Appendix for The Regretful Agent: Heuristic-Aided Navigation through Progress Estimation

Chih-Yao Ma*†, Zuxuan Wu‡, Ghassan AlRegib†, Caiming Xiong§, Zsolt Kira†
†Georgia Institute of Technology, ‡University of Maryland, College Park, §Salesforce Research

Appendix

A. Network Architecture

The embedding dimension of the instruction encoder is 256, followed by a dropout layer with ratio 0.5. We encode the instruction using a regular LSTM, and the hidden state is 512 dimensional. The MLP g used for projecting the raw image feature is $BN \rightarrow FC \rightarrow BN \rightarrow Dropout \rightarrow ReLU$. The FC layer projects the 2176-d input vector to a 1024-d vector, and the dropout ratio is set to be 0.5. The hidden state of the LSTM which allows integration of information across time is 512. When using the progress marker, the markers are tiled $n = 32$ times. The dimension of the learnable matrices are: $\boldsymbol{W}_x \in \mathbb{R}^{512\times512}$, $\boldsymbol{W}_v \in \mathbb{R}^{512\times1024}$, $\boldsymbol{W}_a \in \mathbb{R}^{1024\times1024}$, $\boldsymbol{W}_r \in \mathbb{R}^{1\times2}$, $\boldsymbol{W}_{fr} \in \mathbb{R}^{1024\times1024}$ without progress marker, and $\boldsymbol{W}_{fr} \in \mathbb{R}^{1024\times1056}$ with progress marker.

B. Modern Reinforcement Learning

Modern Reinforcement Learning methods like Asynchronous Advantage Actor Critic (A3C) [4] or Advantage Actor Critic (A2C) methods are related to the baseline Self-Monitoring agent [3] and the proposed Regretful agent. Specifically, the progress monitor in the Self-Monitoring agent (our baseline) is similar to the value function in RL, and the difference between progress marker of a viewpoint and current progress estimation (denote as $\Delta\boldsymbol{v}_{t,k}^{marker}$, see Sec. 4.2 in the main paper) is conceptually similar to the advantage function. However, the advantage function in RL serves as a way to regularize and improve the training of the policy network. We instead associate the $\Delta\boldsymbol{v}_{t,k}^{marker}$ directly to all navigable states, and this $\Delta\boldsymbol{v}_{t,k}^{marker}$ has a direct impact on the agent deciding next action even during inference. While having an accurate value estimate for VLN with dynamic and implicit goals may reduce the need for this formulation, we however believe that this is hardly possible because of the lack of training data. On the other hand, relating to the proposed end-to-end learned regret module, *Leave no Trace* [1] learns a forward and a *reset* policy to reset the environment for preventing the policy entering a non-reversible state. Instead of learning to reset, we learn to rollback to a previous state and continue the navigation task with a policy network that learns to decide a better next step.

* Work partially done while the author was a research intern at Salesforce Research.

Table 1. Comparison of our regretful agent using greedy action selection with beam search.

Method	Beam search	Test set (leaderboard) NE ↓	SR ↑	Length ↓	SPL ↑
Speaker-Follower [2]		6.62	0.35	14.82	0.28
	△	4.87	0.53	1257.38	0.01
Self-Monitoring [3]		5.99	0.43	17.11	0.32
	△	**4.48**	**0.61**	373.09	0.02
Regretful		5.69	0.48	**13.69**	**0.40**

C. Comparison with Beam Search Methods

We compare our method using greedy action selection with existing beam search approaches, e.g., Pragmatic Inference in Speaker-Follower [2] and progressed integrated beam search in Self-Monitoring agent [3]. We can see in Table 1 that, while beam search methods perform well on success rate (SR), their trajectory lengths are significantly longer, achieving low success rate weighted by Path Length (SPL) scores and therefore are impractical for real-world applications. On the other hand, our proposed method significantly improved both SR and SPL when not using beam search.

D. Qualitative Analysis

D.1. Successful examples

We show the complete trajectory of the agents successfully deciding when to roll back and reach the goal in unseen environments in Figure 1, 2, 3, and 4.

In Figure 1, we demonstrate that the agent is capable of performing a local search on the navigation graph. Specifically, from step 0 to step 3, the agent searched two possible directions and decided to move with one particular direction at step 4. Once it reached step 5, the agent decides to continue to move forward, and we observed that

the progress estimate significantly increased to 45% at step 7. Interestingly, unlike other examples we have shown, the agent did not decide to roll back despite the progress estimate slightly decreased from 45% to 40%. We reckon that this is one of the advantages of using a learning-based regret module, where a learned and dynamically changing threshold decides when to rollback. Finally, the agent successfully stopped in front of the microwave.

In Figure 2, the agent is instructed to *walk across living room*. It is ambiguous since both directions seem like a living room. Our agent first decides to move into the direction that leads to a room with a kitchen and living room. It then decided to roll back with the progress monitor output slightly decreased. The agent then followed the rest of the instruction successfully with the progress monitor steadily increased at each step after that. Finally, the agent decides to stop with the progress estimate 99%.

In Figure 3, the agent first moved out of the room and walked up the stairs as instructed, but the second set of stairs makes the instruction ambiguous. The agent continued to walk up the stairs for one more step and then decided to go down the stairs at step 4. As the agent decided to turn right at step 6, we can see the progress estimate significantly increased from 51% to 66%. Once the agent entered the TV room, the progress estimate increased again to 82%. Finally, the agent successfully stopped with the progress monitor output 95%.

In Figure 4, the agent failed to *walk down the stairs* at step 1. Because of the proposed Regret Module and Progress Marker, the agent was able to discover the correct path to go downstairs. Once walking down, the progress estimate increased to 39% immediately, and as the agent goes further down, the progress estimate reached 98% by the time the agent reached the bottom of the stairs. Finally, the agent decided to wait by the bamboo plant with progress estimate 99%.

D.2. Failed examples

We have shown how the agent can successfully utilize the rollback mechanism to reach the goal, even though it is not familiar with the environment and likely to be uncertain about some actions it took. Intuitively, the rollback mechanism can increase the chance that the agent reaches the goal as long as the agent can correctly decide when to stop.

We now discuss two failed examples of our proposed regretful agent in unseen environments that highly resemble the successful examples in terms of the given instruction and ground-truth path. Both examples demonstrate that the agent successfully rolled back to the correct path towards the goal but failed to stop at the goal.

Specifically, in Figure 5, the agent reaches the room with the white cabinet as instructed but decided to move one step forward. The agent then decided to roll back to the room correctly at step 5. However, this does not help the agent to stop at the goal resulting in a failed run.

On the other hand, in Figure 6, we can see that the progress estimate at step 5 significantly dropped by 21%, and the agent correctly decided to roll back. The agent then successfully reached the refrigerator but did not stop immediately. It continued to move forward after step 8, resulting in an unsuccessful run.

Lastly, we discuss a failed example when the agent incorrectly decided when to roll back. In Figure 7, the agent first followed the instruction to *go down the hallway* and tried to find the second door to turn right. As the agent reached the end of the hallway at step 4, it decided to roll back since there is no available navigable direction that leads to *turn right*. The agent then decided to go down the hallway again with completely opposite direction. However, the agent decided to roll back again at step 7 with the progress estimate dropped to 18%. Although the agent eventually was able to *escape* from the hallway leading to the dead end, it ends up unsuccessful.

References

[1] Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 1

[2] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 1

[3] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 1

[4] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning (ICML)*, pages 1928–1937, 2016. 1

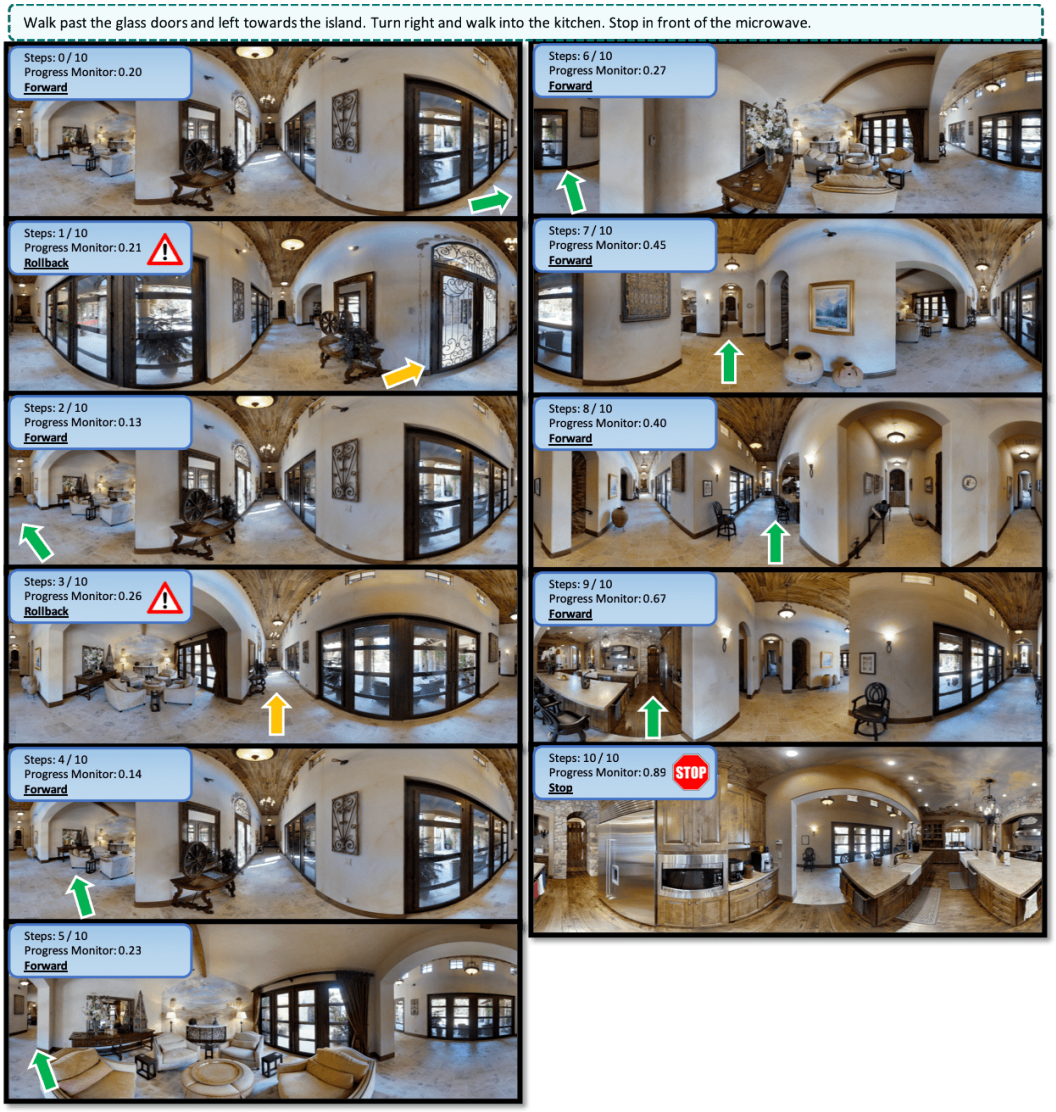

Figure 1. The first part of the instruction *walk past the glass doors* is ambiguous since there are multiple directions that lead to glass doors, and naturally the agent is confused and uncertain where to go. Our agent is able to perform local search on the navigation graph and decides to roll back multiple times at the beginning of the navigation. At step 6, the agent performs an action *turn right*. Consequently, the progress estimate at step 7 significantly increased to 45%. Interestingly, the agent continues to move forward even though the progress estimate slightly decreased from step 7 to step 8. We reckon that this as one of the advantage of using a learning-based regret module as opposed to using a hard-coded threshold. The agent then successfully follows the instruction and *stops in front of the microwave* with progress estimate 89%.

Figure 2. The agent first *walk across living room*, but decides to move into the direction that leads to kitchen and dinning room. At step 1, the agent decides to roll back due to a decreasing of the progress monitor output. The agent then followed the rest of the instruction successfully with the progress monitor steadily increased at each step. Finally, the agent decides to stop with the progress estimate 99%.

Figure 3. The agent walked up the stairs as instructed at step 1, but the second set of stairs makes the instruction ambiguous. The agent continues to walk up stairs but soon realized that it needs to go down the stairs and *turn right* from step 4 - 6. When the agent decides to turn right, we can see the progress estimate significantly increased from 51% to 66%. As the agent turned right to the TV room, the progress estimate increased again to 82%. Finally, the agent stops with the progress monitor output 95%.

Figure 4. The agent walks down the hall way to the stairs but failed to *walk down the stairs* at step 1. With a small increase on the progress monitor output, the agent then decides to roll back and take the action to walk down the stairs. Once walking down, we can see the progress estimate increased to 39%, and as the agent goes further down, the progress estimate reached 98% at the bottom of the stairs. Finally, the agent decides to stop near by the bamboo plant with progress estimate 99%.

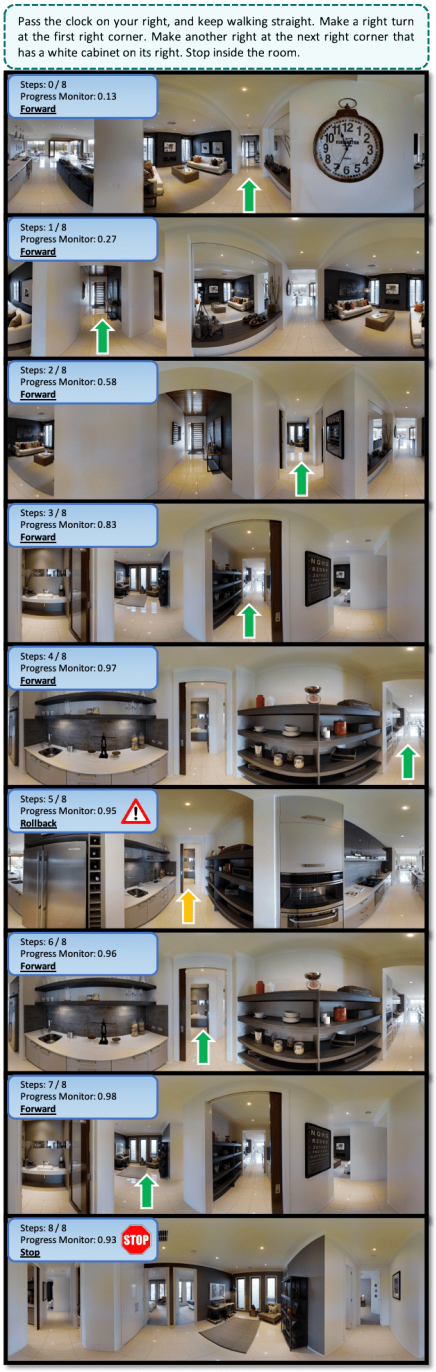

Figure 5. **Failed example.** The agent starts to navigate through the unseen environment by following the given instruction. It was able to successfully follow the instruction and correctly reach the goal at step 4. The agent then decided to move forward towards the kitchen and correctly decided to roll back to the goal. However, the agent did not stop and continue to explore the environment and eventually stopped a bit further from the goal.

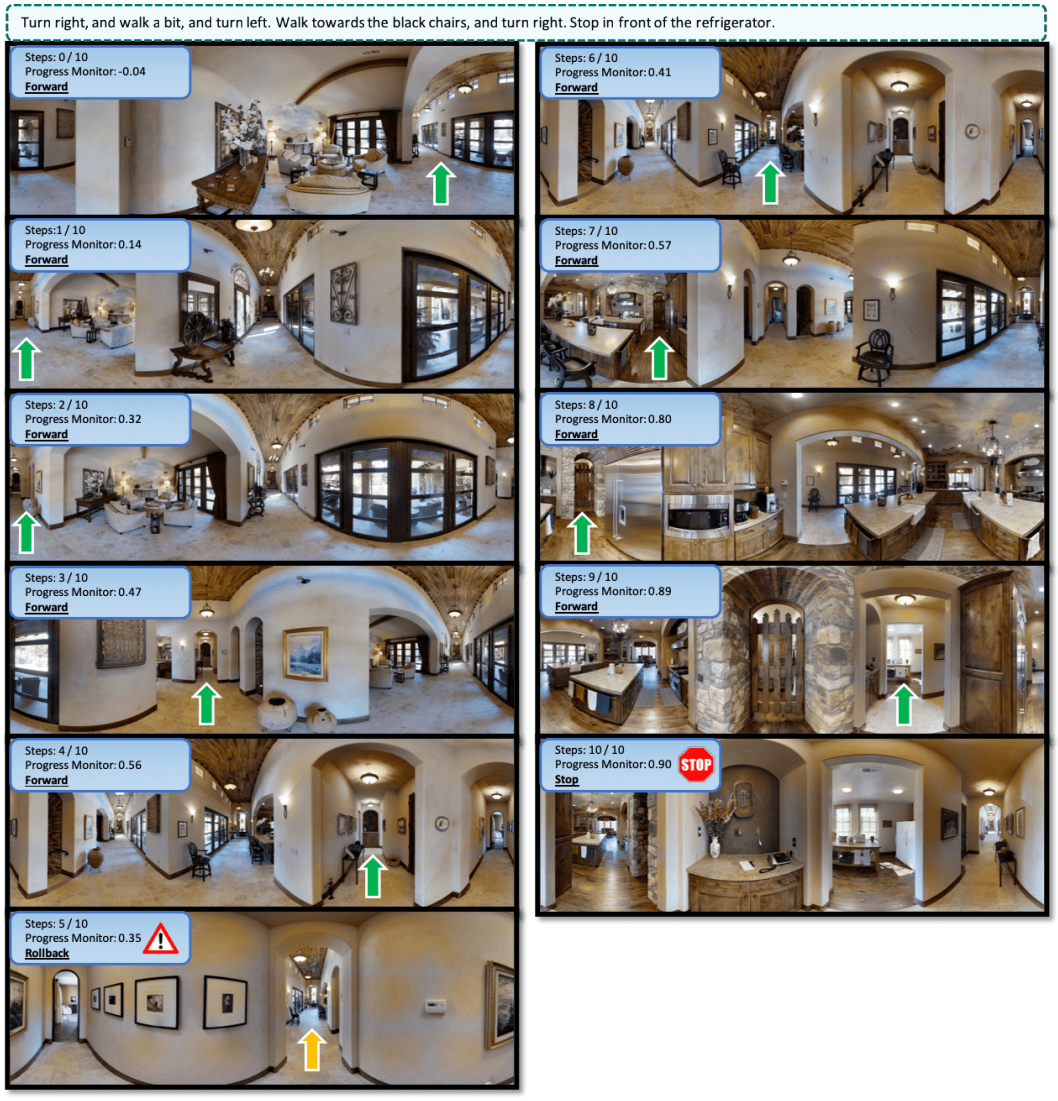

Figure 6. The agent correctly followed the first parts of the instruction until step 4, but it decided to move forward towards the hall. At step 5, the agent correctly decided to roll back with the progress estimate decreased from 56% to 35%. The agent was then able to follow the rest of the instruction successfully and reach the refrigerator at step 8. However, the agent did not stop nearby the refrigerator and continued to take another two forward steps.

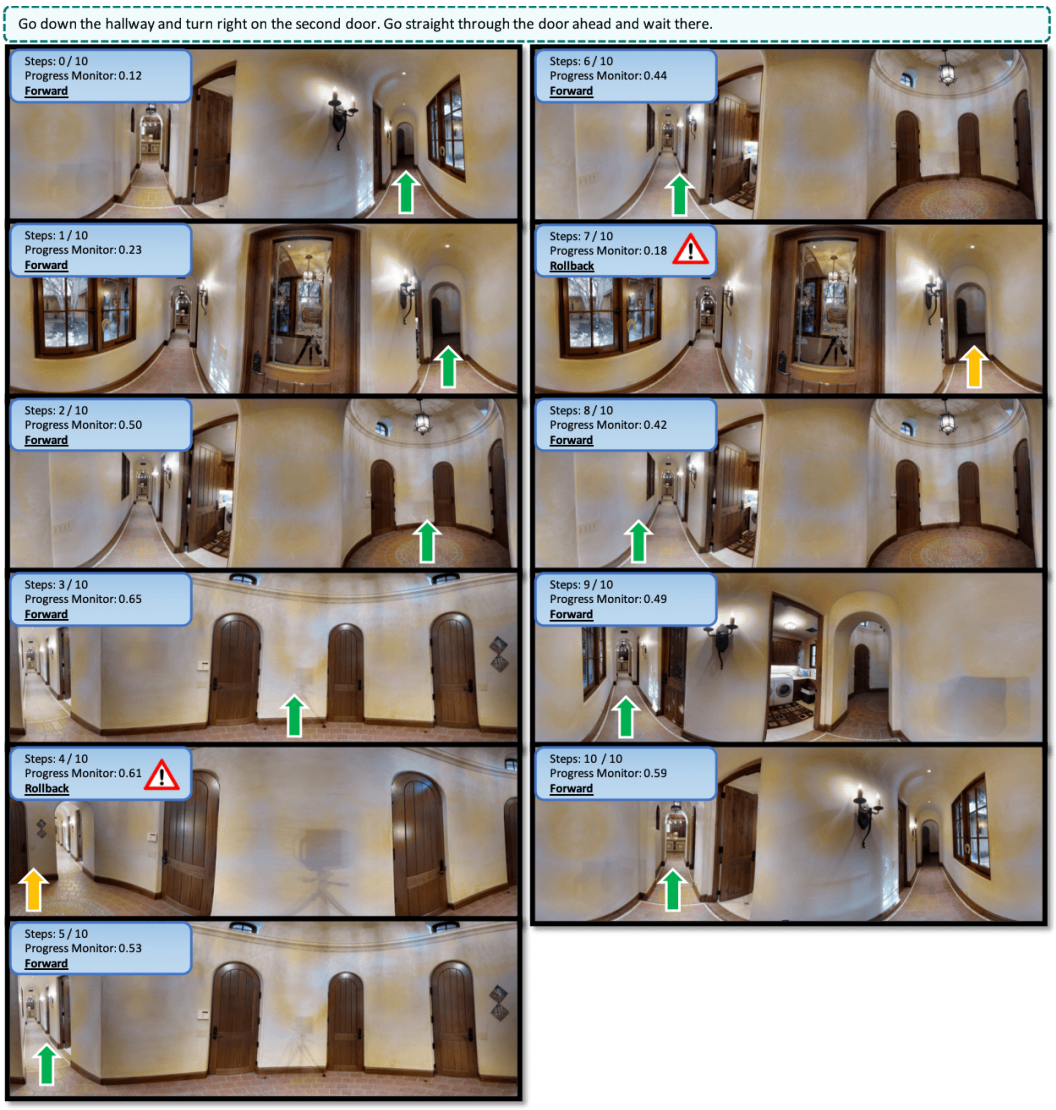

Figure 7. The agent followed the first part of instruction to *go down the hallway*. As the agent reached the end of the hallway, it was not able to find the second door to turn left. The agent then decided to roll back at step 4 with progress estimate decreased from 65% to 61%. The agent continued to go back towards the hallway but decided to roll back again at step 7. Although the agent was able to correct its errors made at the first few steps and *escape* from the hallway leading to the dead end, it ends up unsuccessful.