2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Guided by the Way: The Role of On-the-route Objects and Scene Text in Enhancing Outdoor Navigation

Yanjun Sun[1,2], Yue Qiu[2], Yoshimitsu Aoki[1], Hirokatsu Kataoka[2]
[1]National Institute of Advanced Industrial Science and Technology (AIST), [2]Keio University

***Abstract*— In outdoor environments, Vision-and-Language Navigation (VLN) requires an agent to rely on multi-modal cues from real-world urban environments and natural language instructions. While existing outdoor VLN models predict actions using a combination of panorama and instruction features, this approach ignores objects in the environment and learns data bias to fail navigation. According to our preliminary findings, most instances of navigation failure in previous models were due to turning or stopping at the wrong place. In contrast, humans intuitively frequently use identifiable objects or store names as reference landmarks, ensuring accurate turns and stops, especially in unfamiliar places. To address this insight gap, we propose an Object-Attention VLN (OAVLN) model that helps the agent focus on relevant objects during training and understand the environment better. Our model outperforms previous methods in all evaluation metrics under both seen and unseen scenarios on two existing benchmark datasets, Touchdown and map2seq.**

I. INTRODUCTION

Enabling a robot to navigate real-world environments using natural language instructions is a longstanding goal in AI research. Various approaches have been proposed to achieve this goal in the vision-and-language navigation (VLN) field [1]–[4]. The VLN task requires the agent to understand the instructions, ground it in the observable environment using visual perception, reason about its position to objects and how these relations change as it moves through the environment, and then perform corrective actions to reach the destination.

Recent studies on outdoor VLN models [4]–[8] have utilized simple encoder-decoder models that concatenate instruction features and panoramic features and use a decoder to predict actions. However, these models struggle with semantic learning and produce agents that misunderstand the environment as they ignore objects and tend to learn biases in the data. Nevertheless, by visualizing the paths generated by existing methods, we observed that the current models pay insignificant attention to the object tokens. In numerous cases, the agent ignores the landmarks specified by the instructions and erroneously makes a turn or stops at the wrong location, leading to failed navigation.

Meanwhile, DiagnoseVLN [9] has shown through several masked experiments that, in the case of outdoor VLN, the agent prefers to use directional information and ignores objects from the instructions. However, this is not intuitive to humans. When navigating in an unfamiliar environment, humans prefer to use buildings and other objects and scene text seen along the way as reference landmarks [10]. For example, as shown in Fig. 1, according to the instructions,

Fig. 1. Objects are important clues in outdoor VLN. Our Object-Attention VLN model is designed to navigate using this information. At viewpoint (b), our agent seeks the 'black iron fence' and turns right. Subsequently, it stops at the viewpoint (c) because it has observed the 'blue bikes.'

people should turn right at the 'fence' and stop at 'the last blue bike.' These objects provide crucial clues in outdoor VLN.

Inspired by the significant results of object-aware indoor VLN models [11]–[17] have demonstrated the utility of using object features. We believe better use of landmarks in outdoor VLN scenarios is more rational for navigation and may improve navigation performance. However, it should be noted that the scenarios that indoor models can accommodate are often narrow. Indoor scenario typically takes place in a stable, structured environment. In contrast, outdoor environments are much more complex and diverse and often lack well-defined structures or clear boundaries. Consequently, navigating outdoors requires recognizing a wide range of visual cues, including natural and man-made features.

To address the abovementioned limitations, we propose a simple yet effective Object-Attention VLN (OAVLN) model that allows the agent to focus more on objects and scene texts to understand the environment better. To demonstrate the effectiveness of our proposed method, we have extensively experimented on the Touchdown [4] and map2seq [18] dataset with four baselines outdoor VLN models [4], [5], [7], [8]. The experimental results show that our model outperformed existing methods, even in unseen scenarios. Our qualitative results further verify that the improvement comes from agents' improved capability to utilize objects more effectively and to turn or stop at suitable locations.

979-8-3503-8457-4/24/$31.00 ©2024 IEEE

II. Related Works

A. Vision-and-language Navigation.

VLN unites two scenarios: first, of indoor scenario, the first VLN benchmark R2R [1] was proposed for the indoor scenario based on Matterport3D [19], along with a multi-modal Seq2Seq baseline model. Several indoor VLN benchmarks were created by extending R2R with other languages, such as XL-R2R [20] and RxR [2]. Touchdown [4] dataset was proposed as the first outdoor VLN benchmark and based on Google Street View[1], which provided a more unstable environment. Subsequently, the datasets StreetLearn [21], Retouchdown [22], StreetNav [23], map2seq [18], and Talk2Nav [24] for outdoor VLN were proposed.

Several VLN task methods have been recently proposed for these benchmarks. RCONCAT [4] is the baseline model presented in the original Touchdown study, and it encodes the trajectory and the instructions in an LSTM-based model. ARC [6] proposed ARC+l2s, which cascades the action prediction into a binary stopping decision and subsequent direction classification. VLN-Transformer [7] uses a pre-trained BERT [25] applied to an external multimodal dataset to enrich the navigation data. GA [5] uses gated attention to compute a fused representation of instructions and images to predict actions. ORAR [8] adds junction-type embedding and a heading delta to train a general model, reducing the performance gap between seen and unseen environments. However, these outdoor VLN models exclusively utilize LSTM as their encoder-decoder architecture to encode the instructions and predict the actions. This architecture constrains the model's ability to focus on particular semantics, such as objects, which are crucial clues in VLN.

B. Object-aware VLN.

To handle fine-grained information, such as objects, and to facilitate simultaneous understanding of instructions and visuals, Vision-and-language pre-trained models such as ViLBERT [26] have been widely adopted in VLN because they can provide good joint features for better understanding of instructions and environmental features. ORIST [27] was trained with specific features such as objects and rooms. OAAM [11] extracted tokens from instructions and encoded object tokens to predict actions. SOAT [13] uses a scene classification network and an object detector to produce features that match these two distinct visual cues. However, these VLN models focused on indoor scenarios, which are the most static environments. The vast urban areas being navigated lead to a much larger space for the agent to explore and contain a broader range of encountered objects in the visual environment. Therefore, outdoor VLN requires special treatment in recognizing these diverse objects and using them to understand the environment better. We consider the instability of the outdoor environment, such as the weather, and use scene text recognition to recognize the names of stores and objects in the outdoor environment while navigating.

[1]https://developers.google.com/maps/documentation/streetview/intro

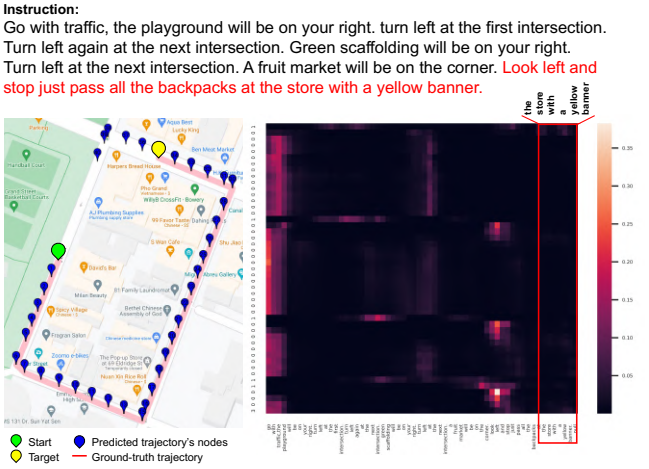


Fig. 2. Example of visualization of the ORAR model on the Touchdown dataset. The top of this figure is the instruction, and the red text is the distribution of stop location, which ORAR disregarded. Left: trajectory generated by ORAR vs. ground truth. Right: Attention to each token from the instructions during predicting actions.

III. Preliminary

A. VLN Problem Definition

The VLN task requires an agent in an environment to follow natural language instruction $X = \{x_1, x_2, \ldots, x_l\}$, within an environment represented as an undirected graph with nodes $v \in \mathbb{V}$ and labeled edges $(v,u) \in \mathbb{E}$. Each node is associated with a panoramic RGB image, and each edge connects nodes to neighboring panoramas at the heading angle $\alpha_{(v,u)}$. The state of the agent at time t is defined by $s_t = (v_t, \alpha_{(v_{t-1},v_t)})$. Following the instructions X, the agent executes an action $a_t \in \{\text{FORWARD, LEFT, RIGHT, STOP}\}$ and is updated to the next state s_{t+1}. Ultimately, the agent produces a sequence of state-action pairs, culminating in the action $a_n = \text{STOP}$ to reach its destination.

B. What do agents focus on when navigating?

DiagnoseVLN [9] reports that task completion nearly drops to zero when the masking direction word tokens are during testing only and that masking out the object tokens has a weaker impact on task completion rate than the masking direction tokens. The authors concluded that direction tokens were more important than object tokens for VLN tasks and suggested that future work explore the use of direction tokens in greater depth. While it may seem counterintuitive, the importance of different types of tokens may vary depending on the specific task and environment being navigated. Object tokens are sometimes more efficient in pinpointing landmarks and deciding turns.

Therefore, to determine which tokens the trained agent was paying attention to during outdoor navigation, we visualized the generated trajectory with the Google Map API[2]. We plotted a heatmap of instruction attention weight for the ORAR [8] model. Fig. 2 shows an example of the

[2]https://developers.google.com/maps/documentation

visualization. The x-axis of the heatmap represents each token of the instructions, while the y-axis represents the predicted actions of the agent at each timestep. Each grid on this heatmap indicates attention received by a token when the agent predicted the action. The more attention a token receives, the brighter color of the grid. This example shows that the agent was instructed to stop at 'the store with a yellow banner', but ignored this information and turned left at the next junction, eventually stopping at the wrong location to fail navigation. The heatmap shows that the attention weight for 'the store with a yellow banner' was almost zero during navigation. Furthermore, we analyze the attention weight of object tokens in the instructions from the test set, and the average weight of each object token is 0.128 in the instructions. According to the preliminary results reported above, existing outdoor VLN models cannot pay attention to object tokens during navigation, leading to turning or stopping at the wrong location. Additionally, even some non-content words, like 'the', have more attention than object tokens, indicating that the existing model has been learning data biases by ignoring objects.

IV. Object Attention VLN

In this section, we introduce the proposed model for outdoor VLN. As illustrated in Fig. 3, the proposed model architecture comprises a two-layer decoder to generate a sequence of agent actions based on four input sequences: navigation instructions text, panorama features, object features, and scene text. The model computes a visual representation of the current agent state within the environment at each decoding timestep, considering the previously predicted actions. Specifically, the first decoder layer encodes both metadata and visual representations, including panorama and object features, whereas the second layer encodes contextualized text, such as attention-based panorama and object features, attention-based scene text, navigation instruction text, and current timestep, to predict the next action. The model follows a sequence-to-sequence architecture, with the input sequences processed sequentially to generate the output sequence of agent actions.

Instruction Encoder. The instruction encoder embeds and encodes the tokens in the navigation instructions sequence $\mathbf{x} = x_1, ..., x_L$ using a bidirectional LSTM [28]:

$$\hat{x}_i = \text{embedding}(x_i) \tag{1}$$

$$((w_1, ..., w_L), z_L^w)) = \text{Bi-LSTM}(\hat{x}_1, ..., \hat{x}_L) \tag{2}$$

where $w_1, ..., w_L$ are the hidden representations for each token and z_L^w is the last LSTM cell state.

Panorama Encoder & Object Encoder At each timestep t, the panorama at the current agent position is represented by extracted visual features. We sliced the panorama into eight projected rectangles with 60° fields of view, such that one of the slices aligned with the agent's heading. There are five slices of a panorama: the center slice and the two left and right. We then fed the five slices into pre-trained ResNet-50 [29] on ImageNet [30] to extract high-level features. Each slice feature vector $\bar{p}_t^s$ was of size 2,048. Then, we extract

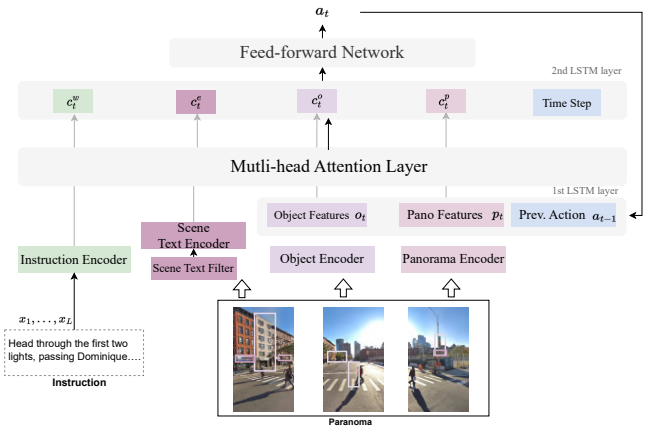


Fig. 3. Overview of the proposed model.

20 objects from each slice. Each object is represented by features extracted using a pre-trained ResNet-101 [29] on Visual Genome[3], and the feature vector for each slice vector of objects $\bar{o}_t^s$ also has a size of 2,048.

Scene Text Filter & Scene Text Encoder. For improved scene text from low-quality panorama images, we developed a Scene Text Filter. We use the Object Encoder to detect the entire panorama to identify the 'sign' regions. Then, we applied scene text recognition only to these 'sign' regions using the MMOCR [31] model on SAR [32] model for text recognition. Finally, we corrected the recognition results using the closest scene text mentioned in the instructions.

These scene text $\mathbf{e} = e_1, ..., e_M$ were embedded and encoded by a bidirectional LSTM:

$$\hat{e}_i = \text{embedding}(e_i) \tag{3}$$

$$((w_1, ..., w_M), z_M^w)) = \text{Bi-LSTM}(\hat{e}_1, ..., \hat{e}_L) \tag{4}$$

where $w_1, ..., w_M$ are the hidden representations for each token and z_M^w is the last LSTM cell state.

Decoder The panorama encoder, as described in detail above generates a fixed size representation $\bar{p}_t$ of the sequence of sliced visual representations of the current panorama view, denoted as $\bar{p}_t^1, ..., \bar{p}_t^S$. Similarly, the object encoder emits a fixed size representation $\bar{o}_t$ of the objects in the current panorama and a sequence of sliced view representations $\bar{o}_t^1, ..., \bar{o}_t^S$. The state z_0^{first} of the cell in the first decoder LSTM layer was initialized using z_L^w. The input to the first decoder layer was the concatenation ($\oplus$) of previous action embedding $\bar{a}_{t-1}$, visual representation $\bar{p}_t$ and object features $\bar{o}_t$. The output of the first decoder layer,

$$h_t^{first} = \text{LSTM}^{first}([\bar{a}_{t-1} \oplus \bar{p}_t \oplus \bar{o}_t]), \tag{5}$$

was then used as the query of multi-head attention [33] over the text encoder. The resulting contextualized text representation c_t^w was then used to attend over the sliced visual representations c_t^p, object representations c_t^o, and scene

[3]http://visualgenome.org/

text encode c_t^e. The input and output of the second decoder layer were

$$h_t^{second} = \text{LSTM}^{second}([\bar{t} \oplus h_t^{first} \oplus c_t^p \oplus c_t^o \oplus c_t^e]), \quad (6)$$

where $\bar{t}$ represents embedded timestep t. The hidden representation h_t^{second} from the second decoder layer goes through a feed-forward network to predict action a_t.

V. Experiments

A. Experimental Setup

Implementation Details. Our framework and baselines were developed in PyTorch [34]. ResNet50 [29] was used for panorama features, while Faster R-CNN [35], pretrained on Visual Genome [36] with ResNet101 [29], was used for object features with an IoU score of 0.6. Scene text was recognized using MMOCR [31]. The object tokens in instructions were summarized with stanza [37], which also optimized scene text recognition by a sequence matching algorithm [38] with 0.8 similarity score. The models were trained using Adam [39] under teacher-forcing, with parameters such as a learning rate of 5e-4, weight decay of 1e-3, batch size of 64, and dropout rates of 0.3. After 150 epochs, the top model was selected from the development set. Instructions and scene texts were converted to byte pair encodings [40] with a 2,000 token vocabulary and embedded at 256. Other embeddings were 256 and 16 in size.

Datasets: The Touchdown [4] and map2seq [18] dataset use urban scenarios to create a large navigation environment based on Google Street View[4]. The environment simulates NYC, comprising 29,641 nodes and 61,319 undirected edges. The touchdown dataset includes 9,326 navigation trajectories, each paired with human-written instructions based on the corresponding panoramas, within 6,525 training, 1,391 development, and 1,409 test samples. The instructions in map2seq instead focused on visual landmarks from OpenStreetMap. map2seq comprises 7,672 navigation instructions, segmented into 6,072 training, 800 development, and 800 test samples. Furthermore, we followed ORAR [8] to split the datasets based on the geographic separation of the training and testing areas for the unseen scenario.

Baselines. We compared our model to previous studies on outdoor VLN, including RCONCAT [4], GA [5], VLN-Transformer [7], and ORAR [8]. These models use an LSTM to encode the instruction text and a single-layer decoder LSTM to predict the next action. We selected these models because they do not specifically handle on-the-route object features in detail. By comparing our results with these baseline models, we demonstrated that incorporating on-the-route object features benefits outdoor VLN.

Metrics. The following metrics were used to evaluate the VLN performance: (1) Task Completion (TC): This metric measures the navigation accuracy of the agent to the correct location, which can be either the exact goal panorama or one of its neighboring panoramas. (2) Shortest-Path Distance (SPD) [4]: This metric calculates the average distance between the final position of the agent and the goal position in the environment graph. (3) Success weighted by Edit Distance (SED): This metric calculates the normalized Levenshtein edit distance [41] between the predicted and ground-truth paths, only awarding points for successful paths. (4) Coverage weighted by Length Score (CLS) [42]: This metric measures the similarity between the path of the agent and the ground-truth path. (5) Normalized Dynamic Time Warping (nDTW) [43]: This metric measures the cumulative distance between the predicted and ground-truth paths. (6) Success-weighted Dynamic Time Warping (SDTW): This metric is the nDTW value calculated only for successful navigations.

[4]https://developers.google.com/maps/documentation/streetview/overview

B. Quantitative Results

In this section, we report the outdoor VLN performance and the quality of object features to validate the effectiveness of our proposed OAVLN model. We compared the results on seen and unseen scenarios and discussed the influence of on-the-route objects and scene text in the outdoor VLN.

Seen Scenario. Tab. I compares our model with other studies in seen scenarios. Our model outperformed the baselines in each metric, showcasing the effectiveness of the different types of datasets. Notably, our model made significant improvements in the path alignment metrics (CLS, sDTW), highlighting the power of object feature attention by instructions to increase instruction following and goal achievement. In particular, OAVLN(+scene text) caused a 6% rise in goal-oriented metrics (TC, SED) on the map2seq dataset, indicating that our model uses objects to stop better than baselines. The Touchdown boost was minor as the map2seq instructions focused more on the objects.

Unseen Scenario. Tab. II shows the results of our model with other studies on unseen scenarios. The results are presented separately for both datasets' development and test sets. While a comparison of all studies on the seen scenarios was reduced in the unseen environment, we still achieved a 3% improvement in both TC and nDTW compared to previous studies. These results indicate that OAVLN can follow instructions and complete tasks with higher accuracy and reliability in the unseen scenario. The detailed on-the-route object features can help the agent identify turn and stop locations in unseen environments.

C. Qualitative Results

Visualization of trajectories. We provided visualizations of four qualitative examples to illustrate further how our OAVLN model learns to stop better and turn at the correct location in Fig. 4. This was achieved by leveraging object features, scene text, and language features, enabling OAVLN to comprehend the environment more effectively and archive more reliable navigation decisions. As shown in Fig. 4, the baseline often overlooks landmarks signifying key turns or stops. Specifically, in 4(a) and 4(b), it misunderstands instructions and its surroundings, turning wrongly. In 4(c) and 4(d), it stops early, neglecting subsequent direction details. Contrarily, our agent turns and stops correctly.

TABLE I

NAVIGATION RESULTS ON TOUCHDOWN AND MAP2SEQ FOR THE SEEN SCENARIO.

Dataset	Touchdown						map2seq					
Model	**TC↑**	**SPD↓**	**SED↑**	**CLS↑**	**nDTW↑**	**sDTW↑**	**TC↑**	**SPD↓**	**SED↑**	**CLS↑**	**nDTW↑**	**sDTW↑**
RCONCAT [4]	8.94	22.48	8.55	43.23	18.20	7.98	14.62	20.61	14.30	54.18	27.43	13.76
GA [5]	9.87	20.34	9.42	47.77	21.51	8.92	17.88	18.25	17.55	58.56	31.46	17.08
VLN Transformer [7]	14.90	21.20	14.60	45.40	25.30	14.00	17.00	-	-	-	29.50	-
ORAR [8]	24.23	17.30	23.70	56.87	37.20	22.87	43.96	6.93	43.09	82.97	60.43	41.78
Ours (+scene text)	24.77	**15.98**	24.14	59.93	37.64	23.14	**50.00**	**6.11**	**49.04**	**84.77**	**65.39**	**47.45**
Ours (+objects)	**25.90**	16.04	**25.40**	**60.84**	**39.00**	**24.47**	49.00	6.40	48.08	84.28	63.38	46.75

TABLE II

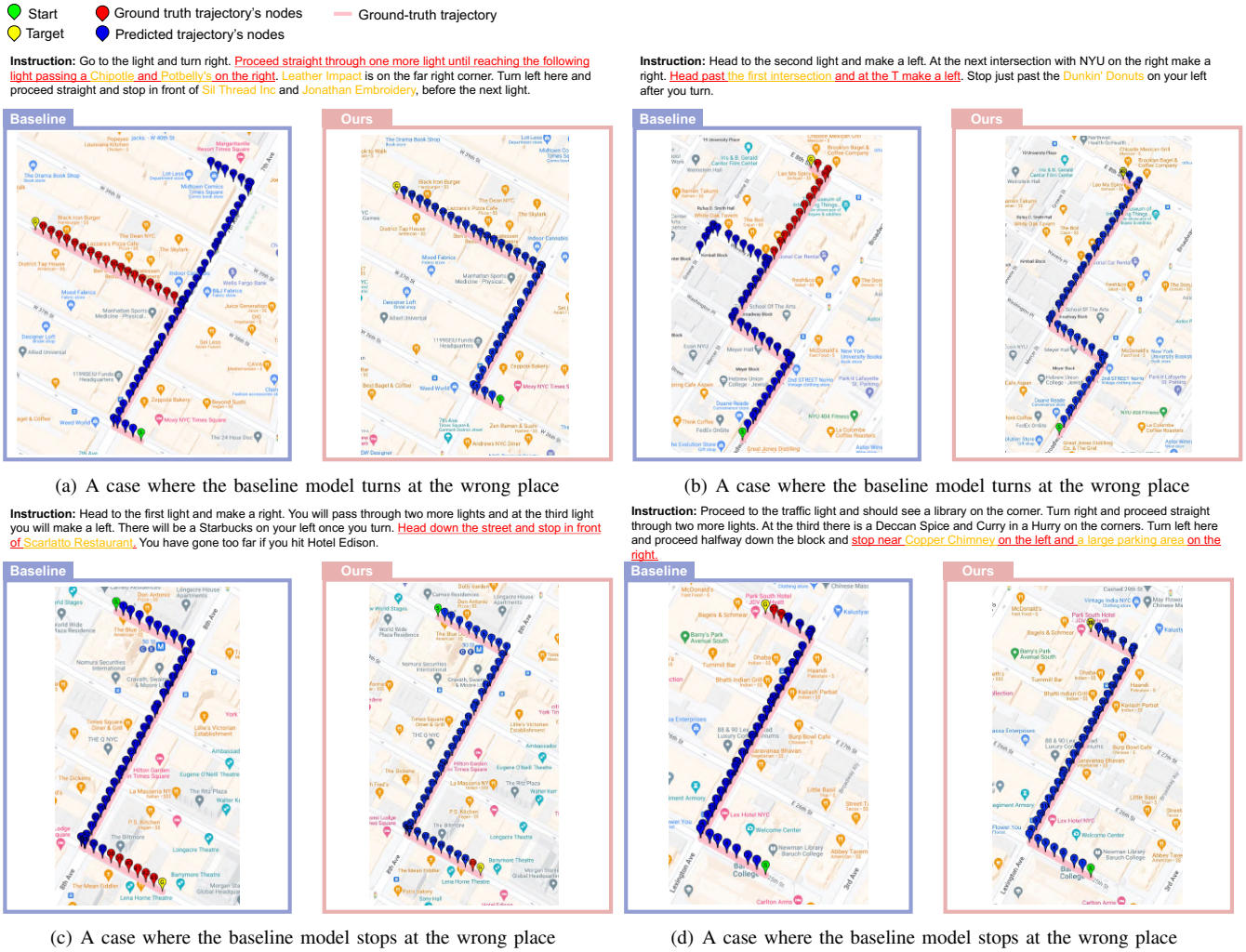
NAVIGATION RESULTS FOR THE UNSEEN SCENARIO.

Dataset	Touchdown				map2seq			
	dev		test		dev		test	
Model	**TC↑**	**nDTW↑**	**TC↑**	**nDTW↑**	**TC↑**	**nDTW↑**	**TC↑**	**nDTW↑**
RCONCAT [4]	2.3	3.9	1.9	3.5	2.0	3.7	2.1	3.8
GA [5]	1.8	3.6	2.2	4.0	1.8	3.9	1.7	4.1
VLN Transformer [7]	2.3	4.7	3.1	5.2	3.6	6.2	3.5	6.1
ORAR [8]	8.50	11.13	**8.76**	11.74	23.88	34.34	22.12	32.69
Ours (+scene text)	9.25	12.83	8.12	11.73	**26.25**	**35.63**	25.25	35.49
Ours (+object)	**10.25**	**13.86**	8.63	**12.12**	25.87	35.27	**25.37**	**36.56**

TABLE III

ACCURACY OF MODELS AT STOP AND TURN LOCATIONS.

	Touchdown		map2seq	
	Stop	Turn	Stop	Turn
	Seen			
ORAR	72.70%	60.48%	50.89%	6.10%
Ours	**73.51%**	**62.94%**	**59.90%**	**12.00%**
	Unseen			
ORAR	92.26%	26.97%	71.15%	6.34%
Ours	**93.17%**	**28.90%**	**80.08%**	**12.25%**



(a) A case where the baseline model turns at the wrong place

(b) A case where the baseline model turns at the wrong place

(c) A case where the baseline model stops at the wrong place

(d) A case where the baseline model stops at the wrong place

Fig. 4. Visualizations of baseline and our model. Four examples from the test set show the success of our approach, but the baseline model stops too early or turns at the wrong place. Left: trajectory generated by ORAR. Right: trajectory generated by OAVLN model. The red text in the instructions is where the existing method made a mistake when generating the path. The orange text represents the object tokens.

In addition, we calculated the agent's accuracy in predicting actions at locations where it needed to stop or turn, and the results are presented in Tab. III. Especially for stop accuracy, we defined wrong stops as examples where the agent was within five steps of reaching the goal but stopped at the wrong location. Tab. III indicates that our OAVLN had a lower percentage of navigation failures due to wrong turns and stops than ORAR.

Token Masking. To further prove that our proposed model is more object-focused than previous methods, we ran masking experiments similar to ORAR [8] and DiagnoseVLN [9]. We masked the object tokens during the test only. Fig. 5 shows the changes in task completion rates when masked object tokens were used. From the widening gap between our study and previous studies, our model enabled learning object tokens from the instructions and panoramas. This finding disagrees with DiagnoseVLN [9], which reports that the object tokens are not crucial for navigating.

To illustrate, in Fig.6(a), ORAR emphasizes the sentence's initial segment, while OAVLN values the time series sentence, highlighting object tokens. We examined the attention weight of object tokens in the test set. As detailed in section III-B, ORAR's average attention is 0.128, and OAVLN's is 0.374 for instruction object tokens.

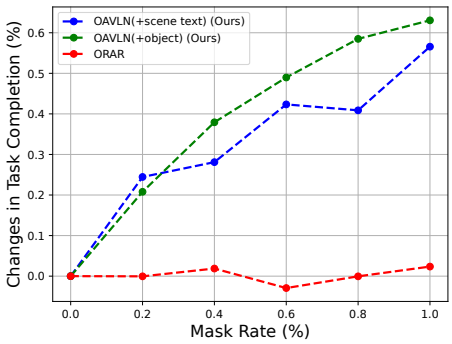
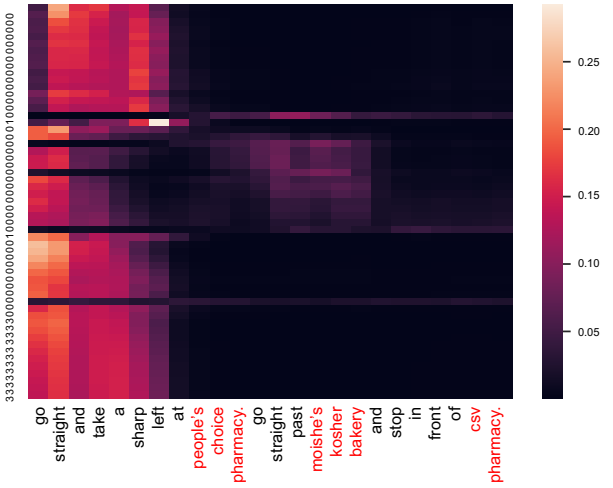


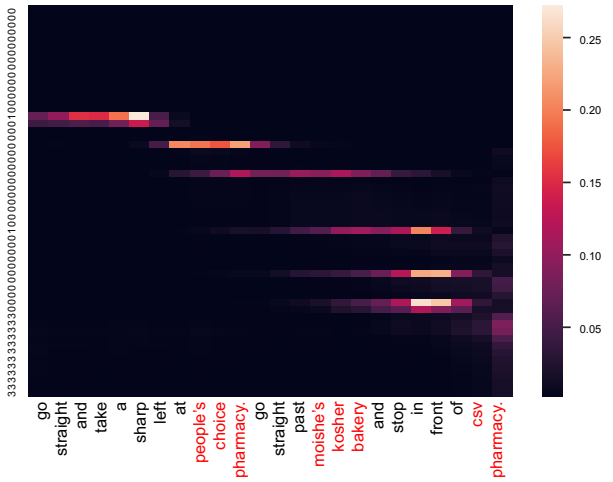
Fig. 5. Changes in the navigation performance (TC metrics) when masking object tokens in instructions on Touchdown seen scenario data.

D. Analysis

Our OAVLN works well in a seen scenario and an unseen environment, proving that the on-the-route object feature is helpful for outdoor VLN. The results shown below indicated that the 'object feature' and 'scene text' are necessary. It can help the agent focus on on-the-route objects, enabling the agent's localization. Specifically, OAVLN assists the agent in turning and stopping more accurately, which is a more intuitive approach. Therefore, even in unknown locations, OAVLN can use surrounding objects as references to reach the goal. Moreover, our work highlights the importance of leveraging contextual information, such as scene text, in navigation tasks. Our approach could serve as a starting point for future research in this area and inspire the development of more advanced models that can better use the contextual information available in real-world environments.



(a) Heatmap of ORAR



(b) Heatmap of OAVLN

Fig. 6. Comparison of ORAR and OAVLN models on instruction attention weights. The red text in the x-axis is the object tokens.

VI. Discussion

In this paper, we address the oversight of object tokens in current outdoor VLN models, which leads to navigation failures; we present the OAVLN model incorporating on-the-route object information to improve outdoor VLN performance. Our extensive experiments on two large-scale datasets show that OAVLN outperforms existing methods in both seen and unseen environments. Additionally, we provide visualizations to illustrate how our model learns to pay more attention to objects, leading to a better understanding of the environment and improved navigation ability.

Limitation Our model may still be affected by the data biases in the scene text encoder and object encoder and need further enhancement for better generalization. Additionally, the proposed method requires significant computing resources and training time, which may limit its practical applications. Future research can explore reducing the model's computational cost and training time.

REFERENCES

[1] 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 *CVPR*, 2018.

[2] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," in *EMNLP*, 2020.

[3] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks," in *CVPR*, 2020. [Online]. Available: https://arxiv.org/abs/1912.01734

[4] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "Touchdown: Natural language navigation and spatial reasoning in visual street environments," in *CVPR*, 2019.

[5] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. Salakhutdinov, "Gated-attention architectures for task-oriented language grounding," in *AAAI*, 2018.

[6] J. Xiang, X. Wang, and W. Y. Wang, "Learning to stop: A simple yet effective approach to urban vision-language navigation," in *EMNLP*, 2020. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.62

[7] W. Zhu, X. Wang, T.-J. Fu, A. Yan, P. Narayana, K. Sone, S. Basu, and W. Y. Wang, "Multimodal text style transfer for outdoor vision-and-language navigation," in *EACL*, 2021. [Online]. Available: https://aclanthology.org/2021.eacl-main.103

[8] R. Schumann and S. Riezler, "Analyzing generalization of vision and language navigation to unseen outdoor areas," in *ACL*, 2022.

[9] W. Zhu, Y. Qi, P. Narayana, K. Sone, S. Basu, X. Wang, Q. Wu, M. Eckstein, and W. Y. Wang, "Diagnosing vision-and-language navigation: What really matters," in *NAACL*, 2022. [Online]. Available: https://aclanthology.org/2022.naacl-main.438

[10] E. Chan, O. Baumann, M. Bellgrove, and J. Mattingley, "From objects to landmarks: The function of visual location information in spatial navigation," *Frontiers in Psychology*, vol. 3, 2012. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpsyg.2012.00304

[11] Y. Qi, Z. Pan, S. Zhang, A. van den Hengel, and Q. Wu, "Object-and-action aware model for visual language navigation," in *ECCV*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., 2020.

[12] C. Gao, J. Chen, S. Liu, L. Wang, Q. Zhang, and Q. Wu, "Room-and-object aware knowledge reasoning for remote embodied referring expression," in *CVPR*, 2021.

[13] A. Moudgil, A. Majumdar, H. Agrawal, S. Lee, and D. Batra, "Soat: A scene- and object-aware transformer for vision-and-language navigation," in *NeurIPS*, 2021.

[14] F. Zhu, X. Liang, Y. Zhu, Q. Yu, X. Chang, and X. Liang, "Soon: Scenario oriented object navigation with graph-based exploration," in *CVPR*, 2021.

[15] R. Hu, D. Fried, A. Rohrbach, D. Klein, T. Darrell, and K. Saenko, "Are you looking? grounding to multiple modalities in vision-and-language navigation," in *ACL*, 2019. [Online]. Available: https://aclanthology.org/P19-1655

[16] Y. Zhang, H. Tan, and M. Bansal, "Diagnosing the environment bias in vision-and-language navigation," in *IJCAI*, 2020.

[17] A. Majumdar, A. Shrivastava, S. Lee, P. Anderson, D. Parikh, and D. Batra, "Improving vision-and-language navigation with image-text pairs from the web," in *ECCV*, 2020.

[18] R. Schumann and S. Riezler, "Generating landmark navigation instructions from maps as a graph-to-text problem," in *ACL*, 2021.

[19] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3d: Learning from rgb-d data in indoor environments," *3DV*, 2017.

[20] A. Yan, X. E. Wang, J. Feng, L. Li, and W. Y. Wang, "Cross-lingual vision-language navigation," 2019. [Online]. Available: https://arxiv.org/abs/1910.11301

[21] P. Mirowski, M. K. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, K. Kavukcuoglu, A. Zisserman, and R. Hadsell, "Learning to navigate in cities without a map," in *NeurIPS*, 2018.

[22] H. Mehta, Y. Artzi, J. Baldridge, E. Ie, and P. Mirowski, "Retouchdown: Releasing touchdown on StreetLearn as a public resource for language grounding tasks in street view," in *EMNLP-SpLU*, 2020.

[23] K. Hermann, M. Malinowski, P. Mirowski, A. Banki-Horvath, K. Anderson, and R. Hadsell, "Learning to follow directions in street view," *AAAI*, vol. 34, 2020.

[24] A. B. Vasudevan, D. Dai, and L. Van Gool, "Talk2nav: Long-range vision-and-language navigation with dual attention and spatial memory," *IJCV*, vol. 129, no. 1, 2021.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[26] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," in *NeurIPS*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, 2019.

[27] Y. Qi, Z. Pan, Y. Hong, M. Yang, A. van den Hengel, and Q. Wu, "The road to know-where: An object-and-room informed sequential bert for indoor vision-language navigation," in *ICCV*, 2021.

[28] A. Graves, S. Fernández, and J. Schmidhuber, "Bidirectional lstm networks for improved phoneme classification and recognition," in *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005*, W. Duch, J. Kacprzyk, E. Oja, and S. Zadrożny, Eds., 2005.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7780459

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[31] Z. Kuang, H. Sun, Z. Li, X. Yue, T. H. Lin, J. Chen, H. Wei, Y. Zhu, T. Gao, W. Zhang, K. Chen, W. Zhang, and D. Lin, "MMOCR: A comprehensive toolbox for text detection, recognition and understanding," *CoRR*, vol. abs/2108.06543, 2021. [Online]. Available: https://arxiv.org/abs/2108.06543

[32] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8610–8617.

[33] 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*, 2017, pp. 5998–6008.

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[36] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.

[37] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, "Stanza: A Python natural language processing toolkit for many human languages," in *ACL*, 2020.

[38] J. W. Ratcliff and D. E. Metzener, "Pattern matching: The gestalt approach," *Dr. Dobb's Journal*, vol. 13, no. 7, p. 46, July 1988.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[40] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *ACL*, 2016. [Online]. Available: https://aclanthology.org/P16-1162

[41] V. Levenshtein, "Leveinshtein distance," 1965.

[42] V. Jain, G. Magalhaes, A. Ku, A. Vaswani, E. Ie, and J. Baldridge, "Stay on the path: Instruction fidelity in vision-and-language navigation," in *ACL*, 2019. [Online]. Available: https://aclanthology.org/P19-1181

[43] G. I. Magalhaes, V. Jain, A. Ku, E. Ie, and J. Baldridge, "General evaluation for instruction conditioned navigation using dynamic time warping," in *NeurIPS Visually Grounded Interaction and Language (ViGIL) Workshop*, 2019.