

# Supplementary Materials: Loc4Plan: Locating Before Planning for Outdoor Vision and Language Navigation

Anonymous Authors

## A GENERATION OF SENTENCE-LEVEL LABEL

In this section, we provide a detailed explanation of the generation process for sentence-level labels employed in Eq. (14) as outlined in the main manuscript. Within the hierarchical semantic association learning, we leverage sentence-level labels denoted as  $\tilde{r}_{t,i}$  to establish associations between node states and instructions at the sentence level. However, these sentence-level correspondences are typically absent in existing datasets, and rendering manual annotation is impractical due to its time-intensive nature. Drawing inspiration from prior research works [3, 7] in the indoor vision-and-language navigation (VLN) task, we propose an economically efficient template matching approach to generate pseudo-labels  $\tilde{r}_{t,i}$ , derived from the coarse supervision provided by the correspondence between trajectories and instructions available in the datasets. Please note that these sentence-level labels are not required in the inference stage.

Specifically, the generation process is structured into three distinct steps: instruction stage segmentation, trajectory stage segmentation, and stage matching, as depicted in Figure S1. Initially, we segment the instruction into multiple navigation stages by identifying key phrases that signify state transitions, such as directional changes (e.g., ‘make a right’, ‘turn left’) and stopping actions. A navigation stage is defined as a sub-navigation segment during which the agent moves straightforwardly until a directional change or stop is required. Subsequently, we segment the trajectory into multiple navigation stages by identifying actions that denote state transitions (e.g., ‘RIGHT’, ‘LEFT’, and ‘STOP’), which involves reconstructing the sequence of navigation actions from the trajectory data. Finally, we validate the correspondence between instruction stages and trajectory stages to derive the sentence-level labels. This process entails matching the stages by their numbers and ensuring the consistency of state transition words.

More formally, given the one-to-one correspondence between the instruction and trajectory stages of a successfully matched sample, the sentence-level relevance labels can be calculated as follows:

$$\tilde{r}_{t,i} = \begin{cases} 1, & p_i^s = p_{v_t}^j \\ 0, & p_i^s \neq p_{v_t}^j \end{cases} \quad (1)$$

where  $p_i^s$  denotes the output of instruction segmentation, indicating the index of the navigation stage corresponding to the  $i$ -th sentence in the instruction. And  $p_{v_t}^j$  represents the output of trajectory segmentation, specifying the stage index corresponding to node  $v_t$ . Conversely, if alignment fails for a particular sample, the generation of sentence-level relevance labels is deemed unsuccessful. Considering that the matching confidence varies across samples, we utilize a set of weights  $\gamma_b \in [1, 0.7, 0.5, 0.2]$  to represent different levels of matching confidence. These weights serve to adjust the  $L_{HSA}$  loss (Eq. (14)) during training, with higher weights attributed to samples exhibiting stronger alignment confidence.

## B IMPLEMENTATION DETAILS

Our model is trained on 1 Nvidia GeForce RTX 4090 GPU, and it takes around 13 hours to train the model on each dataset. Following the setting of previous works [6, 8], we use an Adam[4] optimizer with a learning rate of 0.0005 to train the model. We also follow previous works to train the model for 150 epochs and select the model with the highest task completion (TC) performance on the development set for comparisons with other methods.

## C COMPARISONS WITH SOTA METHODS ON ADDITIONAL METRICS

To facilitate a comprehensive comparison between our Loc4Plan method and previous state-of-the-art (SOTA) approaches, we supplement the metrics presented in the main manuscript with results from two additional evaluation metrics, namely Normalized Dynamic Time Warping (nDTW) and Success weighted by normalized Dynamic Time Warping (SDTW). Results on these additional metrics are reported in Tables S1 and S2. The nDTW metric assesses the overlap between the agent’s trajectory and the ground truth across all routes, offering a holistic measure of trajectory fidelity. On the other hand, SDTW, by weighting nDTW with episode success, provides insights into both the success rate and trajectory fidelity, particularly focusing on successful episodes.

As demonstrated in Tables S1 and S2, our Loc4Plan method consistently outperforms competing approaches in terms of both the nDTW and SDTW metrics across the Touchdown and map2seq datasets. These findings substantiate the superiority of our approach in addressing the challenges of outdoor VLN tasks.

## REFERENCES

- [1] Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumathi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. 2018. Gated-attention architectures for task-oriented language grounding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2819–2826.
- [2] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snaveley, and Yoav Artzi. 2019. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 12538–12547.
- [3] Yicong Hong, Cristian Rodriguez Opazo, Qi Wu, and Stephen Gould. 2020. Sub-Instruction Aware Vision-and-Language Navigation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 3360–3376.
- [4] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [5] Piotr Mirowski, Matt Grimes, Mateusz Malinowski, Karl Moritz Hermann, Keith Anderson, Denis Teplyashin, Karen Simonyan, Andrew Zisserman, Raia Hadsell, et al. 2018. Learning to navigate in cities without a map. *Advances in neural information processing systems (NeurIPS)* (2018), 2424–2435.
- [6] Raphael Schumann and Stefan Riezler. 2022. Analyzing Generalization of Vision and Language Navigation to Unseen Outdoor Areas. In *Association for Computational Linguistics (ACL)*. 7519–7532.
- [7] Wang Zhu, Hexiang Hu, Jiacheng Chen, Zhiwei Deng, Vihan Jain, Eugene Ie, and Fei Sha. 2020. Babywalk: Going farther in vision-and-language navigation by taking baby steps. *arXiv preprint arXiv:2005.04625* (2020).
- [8] Wanrong Zhu, Xin Wang, Tsu-Jui Fu, An Yan, Pradyumna Narayana, Kazuo Sone, Sugato Basu, and William Yang Wang. 2021. Multimodal Text Style Transfer for Outdoor Vision-and-Language Navigation. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*. 1207–1221.

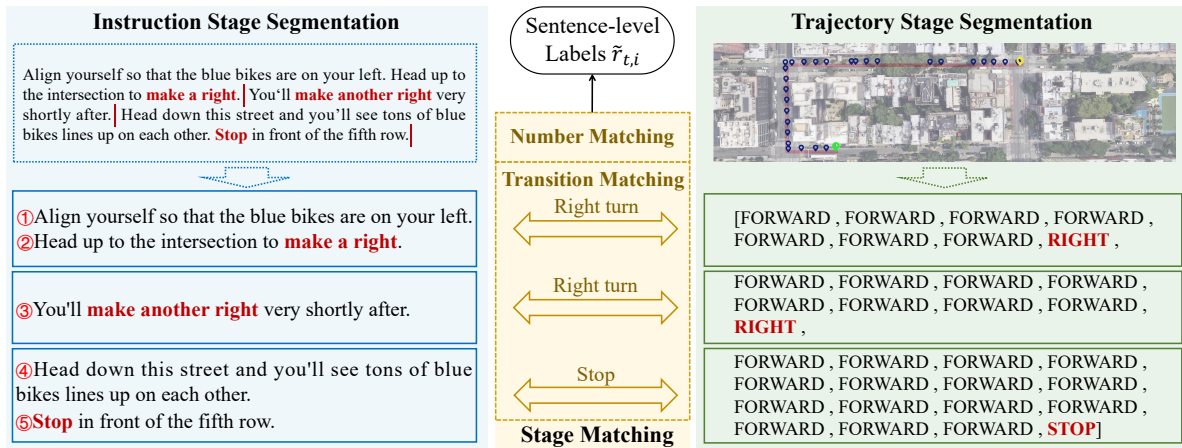


Figure S1: The illustration of sentence-level label generation. ①-⑦ is the index of each sentence in the instruction. By identifying **state transition** in navigation, we divide the instruction and trajectory into several stages. Then we perform stage matching between the instruction stages and trajectory stages to generate sentence-level relevance labels.

**Table S1: Results on Touchdown for the seen and unseen scenarios with nDTW and SDTW metrics.**

	Seen				Unseen			
	dev		test		dev		test	
Model	nDTW↑	SDTW↑	nDTW↑	SDTW↑	nDTW↑	SDTW↑	nDTW↑	SDTW↑
GA[1, 2]	25.2	11.1	24.9	10.9	4.0	1.5	3.3	1.2
RCONCAT[2, 5]	22.5	9.8	22.9	11.1	5.2	3.0	3.9	1.7
VLN Transformer[8]	23.0	12.9	25.3	14.0	4.7	1.9	5.2	2.3
ORAR[6]	45.1	28.3	44.9	27.4	22.2	14.3	21.6	13.6
ours	<b>48.7</b>	<b>32.7</b>	<b>47.9</b>	<b>30.2</b>	<b>28.1</b>	<b>19.2</b>	<b>26.2</b>	<b>17.4</b>

**Table S2: Results on map2seq for the seen and unseen scenarios with nDTW and SDTW metrics.**

	Seen				Unseen			
	dev		test		dev		test	
Model	nDTW↑	SDTW↑	nDTW↑	SDTW↑	nDTW↑	SDTW↑	nDTW↑	SDTW↑
GA[1, 2]	15.2	7.6	13.5	6.7	1.2	0.6	1.5	0.5
RCONCAT[2, 5]	16.7	10.3	13.5	6.7	2.0	1.2	1.2	0.5
VLN Transformer[8]	31.1	17.5	29.5	15.9	6.2	-	6.1	-
ORAR[6]	60.0	41.1	57.8	39.5	41.0	25.8	42.2	28.3
ours	<b>63.2</b>	<b>45.6</b>	<b>60.3</b>	<b>42.8</b>	<b>47.3</b>	<b>32.5</b>	<b>47.2</b>	<b>31.3</b>