The Supplementary of "GeoVLN: Learning Geometry-Enhanced Visual Representation with Slot Attention for Vision-and-Language Navigation"

Jingyang Huo[*], Qiang Sun[*], Boyan Jiang[*], Haitao Lin, Yanwei Fu[†]
Fudan University
jyhuo22@m.fudan.edu.cn, {18110860051, 18110240008, 19110860015, yanweifu}@fudan.edu.cn

1. Details of the expansion on HAMT

Here, we describe how to extend HAMT with our Two Stage Visual Representation Learning Module and Multi-way Attention Module.

At each time step, HAMT adopts the instruction $\boldsymbol{I}$, current observations $\boldsymbol{O}_t$, and historical observations $\boldsymbol{H}_t$ to handle environment information. We augment the original RGB observations using the two-stage visual representation learning module. This involves using local-aware slot attention and multimodal fusion to act upon the current candidate observations, while the other observations, both historical and current, are only augmented with multimodal fusion. As a result, we obtain the geometrically enhanced visual representations $\hat{\boldsymbol{F}}_t$ and $\hat{\boldsymbol{H}}_t$ for current observations $\boldsymbol{O}_t$ and historical observations $\boldsymbol{H}_t$, respectively. The backbone network of HAMT, which includes the unimodal and cross-modal transformer encoders, is then used to obtain the embeddings $\boldsymbol{I}', \hat{\boldsymbol{H}}'_t, \hat{\boldsymbol{F}}'_t$, which can be formulated as:

$$\boldsymbol{I}', \hat{\boldsymbol{H}}'_t, \hat{\boldsymbol{F}}'_t = \mathrm{HAMT}\left(\boldsymbol{I}, \hat{\boldsymbol{H}}_t, \hat{\boldsymbol{F}}_t\right). \tag{1}$$

Since HAMT does not maintain a sufficiently information-rich state vector $\boldsymbol{s}_t$ as RecBERT does, we slightly adapt the MAtt module to fit HAMT's original network framework. Specifically, we use the [CLS] token of the instruction to obtain a state vector, but first multiply the [CLS] token and the features $\hat{\boldsymbol{F}}'_t$ of each observation before calculating a matching score for each modality in the same way as the main text. The reformulated $\boldsymbol{s}_t$ can be written as:

$$\boldsymbol{s}_t = \boldsymbol{I}'_{cls} \odot \hat{\boldsymbol{F}}'_t, \tag{2}$$

where $\odot$ is element-wise multiplication, $\boldsymbol{I}'_{cls}$ is the embedding of [CLS] token of the instruction. This process is consistent with the decision-making module in HAMT and also compensates for the visual information that is difficult to include in the $\boldsymbol{I}'_{cls}$.

2. More Training Details

We train the network with RecBERT as the backbone for 100,000 iterations on a single GeForce GTX TITAN X GPU, and the one with HAMT as the backbone for 200,000 iterations on a single GeForce RTX 3090 GPU. The batch-size is set to 8. The optimizer is AdamW and a cosine annealing scheduler with warmup is used to adjust the learning rate. We set the arguments of the cosine annealing scheduler as follows: first cycle step size is 50, cycle steps magnification is 1, max learning rate is 10^{-5}, min learning rate is 5×10^{-8}, decrease rate of max learning rate by cycle is 0.1. The scheduler is employed to adjust the learning rate at an interval of 2000 iterations.

3. Computation Cost

We compare the parameters, inference time, and memory usage of our extended models with those of the original models. The inference time is measured as the single-run time on the val unseen split. Although our extended models introduce additional computation cost, the cost is manageable compared to the improvement achieved. Importantly, our GeoVLN model achieves competitive performance with the original HAMT model on the val unseen split while incurring lower computational cost.

Model	Params (M)	Memory (GiB)	Inference Time (s)
RecBERT	153	6.0	70
GeoVLN	159	7.8	108
HAMT	163	9.9	134
GeoVLN[†]	174	15.1	191

Table 1. Comparison of the computation cost. GeoVLN uses RecBERT as the backbone, while GeoVLN[†] uses HAMT as the backbone.

[*] Equal contributions.

[†] Corresponding authors.

Yanwei Fu is with School of Data Science, Fudan University, Shanghai Key Lab of Intelligent Information Processing, and Fudan ISTBI–ZJNU Algorithm Centre for Brain-inspired Intelligence, Zhejiang Normal University, Jinhua, China.

4. Structure Variants of Two-Stage Module

In this paper, we introduce a two-stage module to handle the multimodal observations including RGB, depth and normal surface. To further explore the better architecture of the two-stage model, we design three variants (Figs. 1b to 1d) of it and compare them with the original one (Fig. 1a).

The two-stage model (TwoSM) combines a local-aware slot attention module and a vision encoder to acquire a geometry-enhanced visual representation $\hat{\boldsymbol{F}}_t$. The pipeline of the original one, which is adopted in our article, is shown in Fig. 1a. We firstly aggregate RGB features with local-aware slot attention and obtain the slot enhanced RGB features $\hat{\boldsymbol{F}}_t^{rgb}$. And then, the enhanced features $\hat{\boldsymbol{F}}_t^{rgb}$ together with the depth features and normal surface features are fed into the vision encoder, in which they are concatenated and projected to the geometry-enhanced visual representations $\hat{\boldsymbol{F}}_t$. Based on this method, three variants, named TwoSM-1, TwoSM-2 and TwoSM-3, are proposed. As shown in Fig. 1b, TwoSM-1 aggregate the RGB, depth and surface normal information respectively by a slot attention module, and then feeds the features $\hat{\boldsymbol{F}}_t^{rgb}$, $\hat{\boldsymbol{F}}_t^{dep}$, $\hat{\boldsymbol{F}}_t^{nor}$ to the vision encoder, performing the same operations as the original one. Moreover, we design a RGB-guided slot attention module for the second and third variants. In this module, slots are initialized with only RGB candidate features to dominate the fusion process, while multimodal features of the nearby observations are treated as both keys and values. Such a method enables the depth and surface normal information from nearby observations to directly update the slots (*i.e.* the RGB candidate features) as well. TwoSM-2 (Fig. 1c) and TwoSM-3 (Fig. 1d) differ only in their visual encoders. TwoSM-2 takes only the enhanced RGB information $\hat{\boldsymbol{F}}_t^{rgb}$ as input, while TwoSM-3 additionally adding the original depth and normal information as input.

We quantitatively compare the three variants with the original TwoSM in Tab. 2. It can be seen that all variants perform weaker than our original TwoSM. The best performer among the variants is TwoSM-1, but with a 1.5% reduction in SPL and 3.0% reduction in SR on the val-unseen split. We suggest that this could indicate that the local information aggregation based on slot attention in depth and normal features instead interferes with the decision making of the agents. TwoSM-2 and TwoSM-3 additionally add depth and surface normal information to update RGB candidate features in slot module, but present worse performance. This suggests that the simultaneous aggregation of nearby observations and depth/normal maps presents a significant challenge. In particular, there exist certain features that do not correspond spatially or modally with the RGB-initialized slots, making it extremely challenging to align them with the slots. In addition, TwoSM-3 outperforms TwoSM-2, suggesting that adding raw depth and normal information to the visual encoder contributes to the model. The above results illustrate the rationality and superiority of our two-stage model.

	Val Seen		Val Unseen	
Model	SR↑	SPL↑	SR↑	SPL↑
TwoSM	**69.64**	**64.86**	**66.75**	**61.00**
TwoSM-1	65.72	62.23	64.75	60.06
TwoSM-2	67.29	63.27	62.15	56.63
TwoSM-3	67.19	63.14	64.75	59.46

Table 2. Ablation study on the structure variants of the two-stage model.

5. Supplementary Results

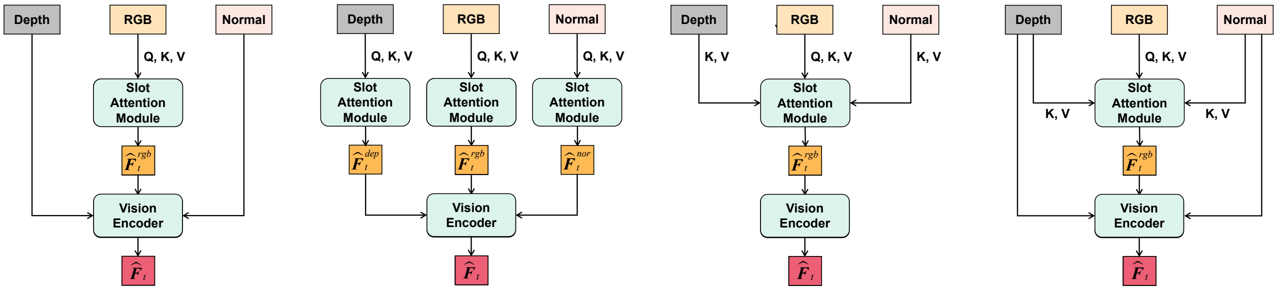
In this section, more results of the navigation path are given visually. Moreover, additional qualitative or quantitative results are given to illustrate the effectiveness of our local-aware slot attention module and multiway attention module.

5.1. Visualization for Local-aware Slot Attention

We show how the local-aware slot attention module aggregates local observations to candidate views in Fig. 2. The left side of the figure shows a panoramic view, with the candidate view selected by our agent marked in red. The observations used to update that candidate view and the corresponding attention weights are labeled on the right side. Note that the updates obtained from the attention weights need to be added to the original candidate view. Thus, even if the candidate view does not have the highest attention score, the final candidate information still maintains the highest relevance to the candidate view itself.

Fig. 3 shows an case of our GeoVLN successfully navigating while the Recurrent VLN-BERT does not in val-unseen set. The red box marks the candidate view we select (the successful one) and the blue box marks the candidate view that the Recurrent VLN-BERT selects incorrectly the first time. At the time steps $t = 0 \sim 3$, our local-aware slot attention module gives high attention weights to the regions containing the bed (even greater than 0.9). Therefore, our model finally succeeds in capturing the relationship between the bed and the candidate views and completes the instruction “Go around the bed and to the right”, while the Recurrent VLN-BERT fails.

Figs. 4 and 5 show additional cases of the success of our model in the val-unseen spit, with the attention weights for local observations marked on the right. The piano and the wooden door are given more attention in Fig. 4 and the hall is given more attention in Fig. 5, which are consistent with the instructions.



(a) TwoSM

(b) TwoSM-1

(c) TwoSM-2

(d) TwoSM-3

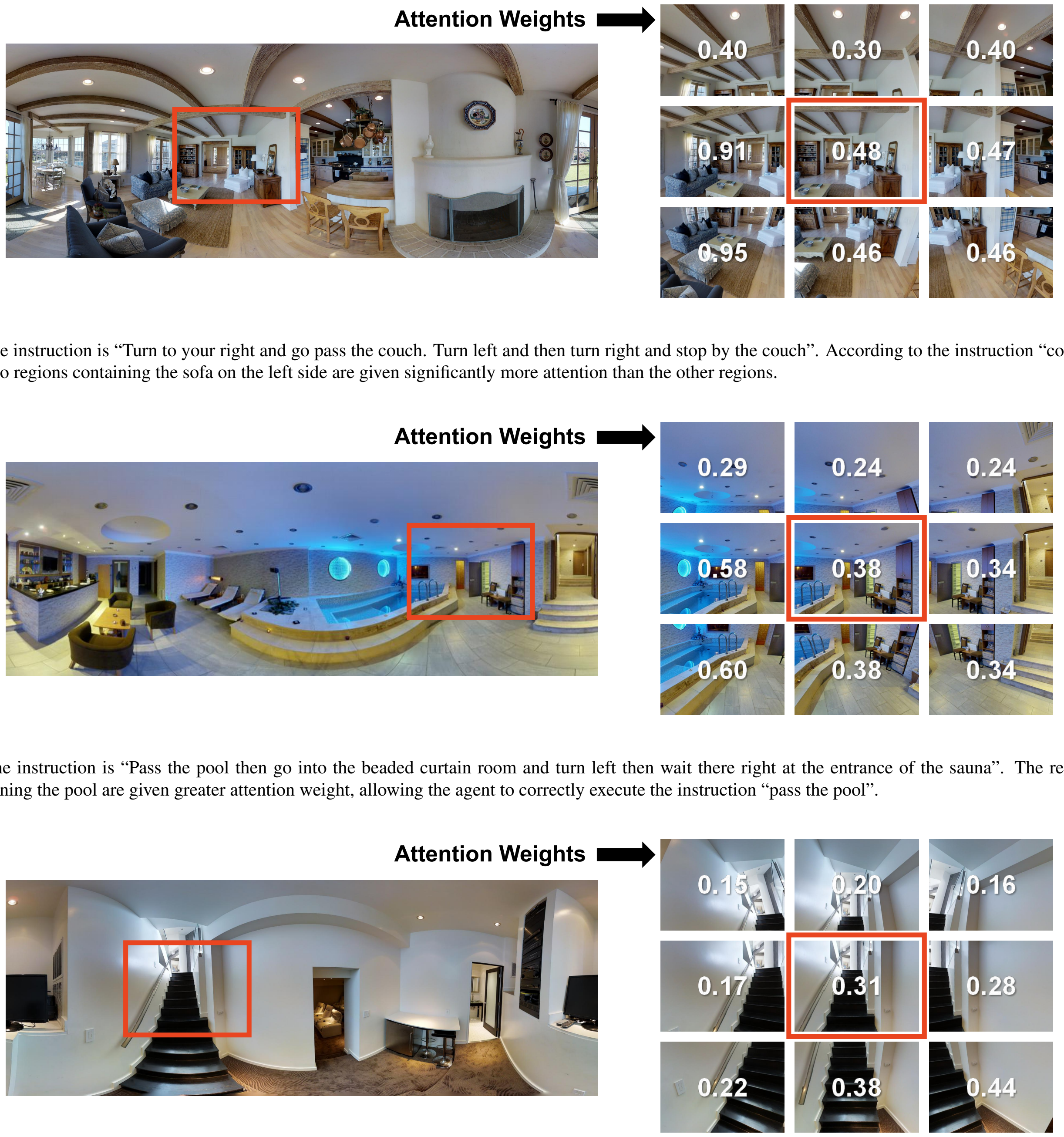
Figure 1. Pipelines of the Two-Stage Model and its structure variants.

5.2. Visualization for Multiway Attention

In the multiway attention module, the final matching score is obtained by dynamically weighting the matching scores of the three modalities of visual inputs (RGB, depth and normal images) and the instructions. Fig. 6 shows how the weight coefficients of the three modalities (w_t^{rgb}, w_t^{dep}, w_t^{nor}) change in a single successful navigation. In this case, the depth information rather than the RGB information is given the highest weight coefficient as the instruction contains "straight through the room". At initialization ($t = 0$), our model correctly selects the candidate view containing the "double doors" (marked by the red box), while the Recurrent VLN-BERT incorrectly selects the view containing the window (marked by the blue box). This is attributed to the fact that the depth map contains depth information about the room behind the door, making the agent aware of the door. In addition, at the time step $t = 5$, the weight coefficient w_t^{rgb} increases to 0.42 and w_t^{dep} decreases to 0.55. This is consistent with the instruction to "wait in the doorway with the double doors at the end", which requires higher attention to the RGB images to confirm the appearance of the door.

5.3. Failure Cases

Examples of navigation failures of GeoVLN are shown in Figs. 7 and 8, which demonstrate that our model still has limitations. More accurate and effective models for vision-and-language navigation are expected to be proposed.



(a) The instruction is "Turn to your right and go pass the couch. Turn left and then turn right and stop by the couch". According to the instruction "couch", the two regions containing the sofa on the left side are given significantly more attention than the other regions.

(b) The instruction is "Pass the pool then go into the beaded curtain room and turn left then wait there right at the entrance of the sauna". The regions containing the pool are given greater attention weight, allowing the agent to correctly execute the instruction "pass the pool".

(c) The instruction is "Go up the stairs then turn right and stand near the bed". The views containing stairs (bottom two rows) are given higher attention weights than the views with almost no stairs (top row).

Figure 2. Visualizations of the attention weights in local-aware slot attention module.

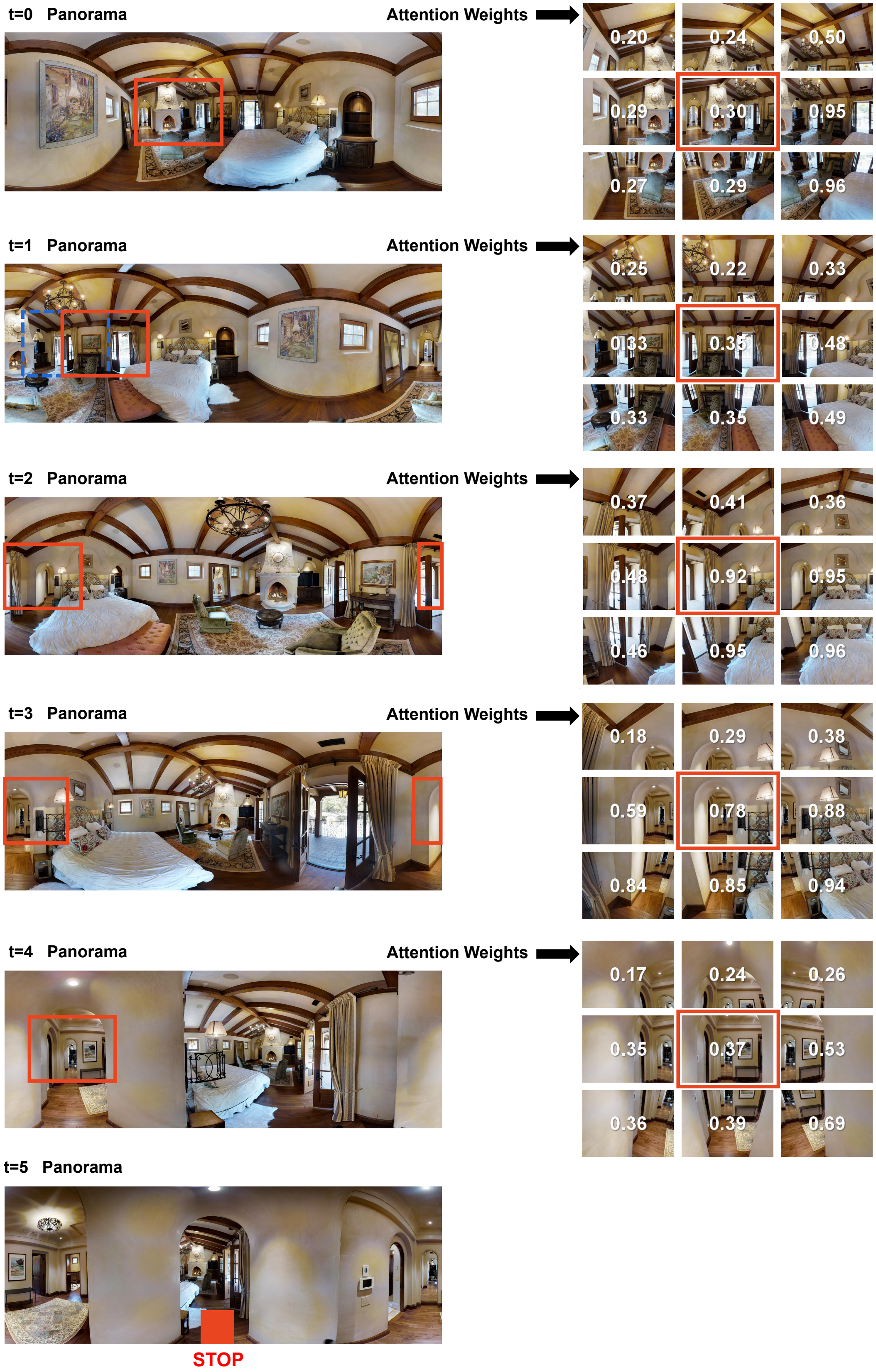


Figure 3. An example of success on the val unseen split. The instruction is “Go around the bed and to the right. Go through the arch opening and wait near the thermostat.”

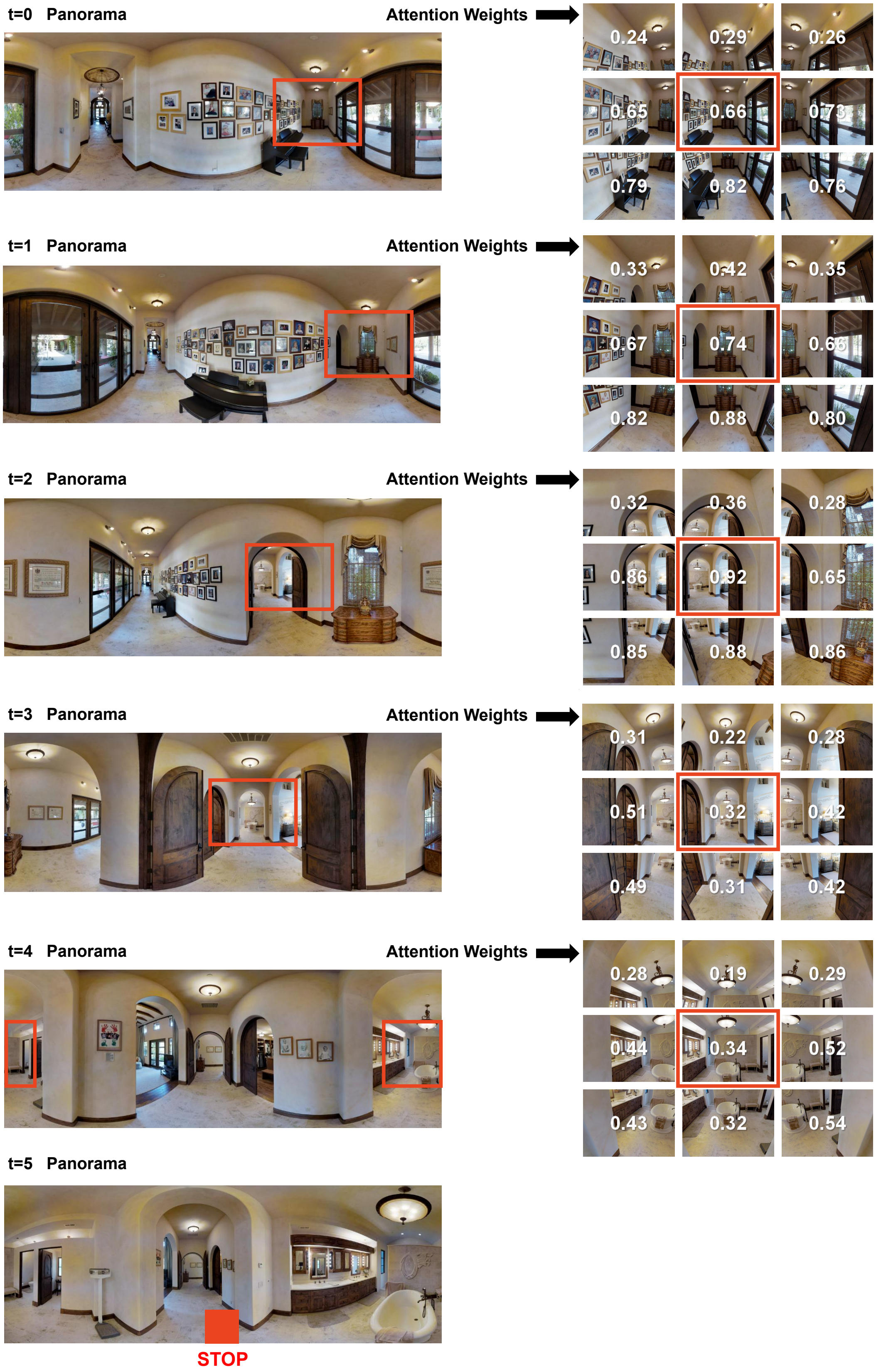


Figure 4. An example of success on the val unseen split. The instruction is "Walk past piano. Walk through arched wooden doors. Wait at bathtub."

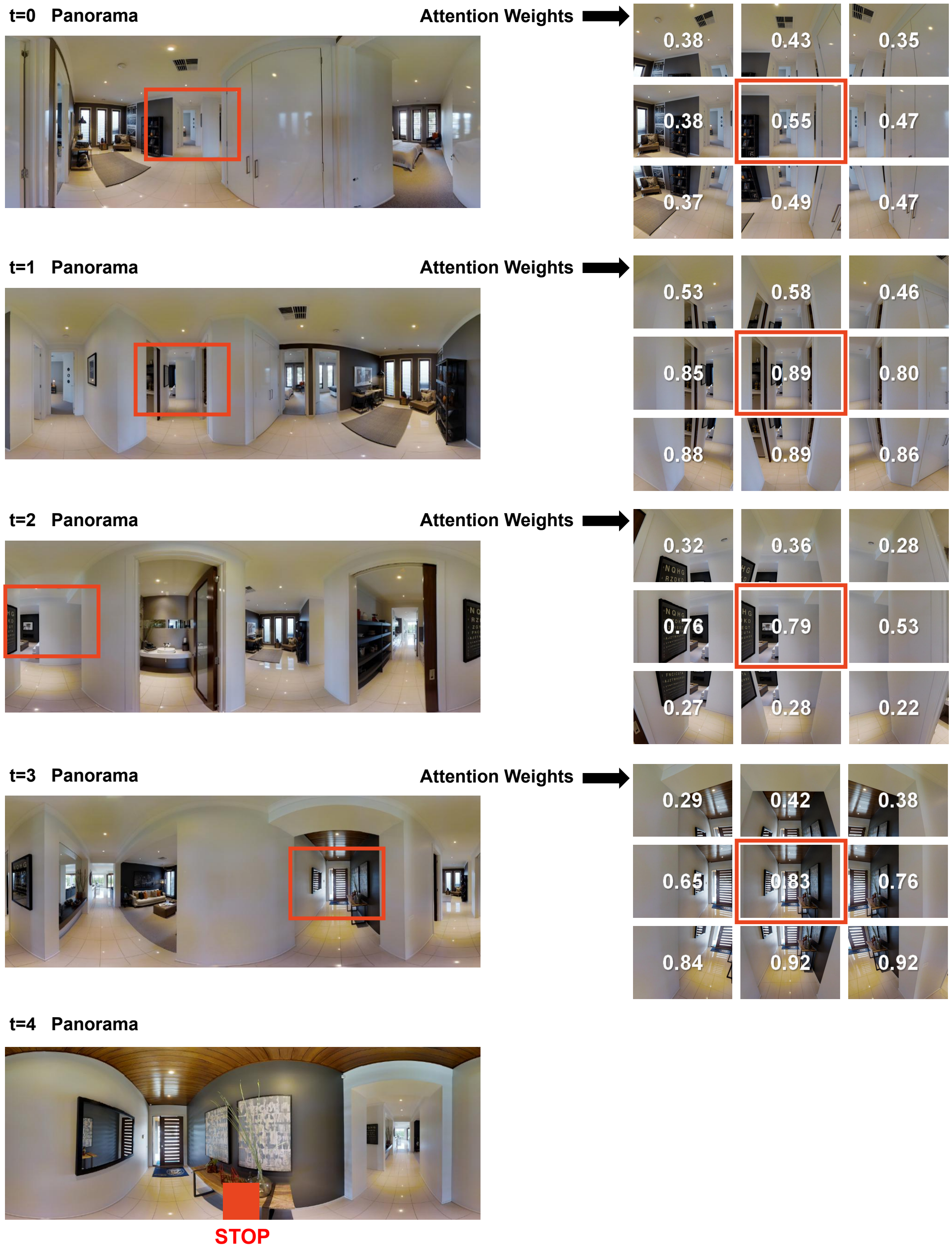


Figure 5. An example of success on the val unseen split. The instruction is "Leave the bedroom and take the first right into a hallway. Take a right at the end of the hall and enter the foyer. Stop before you reach the mirror."

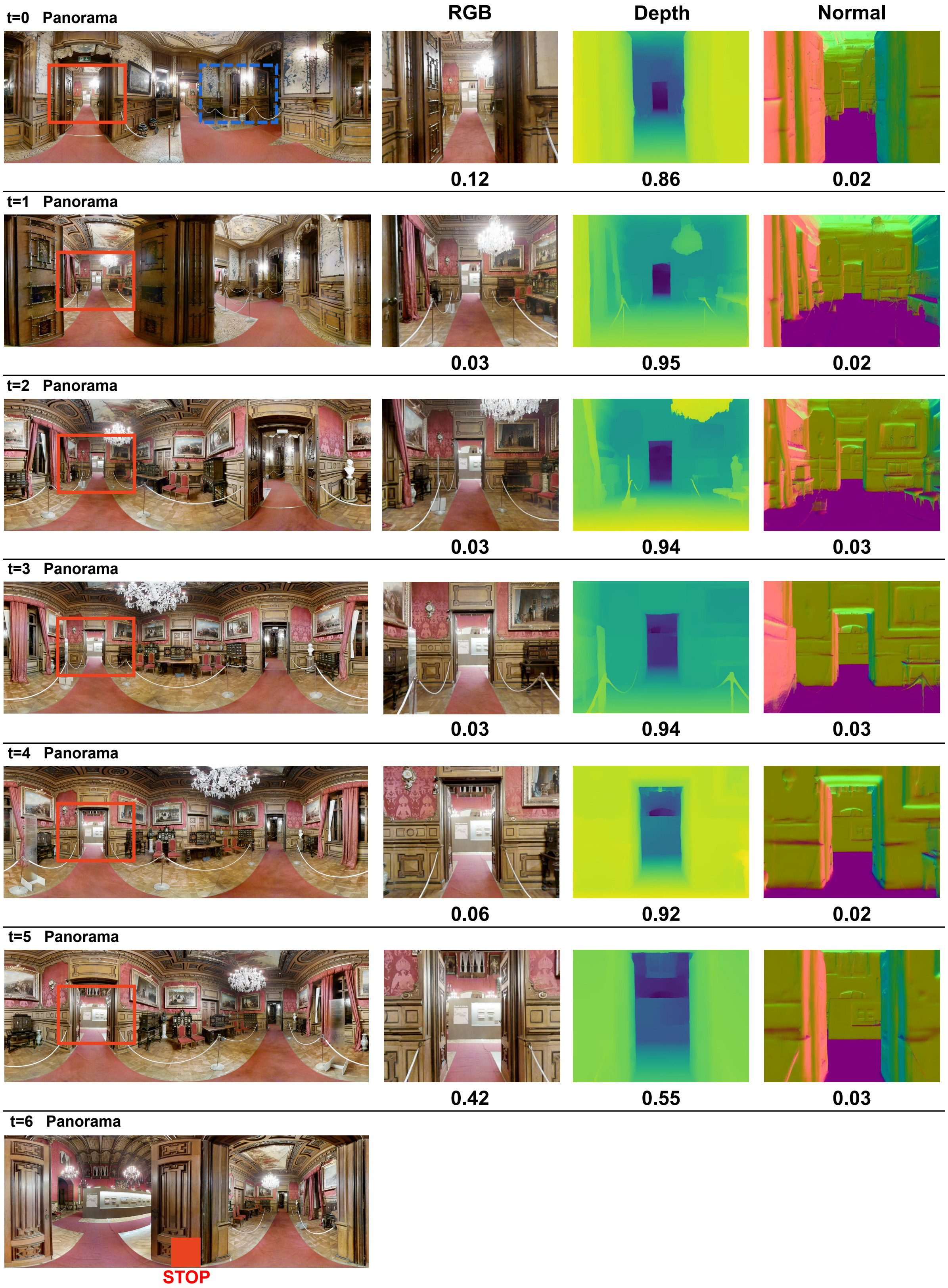
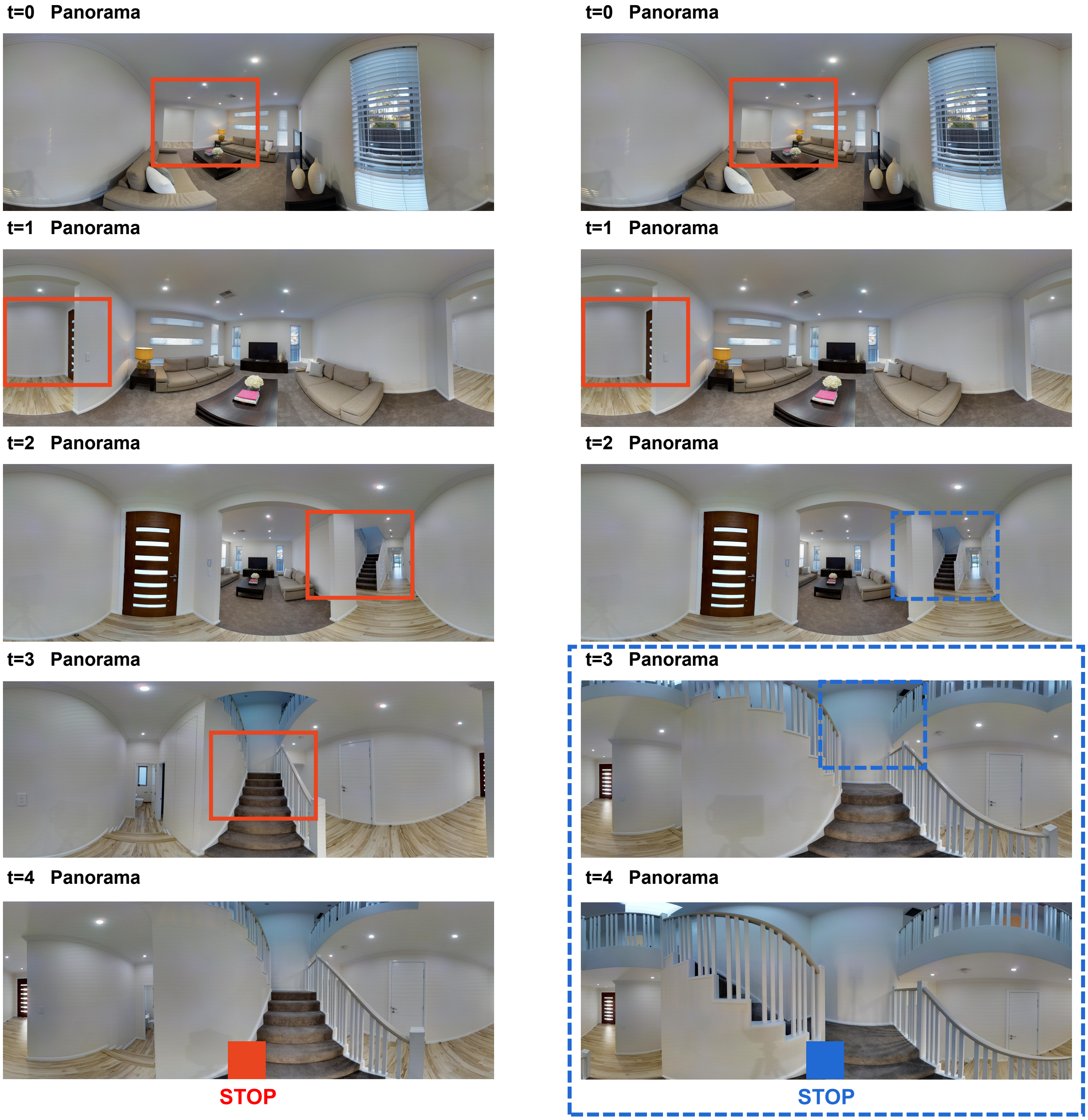


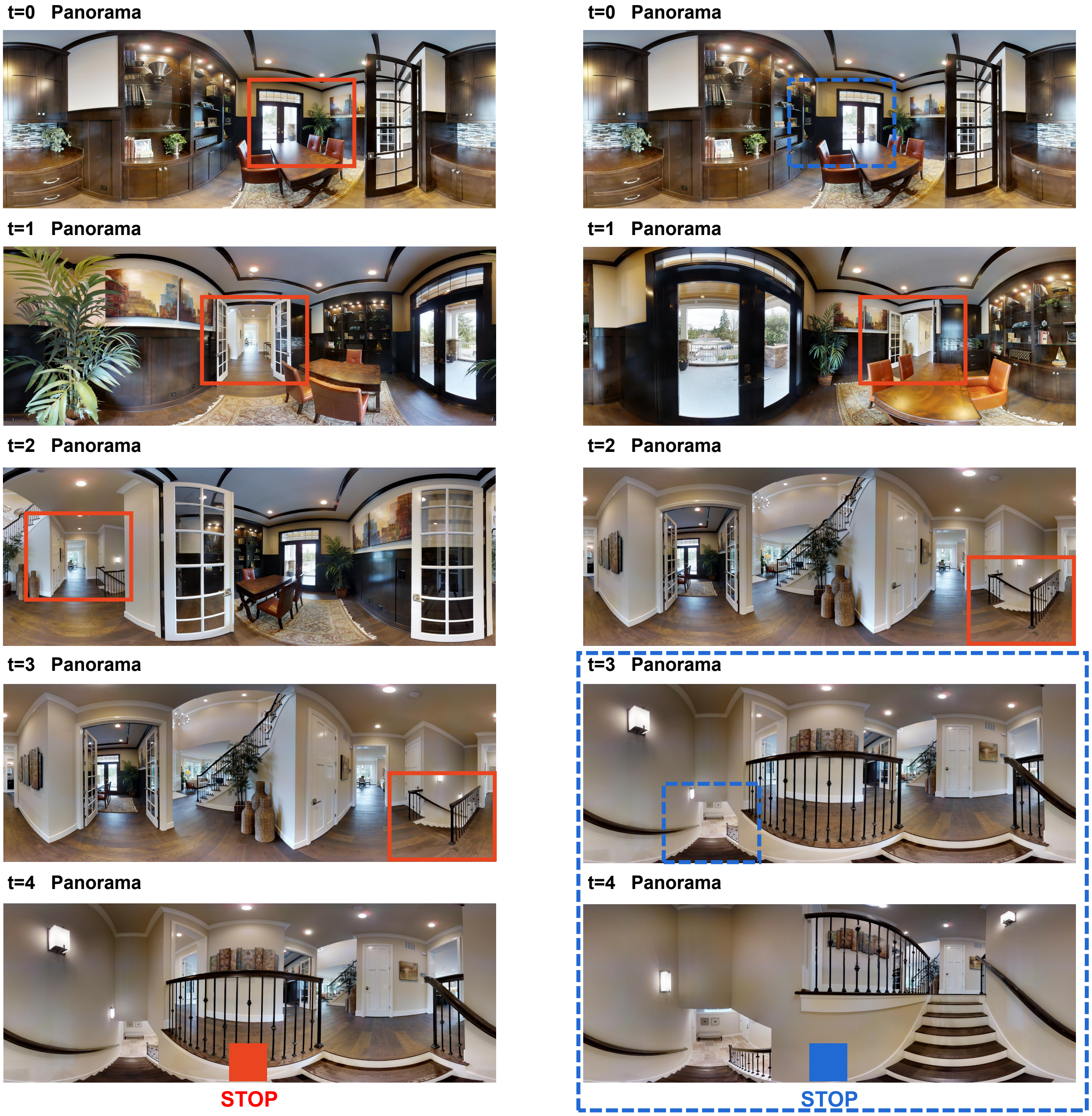
Figure 6. An example of success on the val unseen split. The instruction is "Follow the red carpet through the double doors. Continue straight through the room and wait in the doorway with the double doors at the end." The red box marks the choice of GeoVLN and the blue box marks the wrong choice of the Recurrent VLN-Bert. The numbers below the RGB, depth, and surface normal images indicate the weight coefficients, including w_t^{rgb}, w_t^{dep} and w_t^{nor}, given by the multiway attention module, respectively.



(a) Ground truth trajectory.

(b) Predicted trajectory.

Figure 7. An example of failure to navigate on the val unseen split. The instruction is “Walk through the bedroom and out into the hall way. Turn left and walk up to the stairs. Walk up to the first step and stop”. The blue boxes indicate incorrect selections. Our GeoVLN fails to correctly perform the instruction ”Walk up to the first step and stop” and instead continues up the stairs.



(a) Ground truth trajectory.

(b) Predicted trajectory.

Figure 8. An example of failure to navigate on the val unseen split. The instruction is “Leave the room by exiting through the open double doors. Go down the stairs and stop on the second step from the top and wait there”. The blue boxes indicate incorrect selections. At initialization ($t = 0$), the direction chosen by GeoVLN is deviated, but subsequently this mistake is rectified ($t = 1, 2$). However, at the end, the agent does not stop correctly on the stairs, but continues downward instead.