

# Object-and-Action Aware Model for Visual Language Navigation

Yuankai Qi<sup>1</sup>, Zizheng Pan<sup>1</sup>, Shengping Zhang<sup>2,3</sup>,  
Anton van den Hengel<sup>1</sup>, and Qi Wu<sup>\*1</sup>

<sup>1</sup> The University of Adelaide

<sup>2</sup> Harbin Institute of Technology, Weihai

<sup>3</sup> Aipixel Inc.

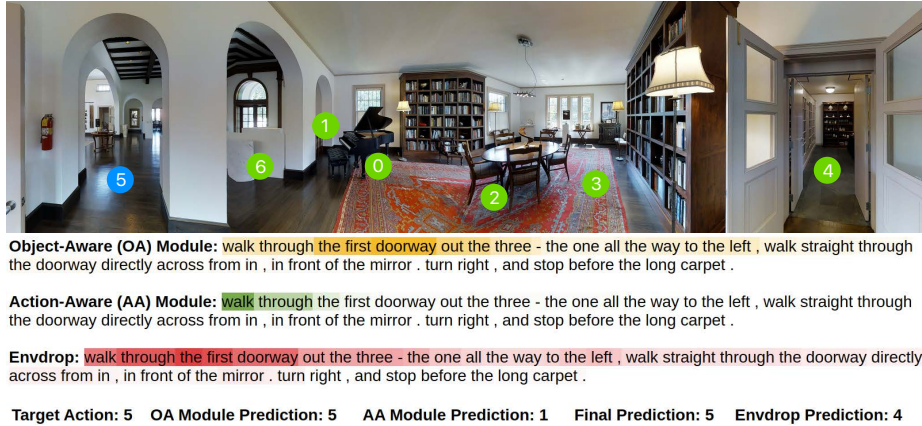
qykshr@gmail.com, zizhpan@gmail.com, s.zhang@hit.edu.cn  
{anton.vandenhengel, qi.wu01}@adelaide.edu.au

**Abstract.** Vision-and-Language Navigation (VLN) is unique in that it requires turning relatively general natural-language instructions into robot agent actions, on the basis of visible environments. This requires to extract value from two very different types of natural-language information. The first is object description (e.g., ‘table’, ‘door’), each presenting as a tip for the agent to determine the next action by finding the item visible in the environment, and the second is action specification (e.g., ‘go straight’, ‘turn left’) which allows the robot to directly predict the next movements without relying on visual perceptions. However, most existing methods pay few attention to distinguish these information from each other during instruction encoding and mix together the matching between textual object/action encoding and visual perception/orientation features of candidate viewpoints. In this paper, we propose an Object-and-Action Aware Model (OAAM) that processes these two different forms of natural language based instruction separately. This enables each process to match object-centered/action-centered instruction to their own counterpart visual perception/action orientation flexibly. However, one side-issue caused by above solution is that an object mentioned in instructions may be observed in the direction of two or more candidate viewpoints, thus the OAAM may not predict the viewpoint on the shortest path as the next action. To handle this problem, we design a simple but effective path loss to penalize trajectories deviating from the ground truth path. Experimental results demonstrate the effectiveness of the proposed model and path loss, and the superiority of their combination with a 50% SPL score on the R2R dataset and a 40% CLS score on the R4R dataset in unseen environments, outperforming the previous state-of-the-art.

**Keywords:** Vision-and-Language Navigation, Modular Network, Reward Shaping

---

<sup>\*</sup>corresponding author



**Fig. 1.** A snapshot of navigating action predicted by the proposed model and our baseline navigator EnvDrop [19]. Our model is able to more flexibly utilize object and action phrases thanks to the object-/action- aware modules. The numbered circles denote navigable viewpoints, and the blue one is our final decision.

## 1 Introduction

Vision-and-language navigation (VLN) has attracted increasing attention, partly due to the prospect that the capability to interpret general spoken instructions might represent a significant step toward enabling intelligent robots [2,4,11,18]. The goal of VLN is for a robot to interpret a navigation instruction expressed in natural language, and carry out the associated actions. Typically this involves an agent navigating through a 3D photo-realistic simulator [2] to a goal location, which is not visible from the start point. The natural language instructions involved are of the form of ‘Go left down the hallway toward the exit sign until the end. Turn right and go down the hallway. Go into the door on the left and stop by the table.’. The simulator in this scenario provides a set of panoramic images of a real environment, and a limited set of navigation choices that can be taken for each. Although it is related to other Vision-and-Language problems that have been extensively studied, VLN remains an open problem due to the difficulty of navigating general real environments, and the complexity of the instructions that humans give for doing so.

A range of approaches have been proposed to address the vision-and-language guided navigation problem [17,4,19,18,16,22,21,5]. For instance, to progressively process long navigation instructions, Ma *et al.* [17] propose a progress monitor network to identify the completed part of an instruction and the part associated with the next action. To expand training data, the Speaker-Follower (SF) [4] and EnvDrop [19] are proposed to generate new trajectories and ‘unseen’ scenes from seen ones, respectively. To mimic human behaviors when navigating, the Regretful model [18] introduces a backtracking mechanism into the navigation process to enable the agent to retrace its steps. In FAST model [14], a strategy

that compromises between greedy search and beam search is developed to balance search efficiency against accuracy. However, all of these approaches entangle the encoding of object descriptions with that of action specifications, instead of processing them separately. Most action specifications (e.g., ‘go straight ahead’ or ‘turn right’), closely relating to orientations<sup>1</sup> rather than visual perceptions of each navigable candidate viewpoint, are able to directly lead to the correct next action. By contrast, the object descriptions (e.g., ‘the stairs’ or ‘the table’) correspond to visual perceptions, instead of the orientations, in the visible scene. Therefore, the mixed encoding scheme may limit the similarity/grounding learning in the decoding phase between instructions and visual perceptions as well as orientations.

To address the above mentioned problem, we propose here an object-and-action aware model (OAAM) for robust navigation that reflects to exploit the important distinction. Specifically, we first utilize two learnable attention modules to highlight language relating to objects and actions within a given instruction. Then, the resulting attention maps guide an object-vision matching module and an action-orientation matching module, respectively. Finally, the action to be taken is identified by combining the outcomes of both modules using weights derived by an additional attention module taking the instruction and visual context as input. Figure 1 provides an example of the VLN process that demonstrates that our design enables the agent to more flexibly utilize object descriptions and action specifications, and finally leads to the correct prediction compared against the strong baseline navigator EnvDrop [19].

The proposed OAAM is able to improve the navigation success as demonstrated later in the experiment. However, its trajectory might not be the shortest because the focused object in the instruction may be observed in the direction of multiple candidate viewpoints. To handle this problem, we design a simple but effective path loss to encourage the agent to stay on the shortest path instead of merely picking alternative viewpoints containing the instruction mentioned object. In particular, the proposed path loss is based on distances calculated at each agent step to its nearest viewpoint on the ground-truth path. Note that this differs from the Coverage weighted by Length Score (CLS) award [13] that computes a normalised score by inversely finding the nearest viewpoint in the agent trajectory to each viewpoint on the ground-truth path. Experimental results show that this loss aids the OAAM to generalise in unseen scenarios.

The main contributions of this work are summarized as follows:

1. We propose an object-and-action aware model, which better reflects the nature of the associated natural language instructions and responds more flexibly.
2. We design a path loss that encourages the agent to closely follow navigation instructions.

---

<sup>1</sup>‘orientation’ means the encoding of the angle that an agent should rotate in order to find a navigable viewpoint from its front direction as in [14,17,18].

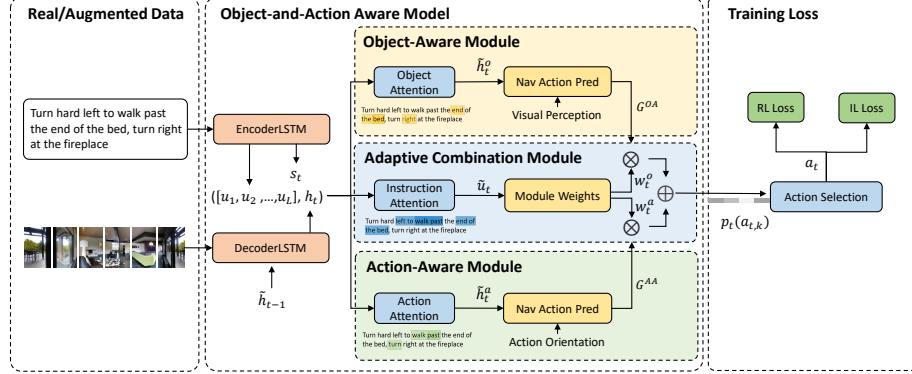
3. Extensive experimental results demonstrate the effectiveness of the proposed method and set new state-of-the-art on the R4R dataset with a CLS score of 0.40.

## 2 Related Work

**Vision-and-Language Navigation.** Numerous methods have been proposed to address the VLN problem. Most of them employ the CNN-LSTM architecture with attention mechanism to first encode instructions and then decode the embedding to a series of actions. Together with the proposing of the VLN task, Anderson *et al.* [2] develop the teacher-forcing and student-forcing training strategy. The former equips the network with basic ability for navigating using ground truth action at each step, while the latter mimics the test scenarios, where the agent may predict wrong actions, by sampling actions instead of using ground truth actions. Deep learning based methods always benefit from massive training data. To generate more training data, Fried *et al.* [4] propose a speaker model to synthesize new instructions for randomly selected robot trajectories. By contrast, Tan *et al.* [19] augment the training data by additionally removing objects from the original training data to generate new ‘unseen’ environments. To further enhance the generalization ability, they also propose to train the model using both imitation learning and reinforcement learning so as to take advantage of both off-policy and on-policy optimization. Not all chunks of an instruction are useful to predict the next action. Ma *et al.* [17] propose a progress monitor to locate the completed sub-instructions and to be aware of the ones for predicting the next action. Another way to improve navigation success is to equip the agent with backtracking. In [18], Ma *et al.* propose to treat the navigation as a graph search problem and predict to move forward or roll back to a previous viewpoint. In [14], each passed viewpoint is viewed as a goal candidate, and the final decision is the viewpoint with the highest local and global matching score.

However, the above mentioned methods neglect to distinguish the object descriptions and the action specifications within the focused sub-instruction. This may limit the learning of matching between object-/action-centered instructions and their counterpart visual perceptions/orientations. To address this problem, we propose to separately learn the object attention and action attention, and so the learning of object-vision matching and action-orientation matching.

**Modular Language Process Networks.** Modular language decomposition has been widely adopted in question answering [3], visual reasoning [7], referring expression comprehension [9,23], etc. Most methods are based on hard parsers or learned parsers. For example, Andreas *et al.* [3] decompose each question into a sequence of modular sub-problems with the help of an fixed off-the-shelf syntactic parsers. Such hard parsers are not able to analyse semantic relationships between entities. By contrast, the learned parses can be flexibly designed according to parsing purposes. For instance, Hu *et al.* [9] design a attention-based network to learn to decompose expressions into (Subject, Relationship, Object) triplets, which facilitates to ground language to visual regions. In [23], Yu *et*



**Fig. 2.** Main steps of the proposed object-and-action aware model, which consists of three modules: the object-aware module that predicts the next action based on object-centered instructions and visual perceptions; the action-aware module that predicts the next action based on action-centered instructions and orientations of candidate viewpoints; and the adaptive combination module that predicts weights to combine predictions obtained by the other two modules.

*al.* propose the (Subject, Location, Relationship) triplet to additionally taking object position into consideration.

Inspired by the above modular decomposition mechanism, we propose to decompose an instruction into object and action components for the VLN task because two fundamental different type of information of candidate viewpoints are used to match to instructions, namely visual perceptions and orientations. Differ from the above mentioned methods, our modular decomposition model takes navigation progress into consideration, i.e., the decomposition should take place on a certain part of the instruction, which is closely related to the current agent location, instead of the whole instruction.

### 3 Method

In this section, we first describe the symbols that will be used later and briefly introduce the baseline navigator, EnvDrop [19], to put our method into a proper context. EnvDrop is adopted due to its good performance. Then, we detail the proposed object-and-action aware model (OAAM).

Figure 2 shows the pipeline of the proposed method in training phase. An instruction is first encoded by a bi-direction LSTM. Then our OAAM model decomposes the instruction encoding into object- and action-centered representations using two attention-based modules. These representations are further used to predict navigation actions via matching to visual perceptions and orientations, respectively. The OAAM model also predicts combination weights to compute the final navigation action probability distribution over candidate viewpoints. Lastly, imitation learning and reinforcement learning losses are calculated

based on the probability distribution and a proposed path loss. In the inference phase, the navigation action with maximum probability is selected.

### 3.1 Problem Setup

Given a natural language instruction  $I = \{w_1, w_2, \dots, w_L\}$  with  $L$  words, at each step  $t$ , the agent observes a panoramic view  $o_t$ , which contains 36 discrete single views  $\{o_t\}_{i=1}^{36}$ . Each view  $o_{t,i}$  is represented by an image  $v_{t,i}$  at an orientation with heading angle  $\theta_{t,i}$  and elevation angle  $\phi_{t,i}$ . At each step  $t$ , there are  $N_t$  navigable viewpoints  $\{P_{t,k}\}_{k=1}^{N_t}$ . The agent needs to select one viewpoint  $P_{t,k}$  as the next navigate action  $a_t$ . Following the common practice, each navigable viewpoint has an orientation feature  $n_{t,k} = (\cos\theta_{t,k}, \sin\theta_{t,k}, \cos\phi_{t,k}, \sin\phi_{t,k})$  and a visual feature  $m_{t,k} = \text{ResNet}(v_{t,k})$ .

### 3.2 Baseline-EnvDrop

Recent works show that data augmentation is able to significantly improve the generalization ability in unseen environment [4, 19]. We make use of this strategy by adopting EnvDrop [19] as our baseline, which simultaneously benefits from the back translation technique [4] to generate new (trajectory, instruction) pairs in existing environments and from its global dropout technique to generate new environments.

EnvDrop first uses a bidirectional LSTM to encode instruction  $I$ :

$$[u_1, u_2, \dots, u_L] = \text{Bi-LSTM}(e_1, \dots, e_L) \quad (1)$$

where  $e_j = \text{embedding}(w_j)$  is the embedded  $j$ -th word in the instruction, and  $u_j$  is the feature containing context information for the  $j$ -th word. Then, a soft attention is imposed on visual feature  $m_t$  to get attentive feature  $\tilde{f}_t = \sum_i \alpha_{t,i} m_{t,i}$ , where  $\alpha_{t,i} = \text{softmax}_i(m_{t,i}^\top W_F \tilde{h}_{t-1})$ . The concatenation of  $\tilde{f}_t$  and the previous action embedding  $\tilde{a}_{t-1}$  is fed into the decoder LSTM:

$$h_t = \text{LSTM\_Decoder}([\tilde{f}_t; \tilde{a}_{t-1}], \tilde{h}_{t-1}). \quad (2)$$

The decoder input also includes the previous instruction-aware hidden output  $\tilde{h}_{t-1}$ , which is updated based on the attentive instruction feature  $\tilde{u}_t$  and the newly obtained  $h_t$ :

$$\tilde{h}_t = \tanh(W_L[\tilde{u}_t; h_t]), \quad (3)$$

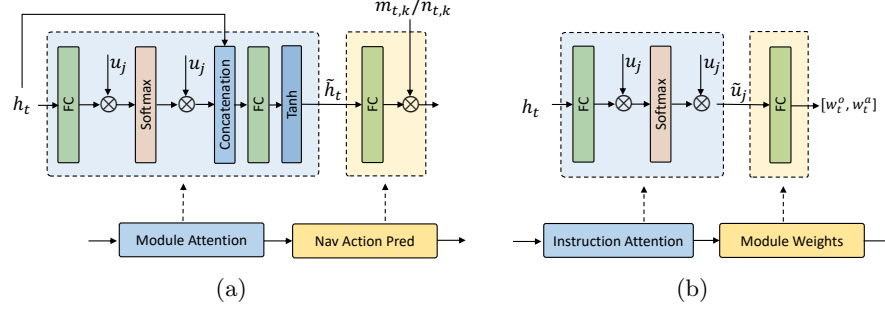
$$\tilde{u}_t = \sum_j \beta_{t,j} u_j, \quad (4)$$

where  $\beta_{t,j} = \text{softmax}_j(u_j^\top W_I h_t)$ , and  $W_L$  is trainable parameter. Finally, EnvDrop predicts navigation action by selecting a navigable view with the biggest probability

$$a_t^* = \arg \max_k P(a_{t,k}), \quad (5)$$

where  $P(a_{t,k}) = \text{softmax}_k([m_{t,k}; n_{t,k}]^\top W_B \tilde{h}_t)$ .

Different from EnvDrop, our navigator calculates the probability  $P(a_{t,k})$  using the proposed object-and-action aware model as detailed in the next section.



**Fig. 3.** Pipeline and configuration of the proposed object-/action-aware module (a) and the adaptive combination module (b).

### 3.3 Object-and-Action Aware Model

To disentangle the encoding of object- and action-related instruction, as well as the matching to their counterpart visual perceptions and orientations of candidate viewpoints, we propose the object-and-action aware model (OAAM). OAAM consists of three key modules: an object-aware (OA) module that is aware of object-related instruction and predicts action using visual perceptions; an action-aware (AA) module that is aware of action-related instruction and predicts action using orientations of candidate viewpoints; and an adaptive combination module that learns weights to combine the action predictions from the other two modules based on the attentive instructions with the consideration of the current panoramic views.

**Object-Aware (OA) Module** Figure 3(a) shows the pipeline and configuration of the proposed OA module. Different from the subject/object module in referring expression comprehension methods, such as [10, 23], which highlight all objects in an expression, our OA module is designed to highlight object phrase *just for the next navigation step* instead of *all objects* in an instruction. This is essential for the VLN task because the long instruction should be carried out step-by-step, which indicates objects not related to the current step may be noise and mislead the action predicting. To this end, our OA module calculates the attentive object feature  $\tilde{u}_t^o$  by taking the decoder hidden state  $h_t$  as input for navigation progress reference:

$$\gamma_{t,j} = \text{softmax}_j(u_j^\top W_O h_t), \quad (6)$$

$$\tilde{u}_t^o = \sum_j \gamma_{t,j} u_j, \quad (7)$$

where  $W_O$  is learnable parameters. Then, the object-aware hidden output  $\tilde{h}_t^o$  related to the current navigation is computed via

$$\tilde{h}_t^o = \tanh(W_P[\tilde{u}_t^o; h_t]), \quad (8)$$

where  $W_P$  is parameter to be learned. Lastly, the action confidence  $G^{OA}(a_{t,k})$  from the object-aware side is obtained by using only the visual feature  $m_{t,k}$ :

$$G^{OA}(a_{t,k}) = m_{t,k}^\top W_H \tilde{h}_t^o, \quad (9)$$

where  $W_H$  are trainable parameters.

**Action-Aware (AA) Module** The architecture of AA module is similar to OA module, except the visual feature  $m_{t,k}$  is replaced with the orientation feature  $n_{t,k}$  when computing the confidence of a navigable viewpoint to be the next action:

$$\delta_{t,j} = \text{softmax}_j(u_j^\top W_A h_t) \quad (10)$$

$$\tilde{u}_t^a = \sum_j \delta_{t,j} u_j \quad (11)$$

$$\tilde{h}_t^a = \tanh(W_K[\tilde{u}_t^a; h_t]) \quad (12)$$

$$G^{AA}(a_{t,k}) = n_{t,k}^\top W_F \tilde{h}_t^a, \quad (13)$$

where  $\delta_{t,j}$  is the action attention on  $j$ -th word,  $\tilde{h}_t^a$  is action-aware hidden state, and  $W_A, W_K, W_F$  are trainable parameters. Both  $\delta_{t,j}$  and  $\tilde{h}_t^a$  have taken the navigation progress into consideration.

**Adaptive Combination Module** Figure 3(b) shows the pipeline and configuration of the proposed adaptive combination module. The final probability  $P(a_{t,k})$  of navigable view  $k$  is a softmax of weighted sum of the object-aware confidence  $G^{OA}(a_{t,k})$  and action-aware probability  $G^{AA}(a_{t,k})$ :

$$P(a_{t,k}) = \text{softmax}([G^{OA}(a_{t,k}); G^{AA}(a_{t,k})]w_t), \quad (14)$$

where  $w_t$  is a predicted weight vector.  $w_t$  should vary as the navigation goes because the importance of an object description and an action specification may change at different processing point of the instruction. Thus, to adaptively combine  $G^{OA}(a_{t,k})$  and  $G^{AA}(a_{t,k})$  in terms of the processing state, we utilize a trainable layer to predict weights  $w_t = W_T \tilde{u}_t$ , where  $W_T$  is trainable parameter,  $\tilde{u}_t$  is the vision-aware attentive instruction feature.

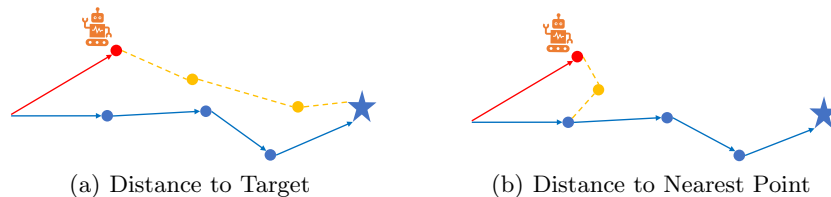
### 3.4 Training Loss

Following the training process of our baseline [19], the model is trained using both imitation learning (IL) and reinforcement learning (RL) with original training data and augmented data. In addition to the losses in the baseline, we introduce a Nearest Point Loss (NPL) to encourage the agent to stay on ground-truth paths. NPL is based on the distance to the nearest viewpoint on the ground-truth path (see Figure 4(b))

$$\mathcal{L}^{NP} = \sum_t \log(p_t(a_t)) * D_t^{NP}, \quad (15)$$

where  $D_t^{NP} = \min_{P_i \in Q} d(P_t, P_i)$ ,  $d(P_t, P_i)$  is the shortest trajectory distance between the current viewpoint  $P_t$  and each viewpoint  $P_i$  on the ground truth path,  $Q$  is the set of viewpoints on the ground-truth path. If an agent stays on the ground-truth path,  $D_t^{NP}$  would be zero; otherwise, the farther the larger.





**Fig. 4.** Illustration of distance to target (a) and distance to nearest point (b). The trajectory of the agent is in red and the ground-truth path is in blue. The star indicates the target viewpoint. The brown dot line indicates the shortest trajectory length.

## 4 Experiments

### 4.1 Dataset and Implementations

**Dataset** Experiments are conducted on the widely used indoor R2R [2] and the R4R [13] VLN datasets. R2R dataset consists of train/seen validation/unseen validation/test splits, involving 7,189 paths with 21,567 human annotated instructions with an average length of 29 words. In particular, these four splits contains 14,025/1,020/4,173/2,349 instructions, respectively. R4R aims to evaluate whether an agent is able to follow instructions by concatenating several trajectories and thus instructions of R2R (the shortest path is therefore no longer the ground-truth path) into long ones. Note that the R4R dataset only contains train/seen validation/unseen validation splits.

**Implementation Details** Following the common practice, we use the pre-computed ResNet [6] feature to represent perceived images. We adopt the same training strategy as our baseline [19]: First, we train the agent using real data (i.e., the training split); Then, we train the agent using augmented data generated by the speaker and environment dropout techniques. We implement the proposed method with PyTorch, and all the experiments are carried out on NVIDIA GeForce RTX 2080 Ti.

### 4.2 Evaluation Metrics

To evaluate the proposed method on R2R, we adopt four metrics from [1,2]: Success Rate (SR), Trajectory Length (TL), Oracle Success Rate (OSR), and Success rate weighted by Path Length (SPL). Among the four metrics, SPL is the main metric because it emphasizes the trade-off between SR and TL, which is also a recommended primary measure of navigation performance by the official VLN Challenge [1]. To measure the performance on R4R, we additionally adopt the Coverage weighted by Length Score (CLS) metric as recommended by the builder of this dataset [13] and the Success weighted by normalized Dynamic Time Warping (SDTW) metric as recommended in [12]. These two metrics focus on navigation fidelity using the ground-truth path as reference.

Model			Val Seen						Val Unseen					
Baseline	OAAM	NPL	SR↑	OSR↑	SDTW↑	SPL↑	TL↓	CLS↑	SR↑	OSR↑	SDTW↑	SPL↑	TL↓	CLS↑
✓			0.63	0.70	0.53	0.60	10.17	0.70	0.50	0.57	0.37	0.47	9.27	0.60
✓	✓		0.65	0.72	0.53	0.62	10.21	0.71	0.51	0.59	0.37	0.47	10.05	0.60
✓		✓	0.68	0.74	0.57	0.66	10.06	0.73	0.48	0.54	0.35	0.45	9.04	0.60
✓	✓	✓	0.65	0.73	0.53	0.62	10.20	0.73	0.54	0.61	0.39	0.50	9.95	0.61

**Table 1.** Ablation study of the proposed method on the R2R dataset (SPL is the main metric). The proposed object-and-action aware model (OAAM) and nearest point loss (NPL) mainly improve performance on the Unseen and Seen scenarios, respectively. Their combination leads to a better performance in unseen environments.

### 4.3 Ablation Study

We conduct the ablation study to find out the effectiveness of the proposed object-and-action aware model (OAAM) and the nearest point loss (NPL). The results are presented in Table 1. The results show that: (I) Both OAAM and NPL improve the performance on the Val Seen split with about 2% and 4% increase in SPL, respectively. (II) The phenomenon on Val Unseen is a bit complicated. When OAAM or NPL works alone with the baseline, OAAM brings about 1% improvement in SR and NPL harms the SR about 2%. When OAAM and NPL work together, the performance is significantly improved about 4% in SR and 3% in SPL. This can be attributed to that NPL is able to help OAAM to find viewpoints on the ground-truth path and therefore improves the SR as well as shortens the trajectory length compared to the case that they work separately. Overall, both OAAM and NPL facilitate our method to achieve better performance, especially in the unseen scenario.

We further study the importance of the object-aware (OA) module and the action-aware (AA) module within the trained OAAM. As these two sub-modules contribute to the final navigation decision via a weighted sum (14), we test the performance of each module by setting its weight to 1 and the other weight to 0. The results are presented in Table 2. There is about 20% performance drop in SR when OA or AA works alone. This indicates that both modules contribute to the final performance and the adaptive combination module plays a crucial role (also see visualized samples in Figure 5 and 6). Furthermore, the AA module performs consistently better than the object-aware module on ValSeen, ValUnSeen, and Test splits, in terms of SPL, SR, or OSR. For example, when AA module is removed, the SR is about 20% lower than that when OA module is removed on the Unseen split. This indicates the AA module is more generalizable than the OA, which is roughly consistent with the claim that visual feature may hurt models in unseen environments [8]. Fortunately, when OA and AA are adaptively combined, a much better performance is achieved (10% ~ 28% improvement in SPL on the Val Unseen split). Additionally, we calculate the mean of  $w_t$  of all steps on ValSeen and ValUnseen. Results show that OA and AA are assigned 0.18 and 0.82 on average, respectively. This also reflects AA contributes more to the final results. Lastly, we make a try to conduct quantitative evaluation of these two modules based on NLTK tags, although it is not perfect.

Model	Val Seen							Val Unseen						
	SR↑	OSR↑	SDTW↑	SPL↑	TL↓	CLS↑		SR↑	OSR↑	SDTW↑	SPL↑	TL↓	CLS↑	
Full Model	0.65	0.73	0.53	0.62	10.20	0.73		0.54	0.61	0.39	0.50	9.95	0.61	
w/o AA Module	0.42	0.52	0.31	0.39	9.26	0.59		0.26	0.34	0.15	0.22	8.86	0.46	
w/o OA Module	0.45	0.53	0.32	0.42	10.63	0.56		0.44	0.52	0.30	0.40	10.46	0.54	

**Table 2.** Comparison of object-aware module (OA) and action-aware module (AA) on the R2R dataset. ‘w/o’ denotes to remove the module from the full model. The AA module contributes more than the OA module.

An attention is considered success if the top3 words attended by OA contain nouns (verbs for AA). OA and AA achieve accuracy of about 80% and 75% on average, respectively. This at some extent indicates improvement room for these modules.

#### 4.4 Comparison to State-of-The-Art Navigators

In this section, we compare our model against six state-of-the-art navigating methods, including FAST [14], RCM [20], EnvDrop [19], Speaker-Follower (SF) [4], RegretAgent (RA) [18], and Self-Monitor (SM) [17]. Results on the R2R and R4R datasets are presented in Table 3 and 4, respectively.

As shown in Table 3, our model achieves the best performance on both the Val UnSeen and Test splits in terms of the main evaluation metric, SPL, which is up to 50%. As analysed in the ablation study, the performance of our model is the result of both the OAAM and the NPL loss. We also observe that on the Val Seen split, our model ranks third falling behind FAST and RA. This indicates that our model may not fit the training data as well as FAST and RA, and there are more information can be learned from the training data for our model. We leave this for the future exploration.

Table 4 shows the result on the R4R dataset, where CLS is the recommended metric [13]. On one side, the proposed method performs consistently better than our baseline, namely Envdrop, with up to 6% improvement in CLS on the unseen split. On the other side, the proposed method achieves the best performance with a 0.40 CLS score in unseen environments, which is 3% higher than the second best and thus set the new SoTA. Similar to the phenomenon on the R2R dataset, we also observe that the proposed model ranks top three in the seen environments, which indicates more information could be learned.

#### 4.5 Qualitative Results

In this section, we visualize several navigation process at each step in Figure 5 and Figure 6, including key information, such as attention distribution and navigation decision.

For the trajectories in Figure 5, most of the attention predicted by the OA and AA modules are able to correctly reflect the object or action that should be considered to make correct decisions. At some midway steps, such as step 1 and

Model	Val Seen				Val Unseen				Test			
	SR↑	OSR↑	SPL↑	TL↓	SR↑	OSR↑	SPL↑	TL↓	SR↑	OSR↑	SPL↑	TL↓
SF [4]	0.67	0.74	0.61	11.73	0.35	0.45	0.28	14.56	0.35	0.44	0.28	14.82
SM [17]	0.70	0.80	0.61	12.90	0.43	0.54	0.29	17.09	0.48	0.59	0.35	18.04
RA [18]	0.69	0.77	0.63	-	0.50	0.59	0.41	-	0.48	0.56	0.40	13.69
RCM [20]	0.67	0.75	-	10.65	0.43	0.50	-	11.46	0.43	0.50	0.38	11.97
FAST [14]	0.70	0.80	0.64	12.34	0.56	0.67	0.44	20.45	0.54	0.64	0.41	22.08
EnvDrop [19]	0.63	0.70	0.60	10.17	0.50	0.57	0.47	9.27	0.50	0.57	0.47	9.70
<b>Ours</b>	0.65	0.73	0.62	10.20	0.54	0.61	0.50	9.95	0.53	0.61	0.50	10.40

**Table 3.** Comparison against several state-of-the-art VLN methods on the R2R dataset. SPL is the main metric. The best three results are highlighted in red, green, and blue fonts, respectively.

Model	Val Seen					Val Unseen				
	SR↑	SDTW↑	SPL↑	TL↓	CLS↑	SR↑	SDTW↑	SPL↑	TL↓	CLS↑
SF [4]	0.52	-	0.37	15.40	0.46	0.24	-	0.12	19.90	0.30
RCM goal-oriented [13]	0.56	-	0.32	24.50	0.40	0.29	-	0.10	32.50	0.20
RCM fidelity-oriented [13]	0.53	-	0.31	18.80	0.55	0.26	-	0.08	28.50	0.35
PTA high-level [15]	0.58	0.41	0.39	16.50	0.60	0.24	0.10	0.10	17.70	0.37
EnvDrop [19]	0.52	0.27	0.41	19.85	0.53	0.29	0.09	0.18	26.97	0.34
<b>Ours</b>	0.56	0.32	0.49	11.75	0.54	0.31	0.11	0.23	13.80	0.40

**Table 4.** Results on the R4R dataset. CLS is the main metric. The best three results are highlighted in red, green, and blue fonts, respectively.

5 on the left panel of Figure 5, only one module gives the correct prediction but the final prediction is correct after combination. We also show examples that some midway decisions are incorrect but finally get to the goal room in Figure 6, such as step 3 in the left panel.

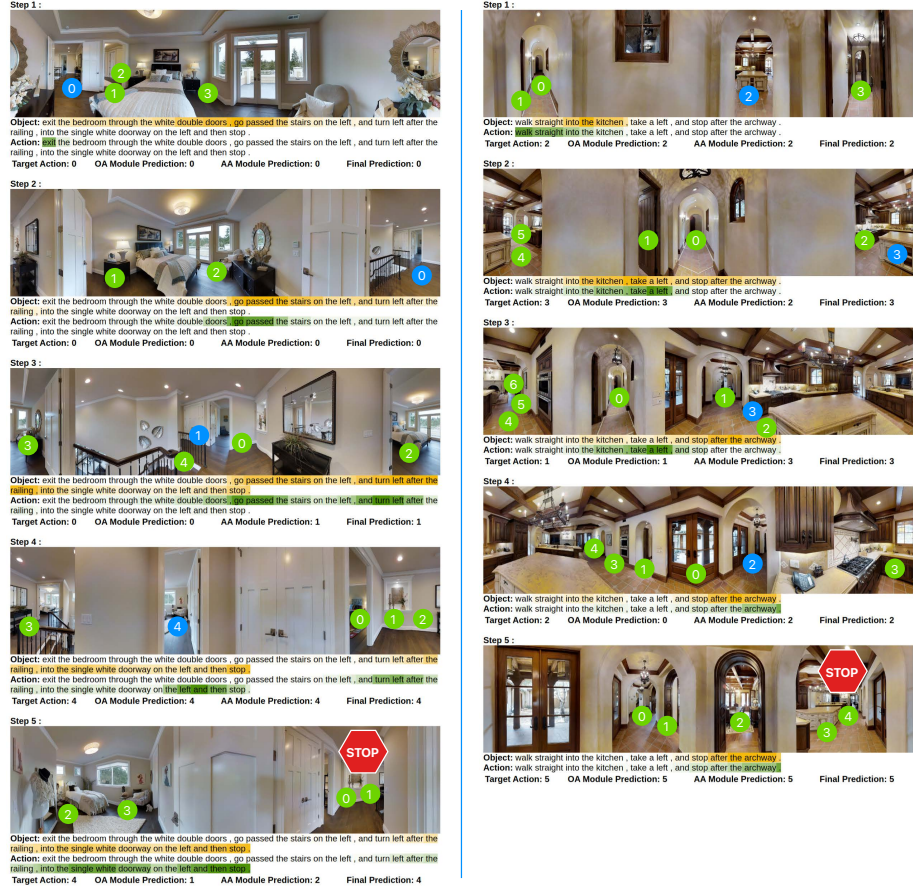
## 5 Conclusion

In this paper, we propose an object-and-action aware model for robust visual-and-language navigation. Object and action information in instructions are separately highlighted, and are then matched to visual perceptions and orientation embeddings, respectively. To encourage the robot agent to stay on the path, we additionally propose a path loss based on the distance to nearest ground-truth viewpoint. Extensive experimental results demonstrate the superiority of our model compared against several state-of-the-art navigating methods, especially on the unseen test scenarios.



**Fig. 5.** Navigation samples of the proposed method in two unseen environments. The dark or light color on the instruction denotes the relative attention value (the darker the larger). The numbered circles denote navigable viewpoints, and the blue ones denote the final decision.





**Fig. 6.** Two more navigation samples of the proposed method in unseen environments. On the left panel, at step 3, our model predicts an incorrect action but gets to the goal room in end. It is worth noting that at step 5, both OA and AA modules give wrong prediction, but after combination our model predicts the correct action. On the right panel, the final decision at each step is correct, but we could observe that the OA or AA module not always gives correct prediction, such as at step 2 and step 3. These indicate that all the OA and AA modules as well as the adaptive combination module are crucial to the final success.

## References

1. Anderson, P., Chang, A.X., Chaplot, D.S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., Zamir, A.R.: On evaluation of embodied navigation agents. CoRR **abs/1807.06757** (2018) [9](#)
2. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I.D., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: CVPR. pp. 3674–3683 (2018) [2](#), [4](#), [9](#)
3. Andreas, J., Rohrbach, M., Darrell, T., Klein, D.: Learning to compose neural networks for question answering. In: NAACL. pp. 1545–1554 (2016) [4](#)
4. Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L., Berg-Kirkpatrick, T., Saenko, K., Klein, D., Darrell, T.: Speaker-follower models for vision-and-language navigation. In: NeurIPS. pp. 3318–3329 (2018) [2](#), [4](#), [6](#), [11](#), [12](#)
5. Fu, T., Wang, X., Peterson, M., Grafton, S., Eckstein, M.P., Wang, W.Y.: Counterfactual vision-and-language navigation via adversarial path sampling. CoRR **abs/1911.07308** (2019) [2](#)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) [9](#)
7. Hu, R., Andreas, J., Rohrbach, M., Darrell, T., Saenko, K.: Learning to reason: End-to-end module networks for visual question answering. In: ICCV. pp. 804–813 (2017) [4](#)
8. Hu, R., Fried, D., Rohrbach, A., Klein, D., Darrell, T., Saenko, K.: Are you looking? grounding to multiple modalities in vision-and-language navigation. In: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28– August 2, 2019, Volume 1: Long Papers. pp. 6551–6557 (2019) [10](#)
9. Hu, R., Rohrbach, M., Andreas, J., Darrell, T., Saenko, K.: Modeling relationships in referential expressions with compositional modular networks. In: CVPR. pp. 4418–4427 (2017) [4](#)
10. Hu, R., Rohrbach, M., Andreas, J., Darrell, T., Saenko, K.: Modeling relationships in referential expressions with compositional modular networks. In: CVPR. pp. 4418–4427 (2017) [7](#)
11. Huang, H., Jain, V., Mehta, H., Ku, A., Magalhães, G., Baldridge, J., Ie, E.: Transferable representation learning in vision-and-language navigation. In: ICCV. pp. 7403–7412 (2019) [2](#)
12. Ilharco, G., Jain, V., Ku, A., Ie, E., Baldridge, J.: General evaluation for instruction conditioned navigation using dynamic time warping. In: Visually Grounded Interaction and Language (ViGIL), NeurIPS 2019 Workshop, Vancouver, Canada, December 13, 2019 (2019) [9](#)
13. Jain, V., Magalhães, G., Ku, A., Vaswani, A., Ie, E., Baldridge, J.: Stay on the path: Instruction fidelity in vision-and-language navigation. In: ACL. pp. 1862–1872 (2019) [3](#), [9](#), [11](#), [12](#)
14. Ke, L., Li, X., Bisk, Y., Holtzman, A., Gan, Z., Liu, J., Gao, J., Choi, Y., Srinivasa, S.S.: Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In: CVPR. pp. 6741–6749 (2019) [2](#), [3](#), [4](#), [11](#), [12](#)
15. Landi, F., Baraldi, L., Cornia, M., Corsini, M., Cucchiara, R.: Perceive, transform, and act: Multi-modal attention networks for vision-and-language navigation. CoRR **abs/1911.12377** (2019) [12](#)

16. Li, X., Li, C., Xia, Q., Bisk, Y., Çelikyilmaz, A., Gao, J., Smith, N.A., Choi, Y.: Robust navigation with language pretraining and stochastic sampling. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) EMNLP. pp. 1494–1499 (2019) [2](#)
17. Ma, C.Y., Lu, J., Wu, Z., AlRegib, G., Kira, Z., Socher, R., Xiong, C.: Self-monitoring navigation agent via auxiliary progress estimation. In: ICLR (2019) [2](#), [3](#), [4](#), [11](#), [12](#)
18. Ma, C., Wu, Z., AlRegib, G., Xiong, C., Kira, Z.: The regretful agent: Heuristic-aided navigation through progress estimation. In: CVPR. pp. 6732–6740 (2019) [2](#), [3](#), [4](#), [11](#), [12](#)
19. Tan, H., Yu, L., Bansal, M.: Learning to navigate unseen environments: Back translation with environmental dropout. In: NAACL. pp. 2610–2621 (2019) [2](#), [3](#), [4](#), [5](#), [6](#), [8](#), [9](#), [11](#), [12](#)
20. Wang, X., Huang, Q., Çelikyilmaz, A., Gao, J., Shen, D., Wang, Y., Wang, W.Y., Zhang, L.: Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In: CVPR. pp. 6629–6638 (2019) [11](#), [12](#)
21. Wang, X., Jain, V., Ie, E., Wang, W.Y., Kozareva, Z., Ravi, S.: Environment-agnostic multitask learning for natural language grounded navigation. CoRR **abs/2003.00443** (2020) [2](#)
22. Wang, X., Xiong, W., Wang, H., Wang, W.Y.: Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV. pp. 38–55 (2018) [2](#)
23. Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., Berg, T.L.: Mattnet: Modular attention network for referring expression comprehension. In: CVPR. pp. 1307–1315 (2018) [4](#), [7](#)