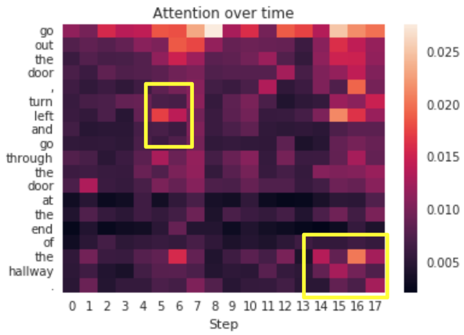


Fig. 6. Language attention heatmap over time steps. Brighter colors indicate more attention. Agent's attention over instruction words shifts based on the agent's location along the path to the goal.

only 40%. This means that the FollowNet has 30% relative increase over the baseline. We also see that FollowNet has fewer fully unsuccessful tasks (39% vs 45%), that is 13% relative decrease.

Attention Analysis: Fig. 6 shows a heatmap of the attention vector over a single episode. Along the y-axis is the tokenized instruction the agent was given. Each word and punctuation mark (commas and periods) is a separate token. Progressing right along the x-axis, at each timestep we see the weight the agent placed on each token, with lighter colors representing higher weights. In steps 5 and 6, the agent increases the attention on the word "left" as it takes a left in the environment. Towards the end of the episode, the agent attends to "the hallway" as it reaches the end of the task.

Fig. 7 depicts the fraction of successful (fully completed) tasks per word. We see that FollowNet is likely to complete tasks with orientation meanings (ahead, take, down, second, left, right, across). The agent without attention generally has the lower success rate across all words, and exhibits slightly different success probabilities across the words. Both agents have difficulty with words that do not appear often in the dataset (past, you), as expected.

Motion plan generalization on complex instructions: We look now at how well the FollowNet agent generalizes the motion plans to new start positions on a complex instruction data set used for the training, up to five-step instructions. Fig. 8 shows statistics relating to number of steps and number of waypoints in each episode. The training tasks are not trivial, with the number of steps needed to complete the task ranging from 7 to 29 steps, with a mean of 17.4. The evaluation tasks need between 3 and 26 steps, averaging 10.5 (Fig. 8a).

FollowNet agent's overall success at following instructions used in training is 54%, just 2% over the the evaluation dataset. This means that the agent generalizes pretty well to the new two-step directions. On the other hand, 30% of tasks make no progress. It is not surprising that the agent fails more often on the training instructions, because the instructions are more complex. When the agent does not complete the task even partially, it simply spins around without knowing what to do (Fig. 8b).

Fig. 8c breaks down the episodes by number of waypoints, a proxy for complexity of instructions. The agent is never fully successful at following four- or five-step directions, although in some cases it makes partial progress. Two- and three-step directions are often fully completed. On the evaluation dataset (Fig. 5b), which contains two-step tasks, we notice an interesting bimodal distribution: An agent which reaches the first waypoint is very likely to reach the second.

V. Conclusions

This paper presents the FollowNet architecture, which uses an attention mechanism over natural language instructions conditioned on multi-modal sensory observations as an action-value function approximator in DQN. The trained model learns to follow natural language instructions using only visual and depth information. The results show promise that we can simultaneously learn to generalize directional instructions and recognize landmarks. The agent is successful in following novel two-step directions most of the time (at the level of toddler), a 30% improvement over the baseline. In the future work, we aim to train the agent on a much larger dataset, do more in-depth analysis and empirical evaluation across several domains, and explore generalization across different environments.

ACKNOWLEDGMENT

The authors thank James Davidson for the helpful comments and discussions.

References

[1] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. D. Reid, S. Gould, and A. van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. *CoRR*, abs/1711.07280, 2017.

[2] D. Arumugam, S. Karamcheti, N. Gopalan, L. L. S. Wong, and S. Tellex. Accurately and efficiently interpreting human-robot instructions of varying granularities. In *Robotics: Science and Systems*, 2017.

[3] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. Salakhutdinov. Gated-attention architectures for task-oriented language grounding. In *AAAI*. AAAI Press, 2018.

[4] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[5] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied question answering. *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page to appear, 2018.

[6] A. Faust, O. Ramirez, M. Fiser, K. Oslund, A. Francis, J. Davidson, and L. Tapia. PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, page to appear, 2018.

[7] D. Golovin, B. Solnik, S. Moitra, G. Kochanski, J. E. Karro, and D. Sculley, editors. *Google Vizier: A Service for Black-Box Optimization*, 2017.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[9] K. M. Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. M. Czarnecki, M. Jaderberg, D. Teplyashin, M. Wainwright, C. Apps, D. Hassabis, and P. Blunsom. Grounded language learning in a simulated 3d world. *CoRR*, abs/1706.06551, 2017.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

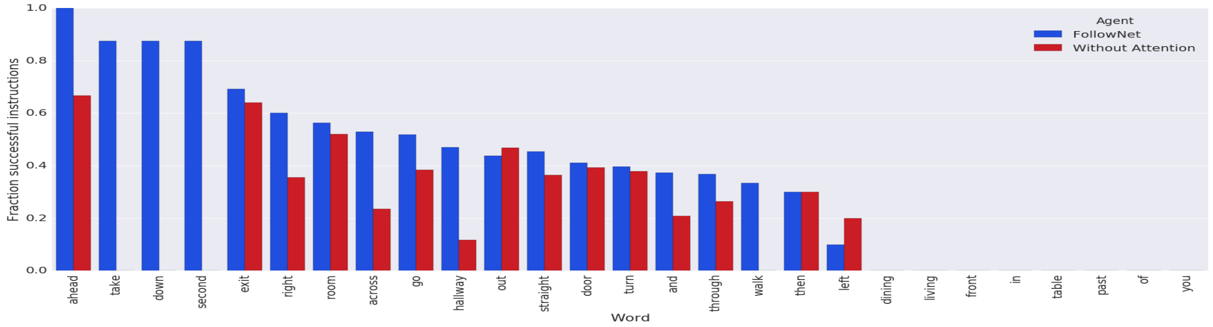
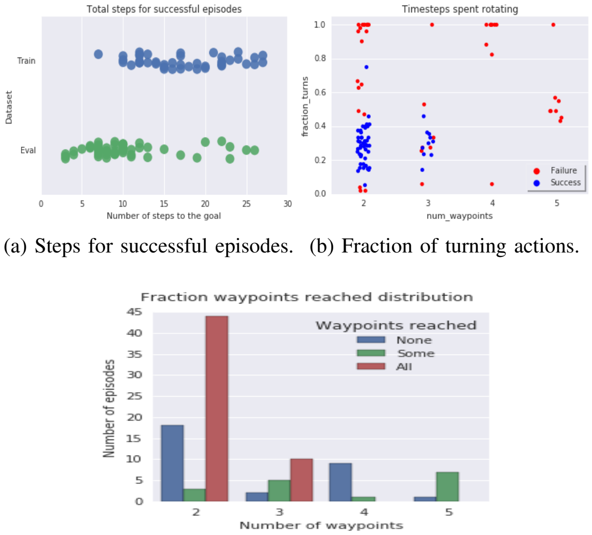


Fig. 7. Fraction of successful tasks per word occurrence on a holdout set.



(a) Steps for successful episodes. (b) Fraction of turning actions.

(c) Success rate vs. instruction complexity.

Fig. 8. Motion plan generalization on training instructions. (a) Number of steps per successful episode on the training (blue), and evaluation (green). (b) Fraction of actions that were a left or right turn in successful and unsuccessful episodes, split by number of waypoints in the instruction. (c) Fully (red), partially (green), and not (blue) completed tasks per number of waypoints on the training set.

[11] N. Malone, A. Faust, B. Rohrer, R. Lumia, J. Wood, and L. Tapia. Efficient motion-based task learning for a serial link manipulator. *Transactions on Control and Mechanical Systems*, 3(1), 2014.

[12] 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.

[13] H. Mei, M. Bansal, and M. R. Walter. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Proceedings of AAAI*, 2016.

[14] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. Hakkani-Tur, X. He, L. Heck, G. Tur, D. Yu, et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539, 2015.

[15] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[16] D. Misra, J. Langford, and Y. Artzi. Mapping instructions and visual observations to actions with reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1004–1015. Association for Computational Linguistics, 2017.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[18] A. Now, P. Vrancx, and Y.-M. Hauwere. *Game Theory and Multi-agent Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*. Springer Berlin Heidelberg, 2012.

[19] Z. Pei, S. Piao, and M. Souidi. Coalition formation for multi-agent pursuit based on neural network and AGRMF model. *CoRR*, abs/1707.05001, 2017.

[20] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[21] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel. Value iteration networks. In *IJCAI*, 2016.

[22] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1507–1514, San Francisco, CA, August 2011.

[23] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and N. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI*, volume 1, page 2, 2011.

[24] J. Thomason and R. J. Mooney. Multi-modal word synset induction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17)*, pages 4116–4122, Melbourne, Australia, 2017.

[25] J. Thomason, A. Padmakumar, J. Sinapov, J. Hart, P. Stone, and R. J. Mooney. Opportunistic active learning for grounding natural language descriptions. In S. Levine, V. Vanhoucke, and K. Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning (CoRL-17)*, pages 67–76, Mountain View, California, November 2017. PMLR.

[26] J. Thomason, J. Sinapov, M. Svetlik, P. Stone, and R. J. Mooney. Learning multi-modal grounded linguistic semantics by playing "i spy". In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 3477–3483. AAAI Press, 2016.

[27] J. Thomason, S. Zhang, R. Mooney, and P. Stone. Learning to interpret natural language commands through human-robot dialog. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1923–1929. AAAI Press, 2015.

[28] J. Thomason, S. Zhang, R. J. Mooney, and P. Stone. Learning to interpret natural language commands through human-robot dialog. In *IJCAI*, pages 1923–1929, 2015.

[29] H. Yu, H. Zhang, and W. Xu. Interactive grounded language acquisition and generalization in a 2d world. In *International Conference on Learning Representations*, 2018.

[30] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3357–3364, May 2017.