Active Visual Information Gathering for Vision-Language Navigation
Supplementary Material

Hanqing Wang[1], ✉Wenguan Wang[2], Tianmin Shu[3], Wei Liang[1], and Jianbing Shen[4]

[1]School of Computer Science, Beijing Institute of Technology [2]ETH Zurich
[3]Massachusetts Institute of Technology [4]Inception Institute of Artificial Intelligence
https://github.com/HanqingWangAI/Active_VLN

In this document, we first detail the work and data flow of our navigation agent (see §2). Then in §3, more ablation studies are conducted to fully assess the effectiveness of our learning protocol. Later, in §4, visual results for some representative successful and failed cases as well as analyses are presented. Finally, in §5, we provide more statistics of our exploration module, to give an in-depth glimpse into our agent.

1 Memory based Late Action-Taking Strategy

As described above, after a round of exploration, the agent goes back to the starting point at t^{th} navigation step to update the knowledge collected about a certain direction. However, this "going back" may result in the repeated visiting of viewpoints in three manners, 1) the re-visiting of the viewpoints on the way back to the starting point, 2) the re-visiting of the viewpoints within the rounds of exploration at t^{th} navigation step. Though each exploration leads to a certain direction (viewpoint), those directions may interact somewhere in front and the intersection may be re-visited. 3) the re-visiting of the viewpoints across different navigation steps, *e.g.*, the viewpoints explored at t^{th} navigation step may be re-visited at $t+1^{th}$ navigation step. Those re-visiting will lead to additional TL.

To shrink the trajectory, a novel moving strategy is adopted. We call it "Lazy" strategy. The core idea of the strategy is that the agent moves only when it is necessary. Specifically, supposing that the agent finished a round exploration at viewpoint v_t. We store the views of all the visited points and the connectivity between the points in a memory graph, so the agent can "image" the execution of his following action without really going back to the starting point. When he needs to visit a point (assumed as v) that is not stored in the memory graph, the new point v will be added to the memory graph and the agent will go to v from v_t following the shortest path in the memory graph, then the view of v will be stored. The agent will not leave v until he needs to visit a new point that is not stored in the memory graph or he decides to stop the navigation. In such "Lazy" way, a lot of repeated visitings are saved.

✉ Corresponding author: *Wenguan Wang* (wenguanwang.ai@gmail.com).

After making exploration towards a certain direction, if the agent directly "goes back" the starting position and making next-round exploration/navigation, it may cause a lot of revisits. To alleviate this, we let the agent store the visited views during exploration in an outside memory. The agent then follows a late action-taking strategy, *i.e.*, moving only when it is necessary. With the memory, the agent can "image" the execution of his following actions without really going back. When he needs to visit a point that is not stored in the memory, he will go to that point from his current position directly and the memory will be updated accordingly. Then agent will stay at there until he needs to visit a new point that is not met before. See the supplementary for more details.

2 Work and Data Flow

Fig. 1 depicts our work and data flow during t^{th} step navigation. At first, the exploration module decides where to explore. Supposing k direction is selected, the agent continues to make exploration decisions and collects surrounding visual information, until `STOP` action is selected. Then he returns to t^{th} navigation point, updates the visual feature of k direction, and prepares for next-round exploration. If the exploration module make a stop decision, the updated visual feature is used to make a navigation decision. Otherwise he will select a direction to make exploration.

3 Diagnostic Experiments of Training Signals

In this section, we make diagnostic experiments to analyze the effectiveness of our training signals. Our agent is trained with two distinct learning paradigms, *i.e.*, (1) imitation learning, and (2) reinforcement learning. Our basic agent [2] has examined the effectiveness of the reinforcement learning loss $\mathcal{L}^{\mathrm{nv}}_{\mathrm{RL}}$ for the whole network training. Here we design the experiments to evaluate the effectiveness of the imitation learning loss $\mathcal{L}^{\mathrm{ep}}_{\mathrm{IL}}$ and the reinforcement learning loss $\mathcal{L}^{\mathrm{ep}}_{\mathrm{RL}}$ used during the training of our exploration module. Specifically, we retrain our agent (at most 4-step exploration) without either $\mathcal{L}^{\mathrm{ep}}_{\mathrm{IL}}$ or $\mathcal{L}^{\mathrm{ep}}_{\mathrm{RL}}$ in the final loss.

As shown in Table 1, with either imitation learning or reinforcement learning signal, our full-model achieves higher **SR** compared to the basic agent, which is improved by 1% and 3% on validation unseen set respectively. This shows the effectiveness of two training signals. Note that the agent with the mixture of $\mathcal{L}^{\mathrm{ep}}_{\mathrm{IL}}+\mathcal{L}^{\mathrm{ep}}_{\mathrm{RL}}$ outperforms the one without $\mathcal{L}^{\mathrm{ep}}_{\mathrm{RL}}$ or $\mathcal{L}^{\mathrm{ep}}_{\mathrm{IL}}$ by 5% and 3% on validation unseen respectively. It means that the mixture signal somewhat overcomes the relative poor generalizability of IL and difficult convergence of RL.

4 Qualitative Analysis

In this section, we visualize some successful and failed cases of our agent and further provide in-depth analyses.

Table 1. Ablation study about training signals on `validation seen` and `validation unseen` sets of R2R dataset [1] under **Single Run** setting.

Model	Loss Component	Single Run Setting									
		`validation seen`					`validation unseen`				
		SR↑	NE↓	TL↓	OR↑	SPL↑	**SR**↑	NE↓	TL↓	OR↑	SPL↑
basic agent	$\mathcal{L}^{\text{nv}}_{\text{IL}} + \mathcal{L}^{\text{nv}}_{\text{RL}}$	0.62	3.99	11.0	0.71	0.59	0.52	5.22	10.7	0.58	0.48
full model	w/o. $\mathcal{L}^{\text{ep}}_{\text{IL}}$ $\mathcal{L}^{nv}_{IL} + \mathcal{L}^{nv}_{RL} + \mathcal{L}^{ep}_{RL}$	0.65	3.70	18.3	0.76	0.53	0.53	5.15	17.1	0.65	0.39
	w/o. $\mathcal{L}^{\text{ep}}_{\text{RL}}$ $\mathcal{L}^{nv}_{IL} + \mathcal{L}^{nv}_{RL} + \mathcal{L}^{ep}_{IL}$	0.67	3.53	17.6	0.79	0.51	0.55	4.65	18.9	0.68	0.39
	full loss $\mathcal{L}^{nv}_{IL} + \mathcal{L}^{nv}_{RL} + \mathcal{L}^{ep}_{IL} + \mathcal{L}^{ep}_{RL}$	0.70	3.20	19.7	0.80	0.52	0.58	4.36	20.6	0.70	0.40
w/o. "Lazy" strategy	full loss	0.70	3.20	45.3	0.80	0.29	0.58	4.36	44.5	0.70	0.18

4.1 Successful Examples

We show three representative examples, over which the basic agent fails while our agent succeeds.

For the first case in Fig. 2, the instruction, "*Turn toward your left*", is inconsistent with the initial orientation of the agent. Additionally, the important landmark, "*grey fence*", can be observed from two different directions. These issues result in the confusion of the basic agent and unluckily lead to failure. Our agent conducts the active exploration to mitigate the ambiguity of the instruction and finally chooses the right route.

The second case is shown in Fig. 3. In this case, the instruction said "*Go up the stairs one level*$\cdots$", where "*one level*" has ambiguous definition in the dataset as the routes are annotated by different workers. The basic agent continues to climb another staircase and fails finally. After the exploration, our agent finds that after climbing another staircase, he cannot execute the following instructions. Then the agent chooses the right direction.

In Fig. 4, the initial instruction is ambiguous, as the visual context of both directions are consistent with the instruction "$\cdots$ *past the large table and chairs.*". Moreover, if considering the mismatch between the instruction "*Go straight* $\cdots$" and the agent's initial orientation, the wrong direction even looks more consistent with the instruction. After two rounds of exploration, our agent finds that the last half routes of the exploration in the right direction is more consistent with the following instructions, which is more related to the goal viewpoint comparing to the starting instruction. Then our agent chooses the right direction.

4.2 Failed Examples

In this section, we show a representative failed example, which also demonstrates possible directions of future efforts.

As shown in Fig. 5, our agent first correctly executes the instruction "*Walk up the flight of stairs to the top.*". At the top of the stairs, our agent is confused by the following instruction "*Turn and follow the railing into the bedroom area.*". This because there are three directions to different bedrooms and two of them have railing along the way. Thus our agent decides to explore the surroundings.

During the first-round exploration, the agent explores the front door and finds this direction is wrong. Then the probability of this direction is suppressed (0.3→0.2). Then, the agent selects a new direction and makes a second-round exploration. Though this direction is correct, the agent does not step into the correct room and misses the critical landmark, "*closet*". Thus the probability of the correct direction is mistakenly suppressed (0.3→0.1). In the third-round exploration, the agent explores the door on the right and finds that it is still inconsistent with the description of the final goal viewpoint. The probability of the last possible direction is suppressed again (0.4→0.2). Then our agent picks the most plausible direction, which is of the highest navigation probability, but fails. We think there are two possible reasons for the failure of the exploration. First, our exploration module mainly focuses on current navigation action related instruction, but lacks the ability of dynamically parsing the following instructions during exploration. Maybe a multi-head co-attention or future navigation action aware exploration module is needed. Second, current navigation module may needs to be equipped with a reference grounding ability, thus the essential reference "*closet*" will be addressed during exploration.

5 Statistics Analysis

In this section, we provide more statistic analyses about our exploration module. The statistics are collected from our full agent with 4-step exploration module on the validation unseen split of R2R dataset [1] .

The overall average exploration rate is 15.3%, *i.e.*, the probabilities of making exploration during a navigation step is 15.3%. When the agent decides to explore, the average number of exploration directions is 1.82, the average, pre-direction exploration steps are 2.05. The average trajectory length (**TL**) is 20.6, in which the length of navigation is 9.4, while the length of exploration is 11.2. We can find that the real navigation routes are very short. For the successful and failed navigation cases, the average trajectory lengths are 18.7 and 23.2 respectively. **TL** of failed cases is longer by 4.5, which possibly means that our agent is confused in the failed cases and tries to collect more information through making more explorations.

We also make specific analysis on the effect of exploration by studying how the exploration influences the navigation action decisions. For the navigation steps with exploration, the rate that the original navigation action is changed through exploration is 75.1%, in which the rate that wrong navigation action is corrected is about 86.5%. That means in most cases our exploration module could collect meaningful information for supporting navigation-decision making.

References

1. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: CVPR (2018) 3, 4, 7, 8, 9, 10
2. Tan, H., Yu, L., Bansal, M.: Learning to navigate unseen environments: Back translation with environmental dropout. In: NAACL (2019) 2

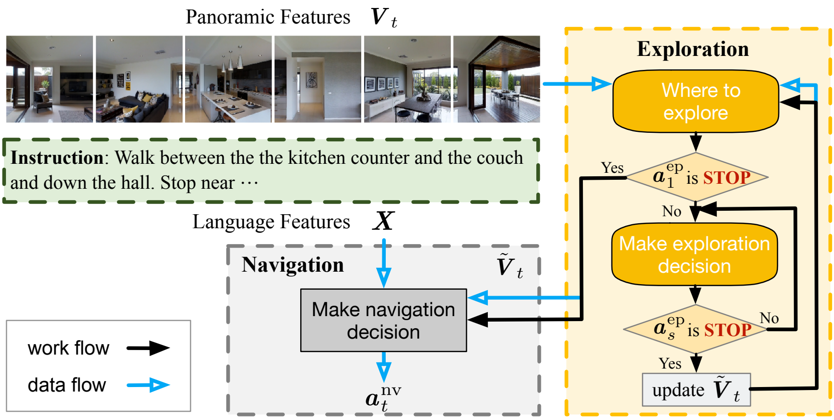

Fig. 1. The chart of work flow and data flow at navigation step t.

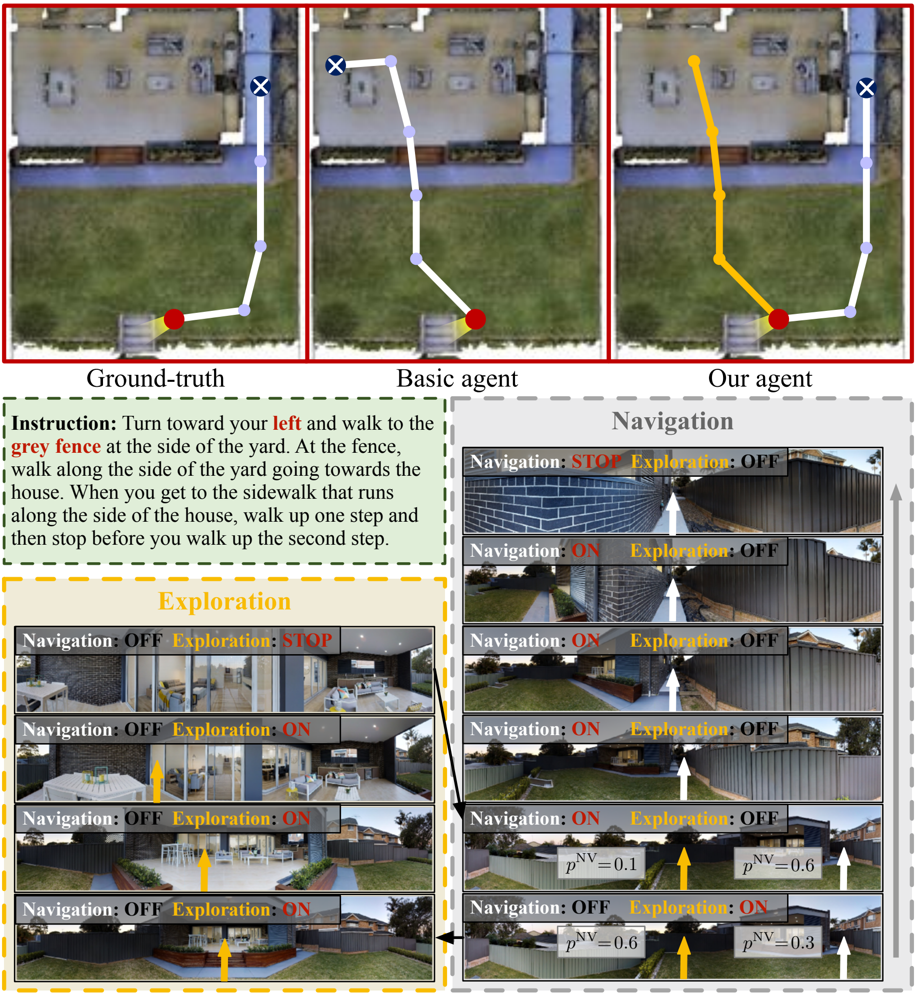

Fig. 2. Case 1. **Top:** Groundtruth navigation route, basic agent's navigation route, and our agent's exploration and navigation routes. Note that the initial orientation of the agent is highlighted. The basic agent gets confused with the wrong instruction "*Turn toward your left*···". In addition, there exist different ways to the "*grey fence*". Thus he fails finally. However, our agent can actively explore the surrounding (the yellow part) and then make more accurate navigation decision. **Bottom Left**: First view during exploration of our agent. **Bottom Right:** First view during navigation of our agent. We can find that, before making exploration, the wrong direction gains a high navigation probability (*i.e.*, 0.6). However, after exploration, the probability for correct direction is improved (*i.e.*, 0.3→0.6). This case is taken from R2R dataset [1], path 3492, instruction set 2.

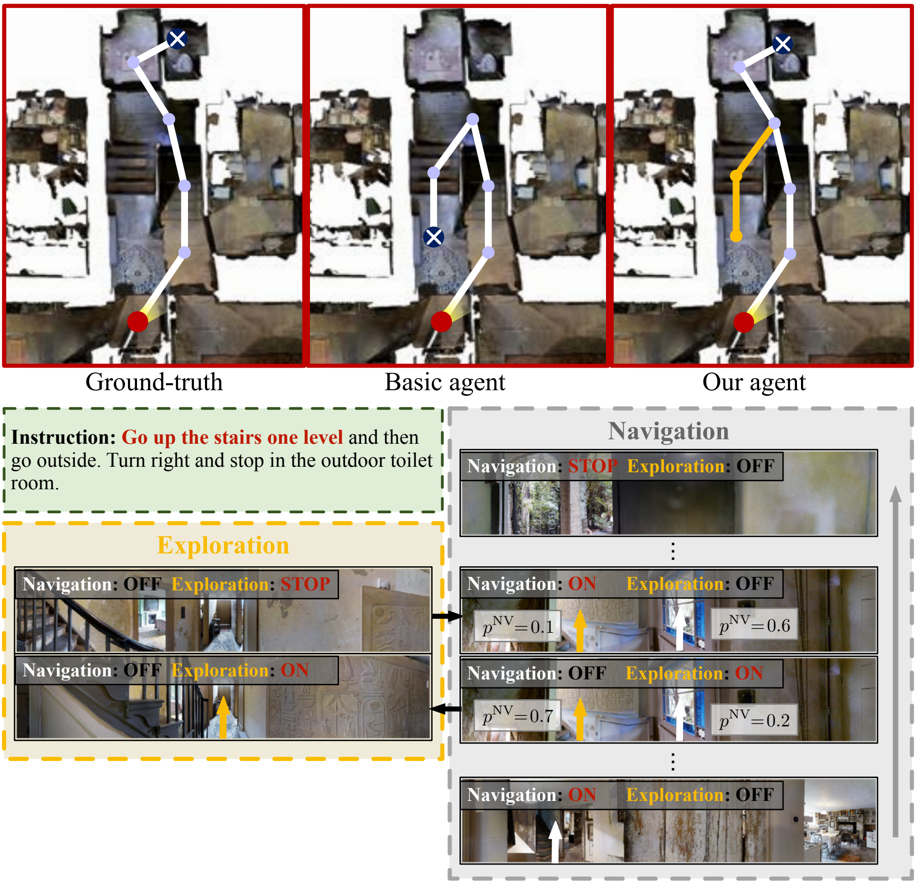

Fig. 3. Case 2. **Top:** Groundtruth navigation route, basic agent's navigation route, and our agent's exploration and navigation routes. Note that the initial orientation of the agent is highlighted. The basic agent is hard to understand the instruction "*Go up the stairs one level*···" and thus continues to climb another staircase. For our agent, after collecting more information, a correct navigation direction is chose and reaches the goal successfully. **Bottom Left**: First view during exploration of our agent. Our agent explores the upstairs and finds the direction is wrong. **Bottom Right:** First view during navigation of our agent. After the exploration, the probability of the correct direction is improved (*i.e.*, 0.2→0.6). This case is taken from R2R dataset [1], path 3766, instruction set 3.

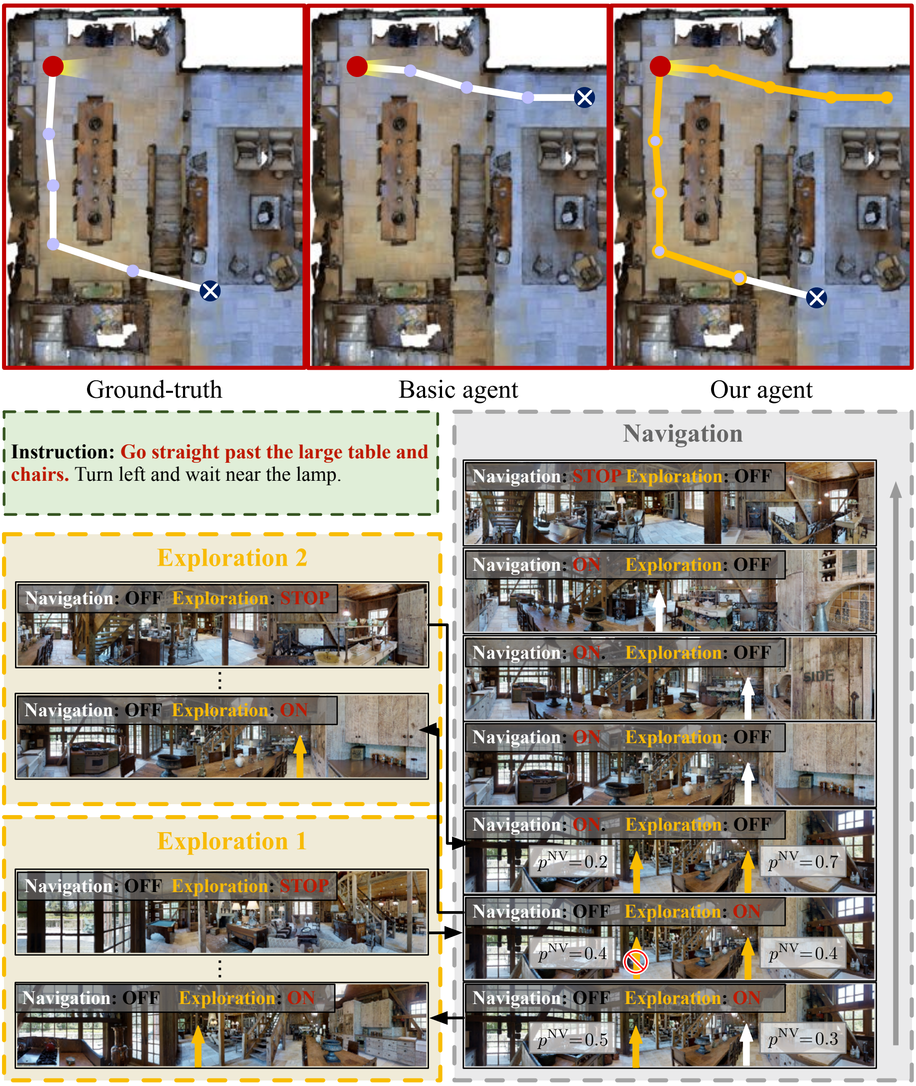

Fig. 4. Case 3. **Top:** Groundtruth navigation route, basic agent's navigation route, and our agent's exploration and navigation routes. Note that the initial orientation of the agent is highlighted. The instruction "*Go straight past the large table and chairs.* $\cdots$" is wrong, as the instructor does not take notice of the agent's initial orientation. **Bottom Left**: First view of our agent during two exploration rounds. **Bottom Right:** First view during navigation of our agent. We can find that, after first-round exploration, the probability of the wrong direction is suppressed (*i.e.*, 0.5→0.4). After the second round of exploration, the probability for correct direction is improved (*i.e.*, 0.4→0.7). This case is taken from R2R dataset [1], path 4782, instruction set 1.

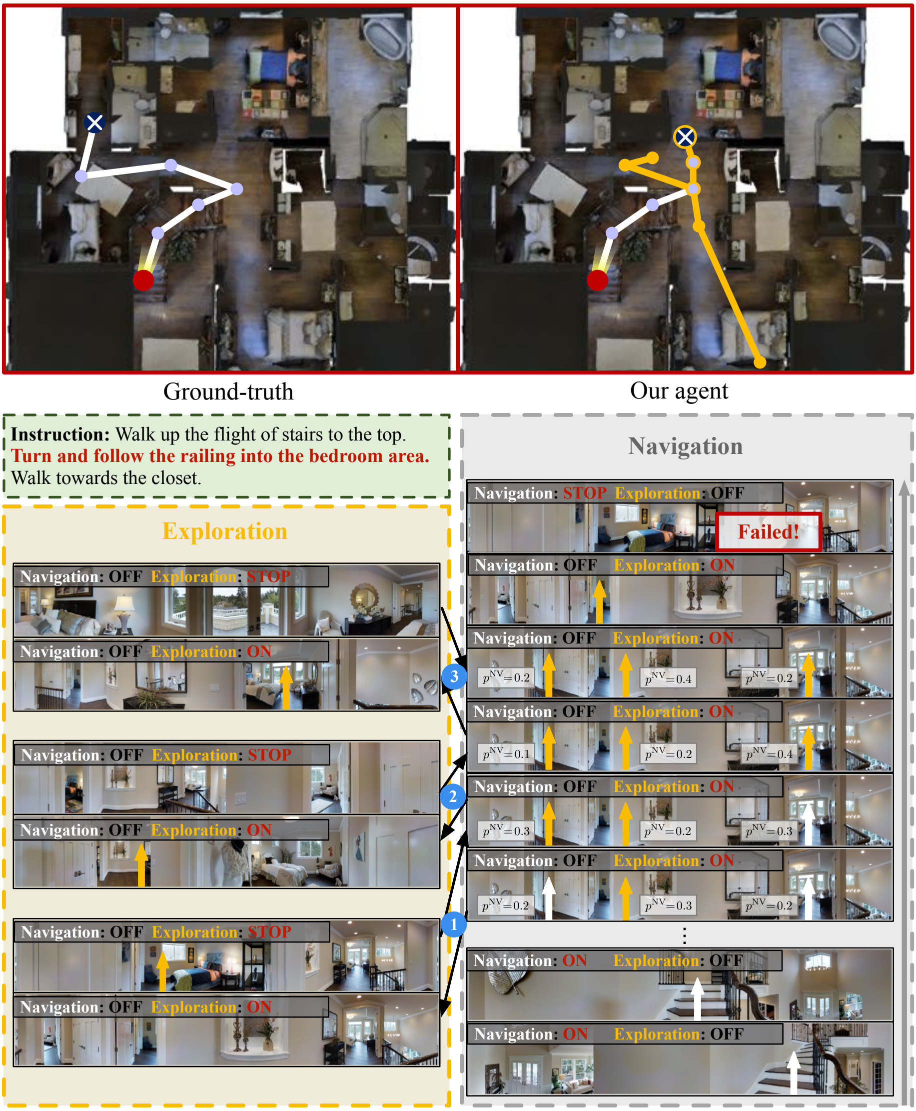

Fig. 5. Failed Case. **Top:** Groundtruth navigation route, basic agent's navigation route, and our agent's exploration and navigation routes. Note that the initial orientation of the agent is highlighted. **Bottom Left**: First view during three rounds of exploration by our agent. **Bottom Right:** First view during navigation of our agent. Our agent conducts three rounds of exploration at the top of the stairs. He fails to collect enough information during the second-round exploration, causing failed navigation. This case is taken from R2R dataset [1], path 21, instruction set 1.