Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation: Supplementary Material

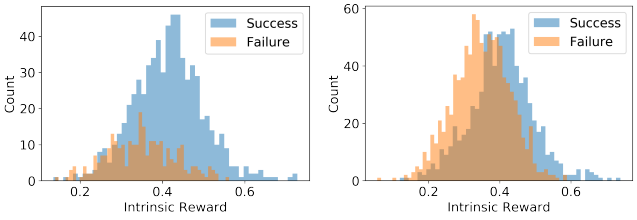
Xin Wang[1] Qiuyuan Huang[2] Asli Celikyilmaz[2] Jianfeng Gao[2] Dinghan Shen[3]
Yuan-Fang Wang[1] William Yang Wang[1] Lei Zhang[2]
[1]University of California, Santa Barbara [2]Microsoft Research, Redmond [3]Duke University
{xwang,yfwang,william}@cs.ucsb.edu, {qihua,jfgao,leizhang}@microsoft.com
aslicel@exchange.microsoft.com, dinghan.shen@duke.edu

A. Training Details

Following prior work [2, 10, 4], ResNet-152 CNN features [5] are extracted for all images without fine-tuning. The pretrained GloVe word embeddings [7] are used for initialization and then fine-tuned during training. All the hyper-parameters are tuned on the validation sets. We adopt the panoramic action space [4] where the action is to choose a navigable direction from the possible candidates. We set the maximal length of the action path as 10. The maximum length of the instruction is set as 80 and longer instructions are truncated. We train the matching critic with a learning rate 1e-4 and then fix it during policy learning. Then we warm start the policy via supervised learning loss with a learning rate 1e-4, and then switch to RL training with a learning rate 1e-5. Self-supervised imitation learning can be performed to further improve the policy: during the first epoch of SIL, the loaded policy produces 10 trajectories, of which the one with the highest intrinsic reward is stored in the replay buffer; those saved trajectories are then utilized to fine-tune the policy for a fixed number of iterations (the learning rate is 1e-5). Early stopping is used for all the training and Adam optimizer [6] is used to optimize all the parameters. To avoid overfitting, we use an L2 weight decay of 0.0005 and a dropout ratio of 0.5. The discounted factor γ of our cumulative reward is 0.95. The weight σ of the intrinsic reward is set as 2.

B. Network Architecture

Reasoning Navigator The language encoder consists of an LSTM with hidden size 512 and a word embedding layer of size 300. The inner dimensions of the three attention modules used to compute the history context, the textual context, and the visual context are 256, 512, and 256 respectively. The trajectory encoder is an LSTM with hidden size 512. The action embedding is a concatenation of the visual appearance feature vector of size 2048 and the orientation feature vector of size 128 (the 4-dimensional orientation feature $[sin\psi; cos\psi; sin\omega; cos\omega]$ are tiled 32 times as used in [4]). The action predictor is composed of three weight matrices: the projection dimensions of W_c and W_u are both 256, and then an output layer W_o together with a softmax layer are followed to obtain the probabilities over the possible navigable directions.



(a) Seen Validation (b) Unseen Validation

Figure 1: Visualization of the intrinsic reward on seen and unseen validation sets.

Matching Critic The matching critic consists of an attention-based trajectory encoder with the same architecture as the one in the navigator, its own word embedding layer of size 300, and an attention-based language decoder. The language decoder is composed of an attention module (whose projection dimension is 512) over the encoded features, an LSTM of hidden size 512, and a multi-layer perceptron (Linear $\rightarrow$ Tanh $\rightarrow$ Linear $\rightarrow$ SoftMax) that converts the hidden state into probabilities of all the words in the vocabulary.

C. Visualizing Intrinsic Reward

In Figure 1, we plot the histogram distributions of the intrinsic rewards (produced by our submitted model) on both seen and unseen validation sets. On the one hand, the intrinsic reward is aligned with the success rate to some ex-

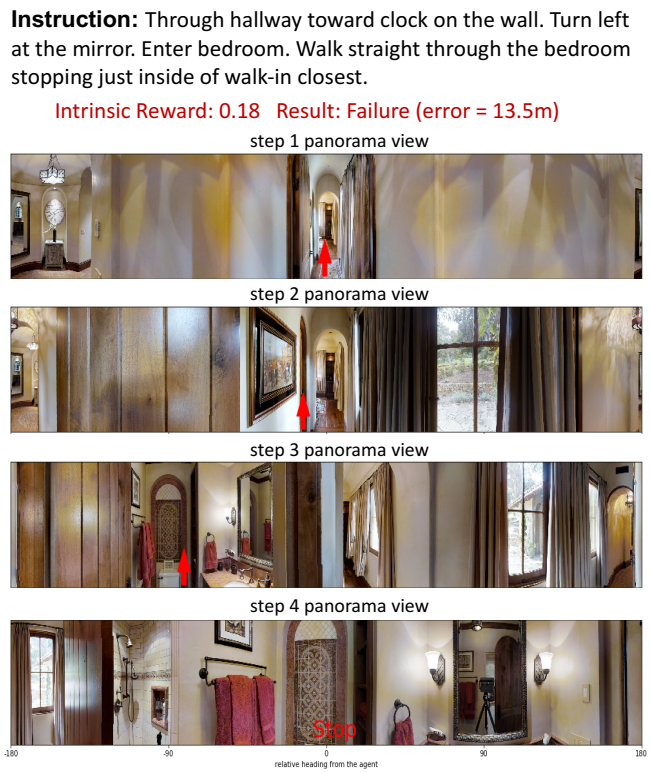


Figure 2: Misunderstanding of the instruction.

tent, because the successful examples are receiving higher averaged intrinsic rewards than the failed ones. On the other hand, the complementary intrinsic reward provides more fine-grained reward signals to reinforce multi-modal grounding and improve the navigation policy learning.

D. Error Analysis

In this section, we further analyze the negative examples and showcase a few common errors in the vision-language navigation task. First, a common mistake comes from the misunderstanding of the natural language instruction. Figure 2 demonstrate such a qualitative example, where the agent successfully perceived the concepts "hallway", "turn left", and "mirror" etc., but misinterpreted the meaning of the whole instruction. It turned left earlier and mistakenly entered the bathroom instead of the bedroom at Step 3.

Secondly, failing to ground objects in the visual scene can usually result in an error. As shown in Figure 3 (a), the agent did not recognize the "mannequins" in the end (Step 5) and stopped at a wrong place even though it executed the instruction pretty well. Similar in Figure 3 (b), the agent failed to detect the "red ropes" at the beginning (Step 1) and thus took a wrong direction which also has the "red carpet". Note that "mannequins" is an out-of-vocabulary word in the training data; besides, both "mannequins" and "red ropes" do not belong to the 1000 classes of the ImageNet [3], so the visual features extracted from a pretrain ImageNet model [5] are not able to represent them.

In Figure 4, we illustrate a long negative trajectory which our agent produced by following a relatively complicated instruction. In this case, the agent match "the floor is in a circle pattern" with the visual scene, which seems to be another limitation of the current visual recognition systems. The above examples also suffer from the error accumulation issue as pointed out by Wang *et al*. [10], where one bad decision leads to a series of bad decisions during the navigation process. Therefore, an agent capable of being aware of and recovering from errors is desired for future study.

E. Trails and Errors

Below are some trials and errors from our experimental experience, which are not the gold standard and used for reference only.

- We tried to incorporate dense bottom-up features as used in [1], but it hurt the performance on unseen environments. We think it is possibly because the navigation instructions require sparse visual representations rather than dense features. Dense features can easily lead to the overfitting problem. Probably more fine-grained detection results rather than dense visual features would help.
- The performances are similar with or without positional encoding [9] on the instructions.
- Pretrained ELMo embeddings [8] without fine-tuning hurts the performance. The summation of pretrained ELMo embeddings and task-specific embeddings has a similar effect of task-specific embeddings only.
- It is not stable to only use the intrinsic reward to train the model. So we adopt the mixed reward for reinforcement learning, which works the best.

References

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3, page 6, 2018. 2

[2] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2018. 1

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 2

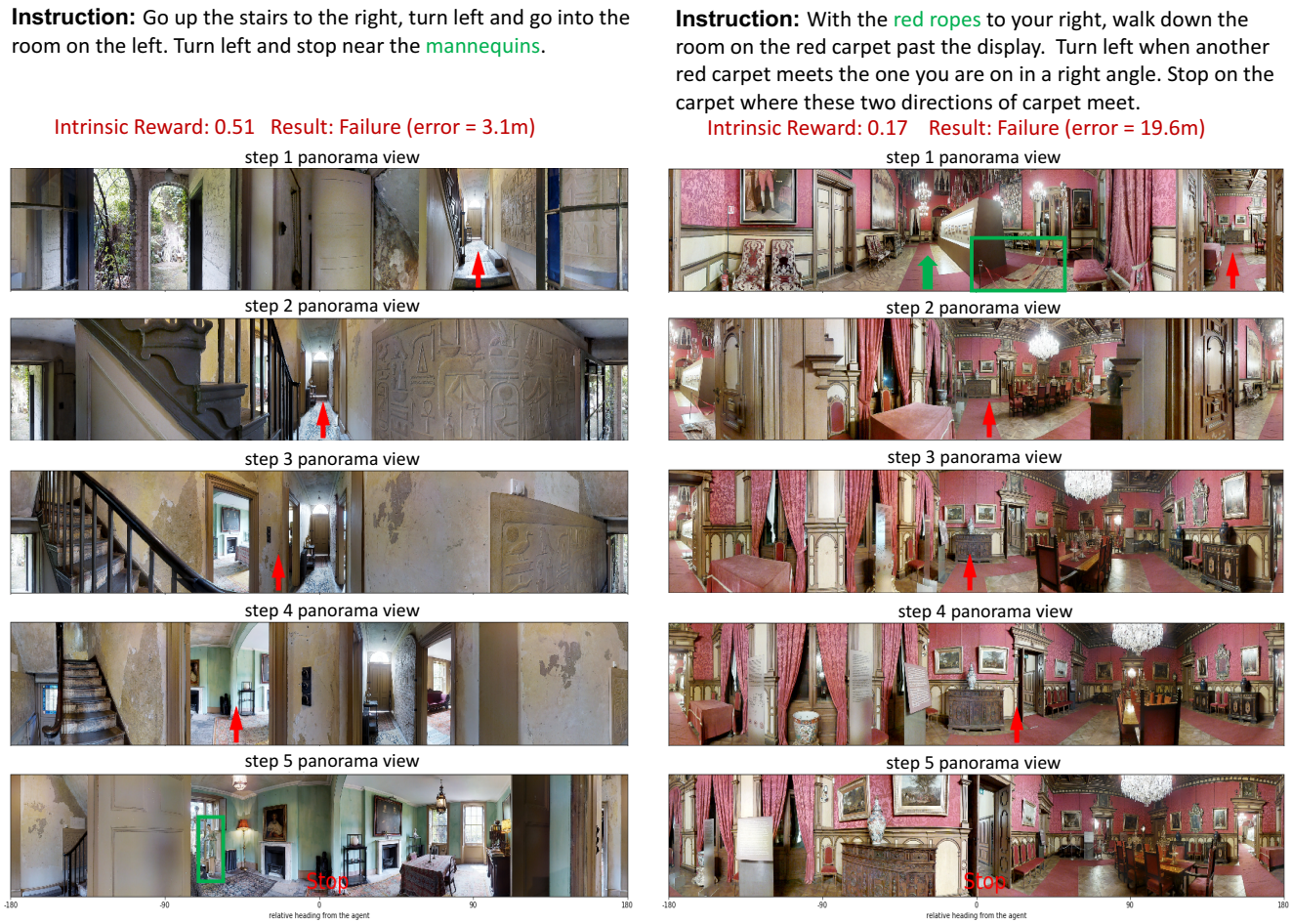


Figure 3: Ground errors where objects were not recognized from the visual scene.

[4] D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 1

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2

[6] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[7] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 1

[8] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018. 2

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 2

[10] 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 *The European Conference on Computer Vision (ECCV)*, September 2018. 1, 2

Instruction: Turn around and exit the room to the right of the TV. Once out turn left and walk to the end of the hallway and then turn right. Walk down the hallway past the piano and then stop when you enter the next doorway and the floor is in a circle pattern.

Intrinsic Reward: 0.39 Result: Failure (error = 5.4m)

Figure 4: Failure of executing a relatively complicated instruction.