Supplementary Material for "Think Global, Act Local: Dual-scale Graph Transformer for Vision-and-Language Navigation"

Shizhe Chen[†], Pierre-Louis Guhur[†], Makarand Tapaswi[‡], Cordelia Schmid[†] and Ivan Laptev[†]
[†]Inria, École normale supérieure, CNRS, PSL Research University [‡]IIIT Hyderabad

https://cshizhe.github.io/projects/vln_duet.html

Section A provides additional details for the model. The experimental setup is described in Section B, including datasets, metrics and implementation details. Section C presents more ablation studies of our DUET model. Section D shows more qualitative examples.

A. Model Details

A.1. Pretraining Objectives

As introduced in Sec 3.3, we employ two auxiliary proxy tasks in pretraining in addition to behavior cloning tasks SAP (single-step action prediction) and OG (object grounding). In the following, we describe the two auxiliary tasks: masked language modeling (MLM) and masked region classification (MRC). The inputs for the two tasks are pairs of instruction $\mathcal{W}$ and demonstration path $\mathcal{P}$.

Masked Language Modeling (MLM) task aims to learn grounded language representations and cross-modal alignment by predicting masked words given contextual words and demonstration path. We randomly replace tokens in $\mathcal{W}$ by a special token `[mask]` with the probability of 15% [1]. Both the coarse-scale encoder and fine-scale encoder can generate contextual word embeddings for masked words as introduced in Sec 3.2.2 and 3.2.3 respectively. The coarse-scale encoder utilizes visual information from an encoded graph at the final step as contexts, while the fine-scale encoder utilizes the last panoramic observation as visual contexts. We average output embeddings of the two encoders for masked words, and employ a two-layer fully-connected network to predict word distributions $p(w_i|\mathcal{W}_{\setminus i}, \mathcal{P})$ where $\mathcal{W}_{\setminus i}$ is the masked instruction and w_i is the label of masked word. The objective of the task is minimizing the negative log-likelihood of original words: $L_{\text{MLM}} = -\log p(w_i|\mathcal{W}_{\setminus i}, \mathcal{P})$.

Masked Region Classification (MRC) aims to predict semantic labels of masked image regions in an observation given an instruction and neighboring regions. As instructions in goal-oriented VLN tasks mainly describe the last observation in the demonstration path, we only apply the MRC task on the fine-scale encoder. We randomly zero out view images and objects in the last observation of $\mathcal{P}$ with the probability of 15%. The target semantic labels for view images are class probability predicted by an image classification model [2] pretrained on ImageNet, while the labels for objects are class probability predicted by an object detector [3] pretrained on VisualGenome. We use a two-layer fully-connected network to predict semantic labels for each masked visual token, and minimize the KL divergence between the predicted and target probability distribution.

A.2. Speaker Model for Data Augmentation

We train a speaker model to synthesize instructions based on visual observations for REVERIE dataset. As REVERIE provides annotated object classes and Matterport3D also contains annotated room classes, we utilize these semantic labels to alleviate the gap between vision and language. Our speaker model consists of a panorama encoder and a sentence decoder. The panorama encoder is fed with image features of the panorama, semantic labels of target object and target room as well as the level of the room. We project all the input features into the same dimension, and utilize a transformer with self-attention to capture relations of each token. The sentence decoder then sequentially generates words conditioning on the encoded tokens. We use LSTM as the decoder and follow the architecture in show-attend-tell image captioning model [8].

Please note that we only employ data in REVERIE training split to learn the speaker model. We initialize the word embeddings in encoder and decoder with pretrained GloVe embeddings [9] and train the speaker model for 50 epochs. We employ the trained speaker model to synthesize instructions for every annotated object in the REVERIE training split, leading to 19,636 instructions in total. We extend the size of the training set from 10,466 instruction-path pairs to 30,102 pairs.

Table 1. Dataset statistics. #house, #instr denote the number of houses and instructions respectively.

VLN Task	Dataset	Train		Val Seen		Val Unseen		Test Unseen	
		#house	#instr	#house	#instr	#house	#instr	#house	#instr
Object-oriented	REVERIE [4]	60	10,466	46	1,423	10	3,521	16	6,292
	SOON [5]	34	2,780	2	113	5	339	14	1,411
Fine-grained	R2R [6]	61	14,039	56	1,021	11	2,349	18	4,173
	R4R [7]	59	233,532	40	1,035	11	45,234	-	-

B. Experimental Setups

B.1. Dataset

We primarily focus our evaluation on goal-oriented VLN benchmarks REVERIE [4] and SOON [5]. To localize target objects in these benchmarks, the agent requires fine-grained object grounding and advanced exploration capabilities. We also test our model on less demanding VLN benchmarks R2R [6] with step-by-step instructions and no object localization. All the benchmarks build upon the Matterport3D [10] environment and contain 90 photo-realistic houses. Each house is defined by a set of navigable locations. Each location is represented by the corresponding panorama image, GPS coordinates and a set of possible actions. We adopt the standard split of houses into *training*, *val seen*, *val unseen*, and *test* subsets. Houses in the *val seen* split are the same as in *training*, while houses in *val unseen* and *test* splits are different from *training*.

Table 1 presents statistics of the three datasets. To be noted, we follow the released challenge split on SOON dataset instead of the split in the original paper [5][1].

B.2. Data Processing for SOON Dataset

The SOON dataset does not provide annotated object bounding boxes per panorama. It only annotates the location of target object bounding boxes for each instruction, including the orientation of object's center point as well as orientation of top left, top right, bottom left, and bottom right corners. The object grounding setting in SOON dataset is to predict the orientation of object's center point. However, we observe that though the annotated objects' center points are of good quality, their annotations of the four corners are quite noisy[2]. Therefore, we propose to clean the object bounding boxes in training and also provide more automatically detected objects as fine-grained visual contexts to represent each panorama.

Specifically, we employ the BUTD detector [3] pre-trained on VisualGenome to detect objects per panorama, which covers 1600 object and scene classes. We filter some unimportant classes for SOON dataset such as 'background', 'floor', 'ceiling', 'wall', 'roof' and so on. We then select one of the detected objects as our pseudo target according to the semantic similarity of object classes and the Euclidean distances of the objects' center points compared to annotated target object. In this way, we convert the object grounding setting in SOON datset similar to the setting in REVERIE dataset, whose goal is to select one object from all candidate objects. In inference, we utilize the orientation of the selected object as our object grounding prediction.

B.3. Evaluation Metrics

Due to the different settings for object grounding in REVERIE and SOON datasets, definitions of success in the two datasets are different. In REVERIE dataset, the success is defined as arriving at a location where the target object is visible and selecting the target object among all annotated candidate objects in the panorama of the location. In SOON dataset, an agent succeeded in carrying out an instruction if it arrives 3 meters near to one of the target locations and the predicted orientation of target object's center point is inside of the annotated polygon of the object in the location.

B.4. Training Details

REVERIE: In pretraining, we combine the original dataset with augmented data synthesized by our speaker model. We pretrain DUET with the batch size of 32 for 100k iterations using 2 Nvidia Tesla P100 GPUs. Then we use Eq. (12) presented in the main paper to fine-tune the policy with the batch size of 8 for 20k iterations on a single Tesla P100. The best epoch is selected by SPL on val unseen split.

SOON: As the size of SOON dataset is much smaller than REVERIE dataset and the instructions are much more complicated, we do not synthesize instructions for SOON dataset. We pretrain model using the original instructions and our automatically cleaned object bounding boxes for 40k iterations with batch size of 32. We fine-tune the model for 40k iterations with batch size of 2 on a single Tesla P100 and select the best model by SPL on val unseen split.

R2R: Following previous works [11–13], we adopt aug-

[1] As shown in https://github.com/ZhuFengdaaa/SOON/issues/1, Zhu *et al*. [5] do not release the split in their original paper. Therefore, performance comparisons on SOON dataset are based on their challenge report https://scenario-oriented-object-navigation.github.io/.

[2] As shown in https://github.com/ZhuFengdaaa/SOON/issues/2, about 50% polygons constructed by the annotated four corners do not contain the objects' center point.

Table 2. Comparison on R4R val unseen split. Methods are grouped according to the used memories: 'Rec' for recurrent state, 'Seq' for sequence and 'Map' for topological map.

Mem	Methods	NE↓	SR↑	CLS↑	nDTW↑	SDTW↑
Rec	SF [14]	8.47	24	30	-	-
	RCM [15]	-	29	35	30	13
	PTA [16]	8.25	24	37	32	10
	RelGraph [17]	7.43	36	41	47	34
	RecBERT [12]	6.67	43.6	51.4	45.1	29.9
Seq	HAMT [13]	6.09	44.6	**57.7**	**50.3**	**31.8**
Map	EGP [18]	8.0	30.2	44.4	37.4	17.5
	SSM [19]	8.27	32	53	39	19
	DUET (Ours)	**5.60**	**50.2**	47.0	42.7	29.3

mented R2R data [11] in pretraining. We pretrain the model for 200k interations with batch size of 64. We fine-tune the model for 20k iterations with batch size of 8.

C. Additional Ablations

C.1. Results on R4R dataset

In Table 2, we further provide results on R4R dataset. The R4R dataset concatenates two paths in R2R dataset whose end and start locations are adjacent, which alleviate the bias of shortest path from the start to the target location. Besides evaluating the navigation success rate (SR), the dataset more focuses on path fidelity metrics to measure the alignment between the predicted and groundtruth paths such as nDTW and SDTW [20]. Our DUET model achieves much better performance on success rate than all existing approaches. However, the map-based methods naturally contain more back-and-forth exploration sub-trajectories in the predicted path, which are harmful to the path fidelity metrics like nDTW. As a result, DUET is inferior to previous methods with local actions in terms of path fidelity metrics, though it still outperforms previous map-based methods with global actions.

C.2. Balance factor λ in fine-tuning objective

Table 3 presents the performance of using different λ in the fine-tuning objective in Eq. (12) of the main paper. The larger λ, the more important of the behavior cloning. We can see that over-emphasizing behavior cloning is harmful to the exploration ability. The model with $\lambda = 1$ achieves the worst OSR and SR. Removing behavior cloning ($\lambda = 0$) achieves good navigation performance such as in OSR, SR and SPL, but it is less competitive in object grounding. We think this is because the agent fails to navigate to target locations in its sampled trajectories, and is unable to train the object grounding module. However, the agent is guaranteed to arrive at target locations in behavior cloning.

Table 3. Ablation of balance factor λ in the fine-tuning loss.

	Navigation			Object Grounding	
	OSR	SR	SPL	RGS	RGSPL
0	**53.00**	**48.22**	33.00	32.12	22.04
0.2	51.07	46.98	**33.73**	32.15	**23.03**
0.5	52.06	46.98	32.38	**32.43**	22.72
1	50.33	45.64	32.54	30.19	21.50

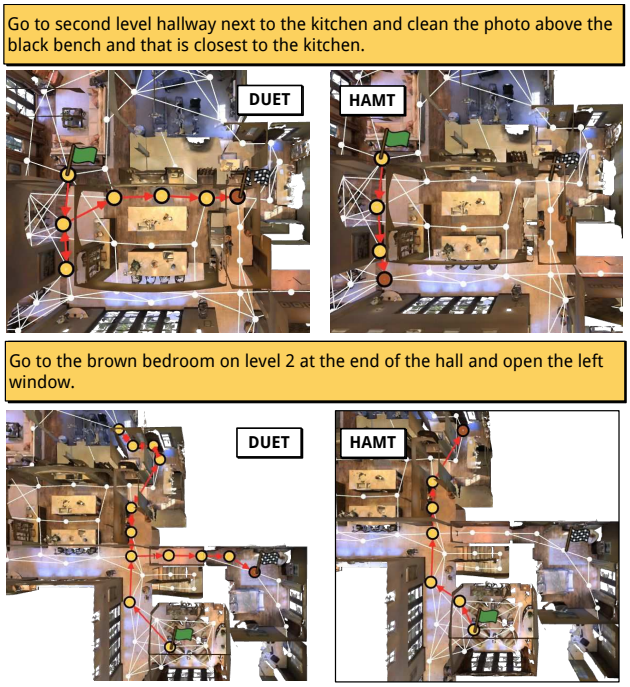

Figure 1. Predicted trajectories of DUET and the state-of-the-art HAMT [13] on REVERIE val unseen split. The green and checkered flags denote start and target locations respectively.

C.3. Backtrack ratio in inference

The backtrack action indicates that the agent does not select a neighboring node from the local action space but jumps to a previously partially observed node through the global action space. We compute the backtrack ratio for DUET. On the REVERIE val seen split, DUET only backtracks in 13.7% of the predicted trajectories; while on the REVERIE val unseen split, DUET backtracks in 48.6% of its predicted trajectories. As the agent has the capacity to memorize house structures in seen environments, it can directly find the target location without much exploration in seen environments. However, when the agent is deployed in unseen environments, it has to explore more to find the target location specified by high-level instructions. When step-by-step instructions are given such as in R2R dataset, we observe the backtrack ratio significantly decreases to 23.2%

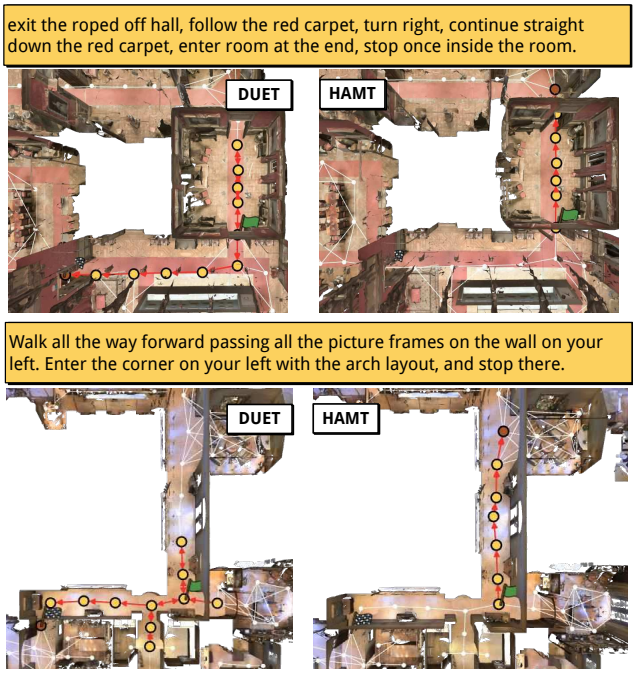

Figure 2. Predicted trajectories of DUET and the state-of-the-art HAMT [13] on R2R val unseen split. The green and checkered flags denote start and target locations respectively.

on val unseen split, which matches our expectation.

C.4. Fusion weights of coarse and fine scales

We observe that the agent typically puts more weights on the fine-scale module in the beginning and at the end of the navigation, and on the coarse-scale module in the middle. Quantitatively, the average weight of the coarse-scale module is 0.36 in the beginning, 0.45 in the middle, and 0.42 at the end. The agent may not need to backtrack at early steps, so it relies more on the local fine-scale module. Then, the agent needs to explore so the global coarse-scale module gets more attention. When deciding where to stop, the agent should identify the target object and the fine-scale module is emphasized again.

C.5. Failure analysis

We perform an additional quantitative evaluation on the REVERIE dataset. For navigation, we measure whether an agent stops at the target room type (*e.g.* a bathroom) or at the correct location. We obtain the following results: (a) incorrect room type: 29.82%; (b) correct room type + incorrect location: 23.20%; (c) correct location: 46.98%. This shows that fine-grained scene understanding remains challenging. With respect to object grounding, once an agent reaches the correct location, the object can be correctly localized 68.43% of the time.

D. Qualitative Examples

Figure 1 visualizes some examples of our DUET and the state-of-the-art HAMT [13] model on REVERIE dataset. In both the cases, the agents explore an incorrect direction in the first attempt. However, DUET is able to efficiently explore another direction towards the goal. Figure 2 shows some examples on R2R dataset. Though step-by-step instructions are provided, the instruction can still be ambiguous. For example, both directions of the start point in the top example of Figure 2 can "exit the rope off hall". DUET is also better at correcting its previous decisions when it finds that the followup instructions do not match with the visual observations.

We further provide some failure cases in REVERIE and R2R datasets in Figure 3. In the top example of Figure 3, there are several bathrooms in the house and our DUET model arrives at one of bathroom. However, the arrived bathroom does not contain the fine-grained objects specified in the instruction. It suggests that our model still needs to improve the fine-grained object grounding capability. The bottom example presents three different instructions for the same trajectory on R2R dataset. The agent succeeds in following the first instruction, but fails for the other two instructions. We observe that the predictions are not very robust across different language instructions.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16× 16 words: Transformers for image recognition at scale. *ICLR*, 2020. 1

[3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018. 1, 2

[4] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*, pages 9982–9991, 2020. 2

[5] Fengda Zhu, Xiwen Liang, Yi Zhu, Qizhi Yu, Xiaojun Chang, and Xiaodan Liang. Soon: Scenario oriented object navigation with graph-based exploration. In *CVPR*, pages 12689–12699, 2021. 2

[6] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation:

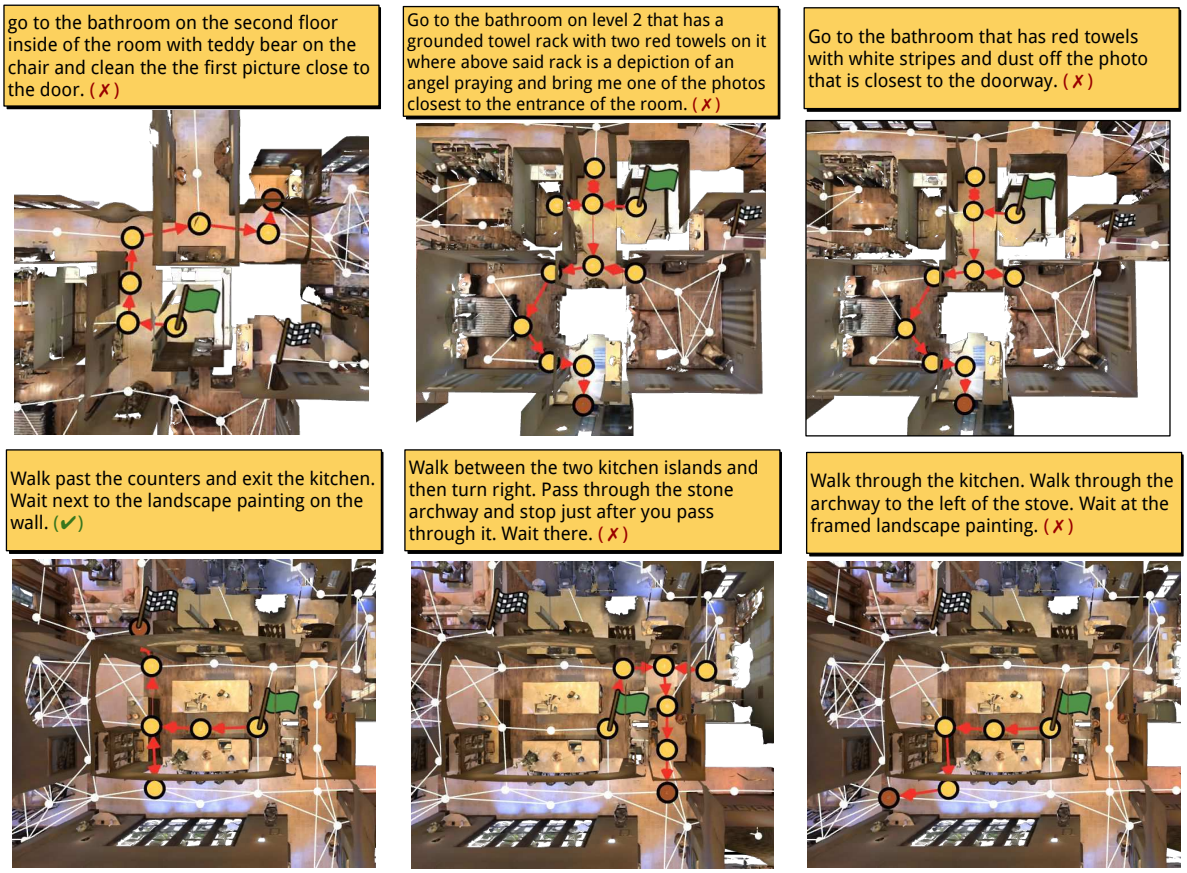

Figure 3. Predicted trajectories of DUET on REVERIE val unseen split (top) and R2R val unseen split (bottom). The green and checkered flags denote start and target locations respectively.

Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683, 2018. 2

[7] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. In *ACL*, pages 1862–1872, 2019. 2

[8] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057. PMLR, 2015. 1

[9] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 1

[10] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *3DV*, pages 667–676. IEEE, 2017. 2

[11] Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *CVPR*, pages 13137–13146, 2020. 2, 3

[12] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln BERT: A recurrent vision-and-language BERT for navigation. In *CVPR*, pages 1643–1653, 2021. 2, 3

[13] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In *NeurIPS*, 2021. 2, 3, 4

[14] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. In *NeurIPS*, pages 3318–3329, 2018. 3

[15] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *CVPR*, pages 6629–6638, 2019. 3

[16] Federico Landi, Lorenzo Baraldi, Marcella Cornia, Massimiliano Corsini, and Rita Cucchiara. Perceive, transform, and

act: Multi-modal attention networks for vision-and-language navigation. *arXiv preprint arXiv:1911.12377*, 2019. 3

[17] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. *NeurIPS*, 33:7685–7696, 2020. 3

[18] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. In *NeurIPS*, volume 33, 2020. 3

[19] Hanqing Wang, Wenguan Wang, Wei Liang, Caiming Xiong, and Jianbing Shen. Structured scene memory for vision-language navigation. In *CVPR*, pages 8455–8464, 2021. 3

[20] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. In *NeurIPS Workshop*, 2019. 3