HOP: History-and-Order Aware Pre-training for Vision-and-Language Navigation

Yanyuan Qiao[1] Yuankai Qi[1] Yicong Hong[2] Zheng Yu[1] Peng Wang[3] Qi Wu[1*]
[1]The University of Adelaide [2]The Australian National University
[3]Northwestern Polytechnical University
{yanyuan.qiao,qi.wu01}@adelaide.edu.au, {qykshr,william.zhengyu}@gmail.com
yicong.hong@anu.edu.au, peng.wang@nwpu.edu.cn

https://github.com/YanyuanQiao/HOP-VLN

Abstract

Pre-training has been adopted in a few of recent works for Vision-and-Language Navigation (VLN). However, previous pre-training methods for VLN either lack the ability to predict future actions or ignore the trajectory contexts, which are essential for a greedy navigation process. In this work, to promote the learning of spatio-temporal visual-textual correspondence as well as the agent's capability of decision making, we propose a novel history-and-order aware pre-training paradigm (HOP) with VLN-specific objectives that exploit the past observations and support future action prediction. Specifically, in addition to the commonly used Masked Language Modeling (MLM) and Trajectory-Instruction Matching (TIM), we design two proxy tasks to model temporal order information: Trajectory Order Modeling (TOM) and Group Order Modeling (GOM). Moreover, our navigation action prediction is also enhanced by introducing the task of Action Prediction with History (APH), which takes into account the history visual perceptions. Extensive experimental results on four downstream VLN tasks (R2R, REVERIE, NDH, RxR) demonstrate the effectiveness of our proposed method compared against several state-of-the-art agents.

1. Introduction

Vision-and-Language Navigation (VLN) has received large attention in communities of computer vision, natural language processing and robotics due to its great importance towards real-world applications such as domestic assistants [3, 5, 7, 17, 28, 29, 38]. VLN requires an agent to navigate to a target location in a 3D simulated environment, according to a given natural language instruction. In the past few years, a great variety of VLN tasks have been proposed, including navigation with low-level instructions

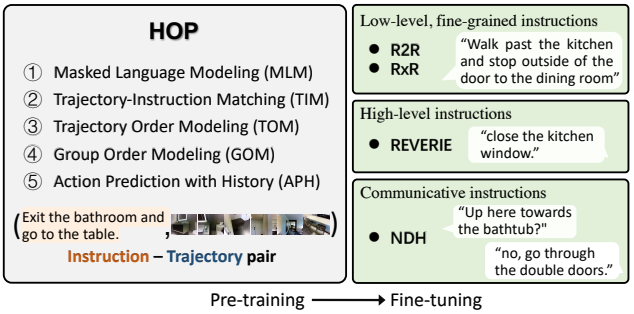

Figure 1. Illustration of the proposed pre-training and fine-tuning paradigm for VLN. The model is pre-trained with five proxy tasks, and fine-tuned on four downstream VLN tasks: R2R, RxR, REVERIE and NDH (detailed in Section 3).

such as R2R [3] and RxR [18], communicative and cooperative instructions such as NDH [36], and high-level instructions for remote object grounding such as REVERIE [32] and SOON [40].

Despite their differences, the agent's navigation is mostly formulated as a sequential text-to-image grounding problem. That is, positioned at a particular node on a predefined connectivity graph, the agent traverses the environment by selecting the adjacent node that has the maximum correspondence between the image representation and the instruction. As a result, the visual-textual matching is considered to be the keystone of addressing VLN tasks.

Inspired by the great success of Vision-Language BERT pre-training on several visual-textual matching tasks, such as image-text retrieval [19] and referring expression grounding [39], several pre-training methods have been proposed for VLN [8, 9, 13, 27]. These approaches are able to achieve better performance, but they still suffer from some limitations. VLN-BERT [27] pre-trains its model by predicting the compatibility of a pair of instruction and visual trajectory. In the downstream tasks, it formulates the navigation as a trajectory selection problem. AirBERT [8] fur-

*Corresponding author

ther adopts a binary classification task to predict whether the given instruction and visual trajectory are paired. Both VLN-BERT and AirBERT discard navigating action prediction during pre-training, weakening the relation between the learned representation and the final goal: navigation action prediction. By contrast, PREVALENT [9] introduces a single-step action prediction task, aiming to learn action-oriented generic visiolinguistic representation, which can be applied to the greedy search VLN. However, PREVALENT largely overlooked the important historical context in pre-training. It only takes the static panoramic image of a single step as visual input, while failing to take into account the history trajectory information. Indeed, VLN is a Partially Observable Markov Decision Process (POMDP), where the agents rely heavily on the past experiences for making future action decisions. Furthermore, VLN is a spatio-temporal task which is sensitive to the sequence order of the trajectory. Thus the ability of temporal order reasoning is also beneficial to the action decision making. Nevertheless, all the above three methods do not explicitly mine temporal order information from either instructions or visual observations.

To address the above mentioned issues, in this work, we propose a novel history-and-order aware pre-training paradigm to enhance the learning of visual-textual correspondence for VLN task. **First**, we provide history visual observations to the action prediction task, called Action Prediction with History (APH), which helps the model locate the sub-instruction to be executed and thus improve the action prediction accuracy. **Second**, we design two order-aware proxy tasks, Trajectory Order Modeling (TOM) and Group Order Modeling (GOM). Given an instruction, TOM requires the model to recover the order of shuffled visual trajectory from a fine-grained level, and GOM requires the model to predict the order of two groups of sub-trajectories from a coarse level. These two tasks explicitly equip the model with the ability to understand the temporal order within instructions, in addition to the visual-textual matching capability. The overall of the proposed pre-training and fine-tuning tasks are illustrated in Figure 1.

To comprehensively evaluate our proposed pre-training methods, we conduct experiments on four downstream tasks: R2R [3], RxR [18], NDH [36], REVERIE [32]. Each task poses a very different challenge to evaluate the agent. R2R serves as an in-domain task, which can verify the agent's generalization ability to unseen environments. The other three tasks are out-of-domain, which are used to study the generalization ability to new tasks. RxR is known for longer instructions. NDH features dialog instructions. REVERIE is characterized by high-level, short instructions. With our proposed pre-training tasks, the fine-tuned downstream model performs favorably on all these tasks: 59% SPL on R2R, 0.33 sDTW on RxR, 3.31 GP on NDH, and 24.34% SPL (14.34% RGSPL) on REVERIE.

2. Related work

In this section, we briefly review several closely related works in VLN and Vision-Language pre-training.

Vision-and-Language Navigation Vision and language navigation task has attracted a lot of attention since it was proposed in the room-to-room task [3]. This task is initially cast as a vision-based sequence-to-sequence trans-coding problem [3]. To improve an agent's generalization ability to unseen environments, a speaker-follower model [7] synthesizes new instructions for data augmentation, and "environmental dropout" method [35] is proposed to mimic unseen environments during training. Wang *et al*. [38] propose a reinforced cross-modal matching framework that combines the strength of reinforcement learning (RL) and imitation learning (IL) for vision language navigation task. To estimate progress made towards the goal, Ma *et al*. [25] introduce a self-monitoring method, which consists of visual-textual co-grounding module and a progress monitor. Hong *et al*. [11] propose a language and visual entity relation graph that exploit the inter and intra modality relationships among the scene. Recently, VLBERT-based methods significantly improve performance on VLN tasks. Hong *et al*. [12] develop a recurrent model that reuses the [CLS] token to maintain the history information. Qi *et al*. [30] propose an object-informed sequential BERT to encode visual perceptions and linguistic instructions.

Vision-Language Pre-training In recent years, many vision-language pre-training works [6, 22, 24, 34] have been proposed to learn cross-modal representations for various vision language problems, such as Image-Text Retrieval [19], Referring Expression Grounding [39] and Visual Question Answering [4]. Unlike conventional vision-language pre-training, VLN tasks additionally require the learned representations to facilitate action decision making. VLN-BERT [27] performs path selection by predicting instruction and trajectory compatibility. Similarly, AirBERT [8] trains the path-instruction matching task by collecting large numbers of indoor image-caption pairs. However, these methods ignore the importance of dynamic action decision. Although PREVALENT [9] adds the task of action prediction (AP), the global-level path information cannot be considered because its input is a global instruction and a panoramic image of the current location, which discards temporal visual context for action prediction. Moreover, all these methods do not explicitly mine temporal order information from either instructions or visual observations, which are crucial for an agent to predict actions. In contrast, our pre-training is designed to relieve the above mentioned limitations, by introducing a *history-aware* proxy task and two *order-aware* proxy tasks. These

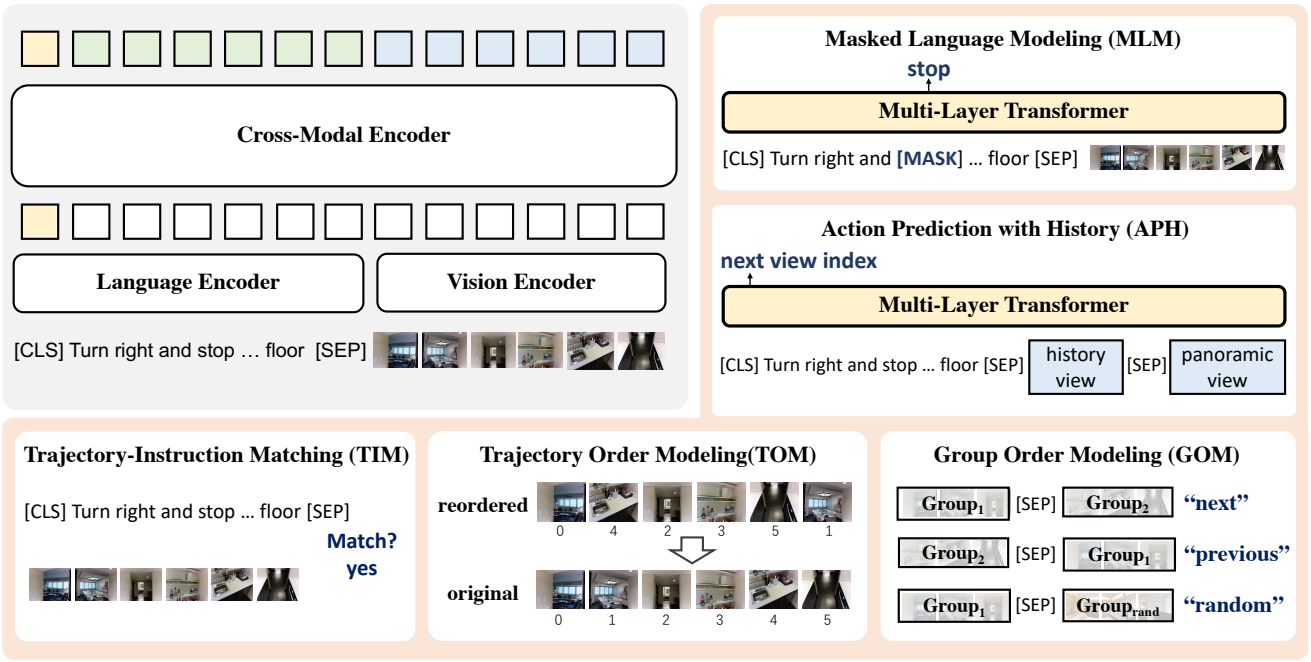

Figure 2. The main architecture of our pre-training model and five proxy tasks.

tasks help the agent understand the history context and temporal order, facilitating the final action prediction. We note ALTR [13] introduces a "Next Visual Scene" task to capture the temporal context, but instead of predicting image orders it predicts visual features of future steps directly, which is a much harder task.

3. Method

In this section, we first present the preliminaries of VLN to put our method in a proper context. Then, we describe the model architecture we adopted. Next, we provide the details of the five pre-training proxy tasks utilized in our proposed history-and-order aware pre-training paradigm. At last, we introduce the datasets used for pre-training.

In VLN, a robot agent is randomly positioned in a 3D simulator with the initial state $\langle u_0, \alpha_0, \beta_0 \rangle$, where u_0 is the starting viewpoint on the pre-defined navigation-graph, α_0 and β_0 are angles of heading and elevation. Given a natural language instruction $x = \langle x_1, x_2, \dots x_L \rangle$, where L is the instruction length, the agent needs to predict the next navigation action according to panoramic perceptions. Following the common practice, the action is predicted by selecting a navigable location from a candidate set provided by the simulator. Each navigable location is represented by its RGB image feature and its orientation feature.

3.1. Model Architecture

The model architecture is illustrated on the top-left of Figure 2, which is similar to LXMERT [34]. Taking the instruction-trajectory pair as input, the model first utilizes a language encoder and a vision encoder to extract single-modal representations from the instruction and image sequence, respectively. Then, these representations are fed into a cross-modal encoder to implement interactions between the two modalities and generate the final fused representations.

Language Encoder We first use WordPieces [15] to tokenize all words in an instruction, obtaining a sequence of tokens: $\texttt{[CLS]}, w_1, w_2, \dots, w_L, \texttt{[SEP]}$, where [CLS] and [SEP] are added special tokens. Then, the text embedding of each token is obtained via summing up the token embedding and the position embedding, followed by Layer Normalization (LN). At last, the text embedding is passed through the single-modal language encoder, of which each layer consists of a self-attention sub-layer and a feed-forward sub-layer. The outputs of the language encoder are used as language features.

Vision Encoder Trajectory $\tau = \langle v_1, v_2, \dots v_T \rangle$ represents the image sequences observed by the agent when traversing the environment, where v_i is the observed image of the environment at step i and T is the number of total steps. To better capture order information from the trajectory, we use the front view image of the agent's observation at each position, rather than using the panoramic image. This is because panoramic images of the adjacent observation points in the same room are similar, causing difficulties for the agent to explore the dynamic and temporal information of the entire trajectory.

We first use ResNet-152 [10] pre-trained on ImageNet [33] to extract a 2048-dimensional image feature vector v_{vis} for each front view image v_i. Then, we com-

pute the orientation feature of heading α and elevation β as $[\sin\alpha; \cos\alpha; \sin\beta; \cos\beta]$, and repeat it for 32 times to constitute a 128-dimensional direction feature vector v_d as same as [35]. Each image v_i in the trajectory is finally represented by a 2176-dimensional feature vector $v_i = [v_{vis}; v_d]$ by concatenating v_{vis} and v_d. At last, the image features of trajectory τ are passed through the single-modal vision encoder, of which each layer consists of a self-attention sub-layer and a feed-forward sub-layer. The outputs of the vision encoder are used as vision features.

Cross-Modal Encoder We use the Cross-Modal Encoder to fuse features from both language and vision modalities. For the cross-modal encoder, each layer contains two self-attention sub-layers, one bi-directional cross-attention sub-layer and two feed-forward sub-layers. The outputs of the cross-modal encoder are used as cross-modal features for pre-training and downstream tasks.

Following [9], we set the layers' number N_{text}, N_{image}, N_{cross} of text encoder, vision encoder and cross-modal encoder to 9, 1 and 3, respectively.

3.2. Pre-training Tasks

Masked Language Modeling (MLM) MLM is the most commonly used proxy task for BERT-based pre-training. For VLN pre-training, the goal of MLM is to recover masked words w_m via reasoning over the surrounding words $w_{\setminus m}$ and the trajectory τ. Specifically, the inputs for MLM are the instruction $w = \langle w_1, w_2, \ldots, w_L \rangle$ and the corresponding trajectory $\tau = \langle v_1, v_2, \ldots, v_T \rangle$. We randomly mask out the input tokens of the instruction with a 15% probability, and replace the masked token w_m with a special token `[mask]`. This task is optimized by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{MLM}}(\theta) = -\mathbb{E}_{(w,\tau)\sim D} \log P_\theta(w_m | w_{\setminus m}, \tau), \quad (1)$$

where θ denotes trainable parameters. Each pair (w, τ) is sampled from the training set D.

Trajectory-Instruction Matching (TIM) TIM is a global matching task, which is designed to predict whether a given image trajectory and an instruction are a matched pair. The inputs for TIM are the instruction-trajectory pairs (w, τ). During training, we generate negative samples by randomly replacing the trajectory with a mis-matched one, with a probability of 50%. Specifically, the generated negative samples are selected only from the same environment, so that the model could focus on distinguishing between paths rather than environments. We use the output representation of the special token of `[CLS]` as the joint representation of the instruction-trajectory pair, and then feed it into an FC layer with a sigmoid function to predict the matching score $s_\theta(w, \tau)$. We optimize this task via the binary cross-entropy loss:

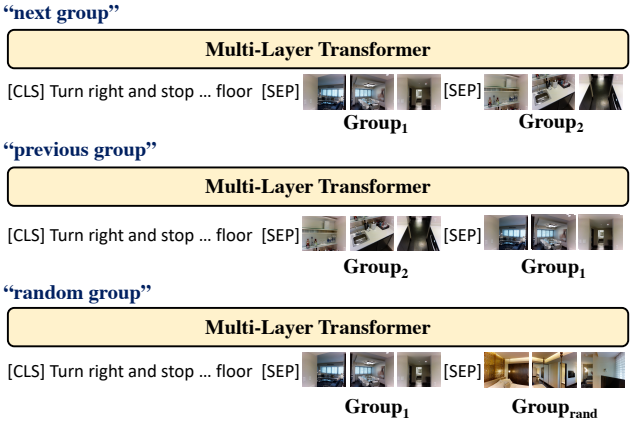

Figure 3. Illustration of Group Order Modeling (GOM).

$$\mathcal{L}_{\text{TIM}}(\theta) = -\mathbb{E}_{(w,\tau)\sim D}[y \log P_\theta + (1-y) \log P_\theta], \quad (2)$$

where $P_\theta = s_\theta(w, \tau)$, and $y \in \{0, 1\}$ indicates whether the sampled trajectory-instruction pair is a match.

Trajectory Order Modeling (TOM) VLN is sensitive to the sequence order of trajectory, thus we design the TOM task to enable the model to learn the temporal order within instructions in addition to visual-textual correspondence. The inputs for TOM are the instruction w and the reordered trajectory τ'. Specifically, we randomly selected 50% images of the original trajectory τ for shuffling. The goal of TOM is to reconstruct the correct order $r = \langle r_1, r_2, ..., r_N \rangle$ of the original trajectory τ with reference to the given instruction w, where N is the number of steps of trajectory. This task is formulated as a classification problem of N classes. We feed the vision output of the cross-modal encoder into an FC layer with softmax to predict the order r'_k for each image k in the reordered trajectory τ', by minimizing the cross-entropy loss:

$$\mathcal{L}_{\text{TOM}}(\theta) = -\mathbb{E}_{(w,\tau')\sim D} \sum_{i=1}^{N} y_i \log P_\theta(r'_k | w, \tau'), \quad (3)$$

where $y_i = 1$ if the predicted order r'_k for image k is the original order i, otherwise $y_i = 0$.

Group Order Modeling (GOM) This task shares the same motivation as TOM but at a sub-trajectory level. It predicts previous, next, or random relations between two sub-trajectories. As shown in Figure 3, the inputs for GOM are the instruction w and image sequence group (G_1, G_2) that is derived from the trajectory τ. Specifically, we divide the trajectory into two parts (G_1, G_2) sequentially in an even way. Furthermore, there is a probability of $1/3$ that G_2 will be placed after G_1, a probability of $1/3$ that G_2 will be placed before G_1, and a remaining probability of

$1/3$ that G_2 will be replaced by random sampling from image sequence groups of other trajectories. This task is cast as a classification problem of three classes. If G_1 happens before G_2, we denote it as $c = 1$; if G_1 happens after G_2, we denote it as $c = 2$; if G_2 is an image sequence group randomly sampled from different environments, we denote it as $c = 3$. The special token `[SEP]` is used to separate the two groups. We use the representation of `[CLS]` token as the joint embedding of the input visual and textual information. Then we apply an FC layer with softmax to make a three-class prediction of c'. This task is optimized by minimizing the cross-entropy loss:

$$\mathcal{L}_{\text{GOM}}(\theta) = -\mathbb{E}_{(w,(G1,G2))\sim D} \sum_{c} y_c \log P_\theta(c'|w,(G1,G2)), \tag{4}$$

where $y_c \in \{0,1\}$ indicates whether the predicted class c' is the desired class c or not.

Action Prediction with History (APH) The motivation of this task is to make the learned representation benefiting the final goal: predicting navigation action. The inputs for APH are the instruction w, the history trajectory $\tau_{t-1} = \langle v_1, v_2, \ldots, v_{t-1} \rangle$, and the panoramic view $\boldsymbol{p} = \{p^1, p^2, \ldots, p^{36}\}$ of the current step t. The panoramic view consists of 36 images from 12 surrounding angles, each with 3 camera poses (up, down, horizon). As in PREVALENT [9], action decision is achieved by selecting next view image v'_{t+1} from the candidate views (*i.e.* panorama observation v_{t_p}), which can be expressed as a classification problem. The output on the special token `[CLS]` represents a fused representation of the two modalities. We apply an FC layer to the representation of `[CLS]` to predict the next view v'_{t+1}. We optimize this task via a cross-entropy loss:

$$\mathcal{L}_{\text{APH}}(\theta) = -\mathbb{E}_{(w,\tau,v_{pano})\sim D} \sum_{p} y_p \log P_\theta(v'_{t+1}|w,\tau_{t-1},v_{t_p}), \tag{5}$$

where p represents labels of the 36 images in the panoramic view image, and $y_p \in \{0,1\}$ indicates whether the predicted next view image v'_{t+1} is the desired next view image of label p or not.

3.3. Pre-training Datasets

We construct our pre-training dataset based on existing datasets: PREVALENT [9] and BnB [8]. PREVALENT uses a pre-trained speaker model to produce more instructions to augment R2R dataset. It contains 104K original R2R samples and 6482K synthesized samples. BnB dataset collects image-caption pairs from Airbnb. We use raw images and captions from the BnB dataset and reprocessed them. Indeed, nearly half of the BnB images are captionless (*i.e.* images without captions). Thus, to better adapt BnB dataset to the designed pre-training tasks such as Trajectory Order Modeling, we remove these captionless images. To construct path-instruction pairs, we concatenate the images and concatenate the corresponding captions. Each path contains 5-7 images, which is consistent with the R2R dataset. For image features, we used a Resnet-152 network pre-trained on ImageNet to extract a mean-pooled feature vector, same as the encoding method of images in Matterport3D. Our processed BnB data contains 342K image sequence-caption pairs.

4. Experiments

In this section, we conduct comprehensive experiments on several downstream VLN tasks and provide detailed ablation studies to validate the effectiveness of our proposed method.

4.1. Downstream tasks

We focus on four downstream VLN tasks that are all based on the Matterport3D simulator [3]: Room-to-Room (R2R) [3], Room-across-Room (RxR) [18], Navigation from Dialog History (NDH) [36] and REVERIE [32]. R2R serves as an in-domain task, and the other three serve as out-of-domain tasks. These tasks have different characteristics and evaluate the agent from different views.

- R2R and RxR are VLN tasks with low-level, fine-grained instructions that aim at verifying the agent's ability to generalize to unseen environments.
- REVERIE is a VLN task with high-level instructions, focusing on grounding remote target object.
- NDH is a VLN task that uses indirect instructions such as dialog history, which can be used to study the agent's generalization ability to new tasks.

4.2. Implementation Details

Pre-training We use 4 Tesla V100 GPUs for pre-training. The batch size for each GPU is set to 128. AdamW [23] optimizer is adopted and the learning rate is set to 5×10^{-5}. The model is trained for 15 epochs. We conduct task sampling training for each mini-batch. For each mini-batch, we choose only one of the five proxy tasks to train the model.

Fine-tuning Different from PREVALENT [9] that only uses the pretrained language representations to finetune downstream tasks, we use an architecture similar to RecBERT [27] as our baseline for finetune, of which both the image and language representations could be used for downstream tasks. Following RecBERT, we employ a recurrent function to update state `[CLS]` and use its attention distribution on navigation candidates to determine the next action. For more details please refer to [12]. For R2R task, we set the batch size to 16 and the learning rate to 1×10^{-5}. Following previous works [12], we use both the original training data of R2R and the augmented data from PREVALENT [9] to train the agent. For both NDH and REVERIE tasks, we set the batch size to 8 and the learning

Methods	R2R Validation Seen				R2R Validation Unseen				R2R Test Unseen			
	TL	NE ↓	SR ↑	SPL ↑	TL	NE ↓	SR ↑	SPL↑	TL	NE ↓	SR ↑	SPL ↑
SF [7]	-	3.36	66	-	-	6.62	35	-	14.82	6.62	35	28
RCM [38]	10.65	3.53	67	-	11.46	6.09	43	-	11.97	6.12	43	38
Regretful [26]	-	3.23	69	63	-	5.32	50	41	13.69	5.69	48	40
Fast-short [16]	-	-	-	-	21.17	4.97	56	43	22.08	5.14	54	41
EnvDrop [35]	11.00	3.99	62	59	10.70	5.22	52	48	11.66	5.23	51	47
OAAM [31]	10.20	-	65	62	9.95	-	54	50	10.40	5.30	53	50
EntityGraph [11]	10.13	3.47	67	65	9.99	4.73	57	53	10.29	4.75	55	52
NvEM [1]	11.09	3.44	69	65	11.83	4.27	60	55	12.98	4.37	58	54
ActiveVLN [37]	19.70	3.20	70	52	20.6	4.36	58	40	21.6	4.33	60	41
Press [21]	10.57	4.39	58	55	10.36	5.28	49	45	10.77	5.49	49	45
PREVALENT [9]	10.32	3.67	69	65	10.19	4.71	58	53	10.51	5.30	54	51
RecBERT [12]	11.13	2.90	72	68	12.01	3.93	**63**	**57**	12.35	4.09	**63**	57
AirBERT [8]	11.09	**2.68**	75	**70**	11.78	4.01	62	**56**	12.41	4.13	62	57
HOP (0)	10.75	3.50	66	63	11.80	4.74	54	49	12.53	4.93	55	50
HOP (1)	11.51	**2.46**	**76**	**70**	12.52	**3.79**	**64**	**57**	13.29	**3.87**	**64**	**58**
HOP (2)	11.26	2.72	**75**	**70**	12.27	**3.80**	**64**	**57**	12.68	**3.83**	**64**	**59**

Table 1. Comparison with state-of-the-art methods on R2R. First group are no pre-training methods. The second group are existing pre-training-based methods. The third group are our methods. HOP (0) denotes our baseline model without pre-training. HOP (1) denotes finetuned model pre-trained on the same data as PREVALENT. HOP (2) denotes finetuned model pre-trained on data of both PREVALENT and our processed data from BnB. **Blue** and **Black** denote the best and runner-up results, respectively.

rate to 1×10^{-5}. All the above three downstream tasks are finetuned on a single 1080Ti GPU. For RxR task which has longer instructions, we finetune on a single V100 GPU, and we set the batch size to 16 and the learning rate to 7×10^{-6}.

4.3. Results

For each of the four downstream VLN tasks, we first introduce the evaluation metrics used for the task, and then we compare our method with SoTA methods. Specifically, we report results of our method on three settings: (I) baseline results without pre-training, which is denoted as HOP (0); (II) finetuned results with pre-training only using data of PREVALENT, which is denoted as HOP (1); (III) finetuned results with pre-training using data of PREVALENT and processed data from BnB, which is denoted as HOP (2).

4.3.1 Room-to-Room (R2R)

Evaluation Metrics Four commonly used metrics are adopted: Trajectory Length (TL) that measures the average length of the navigation trajectory in meters; Navigation Error (NE) which is the mean of the shortest path distance in meters between the agent's stop location and the target location; Success Rate (SR) that measures the ratio of successful tasks, of which the agent's stop position is less than 3 meters away from the target position; Success rate weighted by Path Length (SPL) [2] that measures both the accuracy and efficiency of navigation. SPL is the key metric for R2R.

Comparison with SoTA Table 1 presents the results on R2R task. It shows that our model outperforms other methods on all data splits and metrics. Specifically, our method achieves better performance than other SoTA methods, such as RecBERT and AirBERT. Our method outperforms them with a large margin of 2% in terms of the main metric SPL on the test unseen split. Note that RecBERT is initialised from pre-trained model of PREVALENT and shares the same architecture as ours. This indicates that our pre-training can effectively improve the navigation ability of the agent. If removing our pre-training, we observe significant performance drop on all metrics, as shown by the results of HOP (0). In particular, it drops 9% of SR and 9% of SPL on the test unseen split.

Methods	NDH Validation Unseen			NDH Test Unseen		
	Oracle	**Navigator**	**Mixed**	**Oracle**	**Navigator**	**Mixed**
Seq2Seq [36]	1.23	1.98	2.10	1.25	2.11	2.35
CMN [41]	2.68	2.28	2.97	2.69	2.26	2.95
PREVALENT [9]	2.58	2.99	3.15	1.67	2.39	2.44
ORIST [30]	3.30	3.29	3.55	2.78	3.17	3.15
HOP (0)	3.08	3.10	3.38	2.05	2.12	2.26
HOP (1)	**3.96**	**3.99**	**4.37**	**2.92**	**3.20**	**3.31**
HOP (2)	**4.07**	**4.05**	**4.41**	**2.99**	**3.18**	**3.24**

Table 2. Comparison with state-of-the-art methods on NDH measured by the Goal Progress (m).

4.3.2 Navigation from Dialog History (NDH)

Evaluation Metric NDH evaluates the performance using Goal Progress (GP) in meters, which measures the average progress of the agent towards the target location.

Comparison with SoTA The results are shown in Table 2. Our model outperforms the SoTA method ORIST

Methods	REVERIE Validation Seen						REVERIE Validation Unseen						REVERIE Test Unseen					
	Navigation				RGS↑	RGSPL↑	Navigation				RGS↑	RGSPL↑	Navigation				RGS↑	RGSPL↑
	SR↑	OSR↑	SPL↑	TL			SR↑	OSR↑	SPL↑	TL			SR↑	OSR↑	SPL↑	TL		
Human	–	–	–	–	–	–	–	–	–	–	–	–	81.51	86.83	53.66	21.18	77.84	51.44
RCM [38]	23.33	29.44	21.82	10.70	16.23	15.36	9.29	14.23	6.97	11.98	4.89	3.89	7.84	11.68	6.67	10.60	3.67	3.14
SMNA [25]	41.25	43.29	39.61	7.54	30.07	28.98	8.15	11.28	6.44	9.07	4.54	3.61	5.80	8.39	4.53	9.23	3.10	2.39
FAST-Short [16]	45.12	49.68	40.18	13.22	31.41	28.11	10.08	20.48	6.17	29.70	6.24	3.97	14.18	23.36	8.74	30.69	7.07	4.52
FAST-MATTN [32]	50.53	55.17	45.50	16.35	31.97	29.66	14.40	28.20	7.19	45.28	7.84	4.67	19.88	30.63	11.61	39.05	11.28	6.08
ORIST [30]	45.19	49.12	42.21	10.73	29.87	27.77	16.84	25.02	15.14	10.90	8.52	7.58	22.19	29.20	18.97	11.38	10.68	9.28
RecBERT [12]	51.79	53.90	47.96	13.44	38.23	**35.61**	30.67	35.02	24.90	16.78	**18.77**	15.27	29.61	32.91	23.99	15.86	16.50	13.51
AirBERT [8]	47.01	48.98	42.34	15.16	32.75	30.01	27.89	34.51	21.88	18.71	18.23	14.18	30.28	34.20	**23.61**	17.91	16.83	13.28
HOP (0)	43.78	46.03	40.11	11.67	28.95	26.69	24.17	30.16	20.07	16.52	12.35	10.18	23.12	26.27	18.5	16.15	11.17	9.1
HOP (1)	54.81	56.08	**48.05**	14.05	**40.55**	**35.79**	30.39	35.30	**25.10**	17.16	18.23	**15.31**	29.12	32.26	23.37	17.05	**17.13**	**13.90**
HOP (2)	53.76	54.88	**47.19**	13.80	**38.65**	33.85	31.78	36.24	**26.11**	16.46	**18.85**	**15.73**	30.17	33.06	**24.34**	16.38	**17.69**	**14.34**

Table 3. Comparison with the state-of-the-art methods on REVERIE. SPL is the main metric for its navigation sub-task, and RGSPL is the main metric for the REVERIE task.

[30] on both validation and test unseen environments in all settings. Particularly, our HOP (2) achieves up to 1 meter improvement over ORIST on validation unseen split under the mixed setting. Our method also performs much better than the pre-training method PREVALENT (about 1.3 meter improvement on the Test split under the Oracle setting). These results demonstrate the effectiveness and generalization ability of our pre-trained model.

4.3.3 REVERIE

Evaluation Metrics REVERIE uses the same metrics as R2R to evaluate its navigation sub-task. Additionally, Oracle Success Rate (OSR), Remote Grounding Success rate (RGS) and RGS weighted by Path Length (RGSPL) are adopted. OSR measures the ratio of tasks of which one of its trajectory viewpoints can observe the target object within 3 meters. RGS measures the ratio of tasks that successfully locate the target object. RGSPL is RGS weighted by Path Length, which is the main metric for this task.

Comparison with SoTA The results are presented in Table 3. It shows that our model outperforms previous methods on all the splits in terms of the main metric SPL for navigation sub-task and in terms of the main metric RGSPL for the entire REVERIE task. In addition, we also note that on the Test split, although our method performs slightly ($\downarrow$ 0.1) worse than AirBERT according to SR for navigation, our SPL and RGSPL results are significantly better ($\uparrow$ 1.1) than AirBERT. This indicates our method is more efficient on navigation and object grounding.

4.3.4 Room-Across-Room (RxR)

Evaluation Metrics In addition to the aforementioned SR and SPL metrics, RxR also adopts normalized Dynamic Time Warping (nDTW) [14] and success rate weighted by Dynamic Time Warping (sDTW) for evaluating performance. These two metrics are designed to measure the path fidelity compared to the ground truth path.

Methods	RxR Validation Seen				RxR Validation Unseen			
	SR↑	SPL ↑	nDTW ↑	sDTW ↑	SR↑	SPL ↑	nDTW ↑	sDTW↑
Baseline [18]	28.6	-	0.45	0.23	26.1	-	0.42	0.21
EnvDrop [35]	48.1	0.44	0.57	**0.40**	38.5	0.34	0.51	0.32
+Syntax [20]	48.1	0.44	**0.58**	**0.40**	39.2	0.35	**0.52**	0.32
HOP (0)	42.0	0.41	0.51	0.34	36.3	0.31	0.48	0.29
HOP (1)	48.3	**0.45**	0.57	**0.40**	42.1	**0.36**	0.51	**0.33**
HOP (2)	**49.4**	**0.45**	**0.58**	**0.40**	**42.3**	**0.36**	**0.52**	**0.33**

Table 4. Comparison with state-of-the-art methods on RxR using English instructions.

Comparison with SoTA As shown in Table 4, our model performs favourably against SoTA methods in terms of all metrics. In particular, our model achieves 3.1% improvement in SR over previous SoTA on the unseen split.

4.4. Ablation Study

The Effect of Pre-training Tasks To evaluate the effectiveness of different pre-training tasks, we conduct an ablation study on R2R, REVERIE and NDH validation unseen set. The results are presented in Table 5.

First, we evaluate the effect of the generic MLM task alone. Model 1 shows the results of the baseline model directly trained on downstream VLN tasks without any pre-training. Model 2 shows the results when only applying MLM for pre-training. MLM brings large improvements on all downstream VLN tasks, especially on the R2R task ($\uparrow$ 6% SR).

Secondly, we evaluate the effect of our proposed pre-training tasks that are specifically designed for VLN, by combining these tasks with MLM during pre-training. Model 3$\sim$ Model 6 present the results of combining MLM with the task of TIM, TOM, GOM, and APH respectively. The result shows that all these four proxy tasks can further improve the navigation performance. Among these four tasks, APH contributes the most, then TOM, GOM, and TIM. This demonstrates that action prediction with history information indeed helps learn better representations.

Furthermore, we find that these proxy tasks are complementary. When we combine these tasks together, the com-

Pre-training Data		Pre-training Tasks	R2R		REVERIE					NDH
			SR	SPL	SR	OSR	SPL	RGS	RGSPL	Goal Progress
None	1	None	54.19	49.35	24.17	30.16	20.07	12.35	10.18	3.38
PREVALENT	2	MLM	60.75	54.81	27.18	31.84	21.83	15.31	12.48	3.76
	3	MLM + TIM	61.52	55.31	28.06	34.76	22.84	16.30	13.29	3.86
	4	MLM + TOM	61.81	54.96	28.57	31.27	22.67	17.98	14.49	3.88
	5	MLM + GOM	61.98	55.22	27.83	35.08	22.53	17.24	14.14	3.84
	6	MLM + APH	62.01	56.13	29.37	34.88	23.76	17.52	14.16	3.88
	7	MLM + AP	61.27	55.68	28.69	33.20	23.25	16.76	13.64	3.83
	8	MLM + TIM + TOM + GOM	63.09	56.61	29.99	35.13	24.66	18.03	15.06	4.04
	9	MLM + TIM + TOM + GOM + APH	**63.86**	**57.07**	**30.39**	**35.30**	**25.10**	**18.35**	**15.31**	**4.37**
PREVALENT + BnB*	10	MLM + TIM + TOM + GOM + APH	**63.52**	**57.22**	**31.78**	**36.24**	**26.11**	**18.85**	**15.73**	**4.41**

Table 5. Ablation study of the pre-training tasks and data. We use the task of R2R, REVERIE and NDH as benchmarks. Where BnB* represents our processed data from BnB dataset.

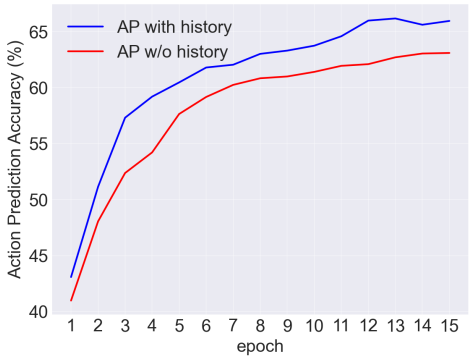

Figure 4. APH *v.s.* AP regarding action prediction accuracy.

bined scores (Model 8 and Model 9) are much higher than the separate scores (Model 3∼ Model 6). We observe a gain of 9% SR on R2R, 6% SR on REVERIE, 1 meter on NDH compared to the baseline model without pre-training.

The Effect of Pre-training Data As shown in Model 10 of Table 5, the model pre-trained with data from both PREVALENT and BnB* (our processed BnB data) achieves the best performance. We find that the model achieves a significant improvement on REVERIE task when pre-trained with additional BnB* data than only with PREVALENT data, while keep competitive on R2R and NDH task. This could be because BnB's captions primarily describe rooms and objects, which matches REVERIE's mission of grounding objects.

The Effect of History Information in APH We also pre-train our model using the Action Prediction task without history to verify the importance of referring history information for decision making. As shown in the validation curves of Action Prediction of Figure 4, our APH converges faster than AP during pre-training, and results in higher accuracy. In addition, as shown in Table 5 (Model 6 and Model 7), Action Prediction with History achieves better performance on all three downstream tasks and all metrics than without history. Therefore, the history information is beneficial to the vision-language pre-training for VLN tasks.

4.5. Limitations and Future work

Like most of existing pre-training methods, the training of our model also requires a large number of computational resources. In the future, we will explore more efficient model architectures. In addition, our current work focuses on indoor and navigation graph based environments. We can devote efforts on outdoor and/or continuous environments in the future.

5. Conclusion

In this paper, we present a history-and-order aware pre-training (HOP) paradigm to solve the issues existing in the previous Vision-and-Language Navigation pre-training methods. We first carefully examine and compare previous methods, and we find these methods either overlook the important historical context in pre-training or neglect the role of the action ordering. Thus, we design an Action Prediction with History (APH) task that provides history visual observations to the action prediction in the pre-training. We then propose two order-aware proxy tasks, including Trajectory Order Modeling (TOM) and Group Order Modeling (GOM). These two tasks equip the pre-trained model with the ability to understand the temporal order within instructions, in addition to the widely used visual-textual matching capability. Extensive experimental results on four mainstream downstream VLN tasks, including R2R, RxR, NDH and REVERIE, demonstrate the effectiveness of our proposed VLN pre-training paradigm, the HOP.

References

[1] Dong An, Yuankai Qi, Yan Huang, Qi Wu, Liang Wang, and Tieniu Tan. Neighbor-view enhanced model for vision and language navigation. In *ACM MM*, 2021. 6

[2] Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir Roshan Zamir. On evaluation of embodied navigation agents. *CoRR*, abs/1807.06757, 2018. 6

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683, 2018. 1, 2, 5

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. In *ICCV*, pages 2425–2433, 2015. 2

[5] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. TOUCHDOWN: natural language navigation and spatial reasoning in visual street environments. In *CVPR*, pages 12538–12547, 2019. 1

[6] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: universal image-text representation learning. In *ECCV*, pages 104–120, 2020. 2

[7] 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. 1, 2, 6

[8] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. In *ICCV*, pages 1634–1643, 2021. 1, 2, 5, 6, 7

[9] 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 13134–13143, 2020. 1, 2, 4, 5, 6

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016. 3

[11] Yicong Hong, Cristian Rodriguez Opazo, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. In *NeuIPS*, 2020. 2, 6

[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, 5, 6, 7

[13] Haoshuo Huang, Vihan Jain, Harsh Mehta, Alexander Ku, Gabriel Magalhães, Jason Baldridge, and Eugene Ie. Transferable representation learning in vision-and-language navigation. In *ICCV*, pages 7403–7412, 2019. 1, 3

[14] Gabriel Ilharco, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. In *NeuIPS*, 2019. 7

[15] Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *TACL*, 5:339–351, 2017. 3

[16] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha S. Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *CVPR*, pages 6741–6749, 2019. 6, 7

[17] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, volume 12373 of *Lecture Notes in Computer Science*, pages 104–120. Springer, 2020. 1

[18] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *EMNLP*, pages 4392–4412, 2020. 1, 2, 5, 7

[19] Kuang-Huei Lee, Xi Chen, Gang Hua, Houdong Hu, and Xiaodong He. Stacked cross attention for image-text matching. In *ECCV*, pages 212–228, 2018. 1, 2

[20] Jialu Li, Hao Tan, and Mohit Bansal. Improving cross-modal alignment in vision language navigation via syntactic information. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *NAACL*, pages 1041–1050, 2021. 7

[21] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Çelikyilmaz, Jianfeng Gao, Noah A. Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. In *EMNLP*, pages 1494–1499, 2019. 6

[22] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, pages 121–137, 2020. 2

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*. OpenReview.net, 2019. 5

[24] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, pages 13–23, 2019. 2

[25] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *ICLR*, 2019. 2, 7

[26] Chih-Yao Ma, Zuxuan Wu, Ghassan AlRegib, Caiming Xiong, and Zsolt Kira. The regretful agent: Heuristic-aided navigation through progress estimation. In *CVPR*, pages 6732–6740, 2019. 6

[27] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *ECCV*, pages 259–274, 2020. 1, 2, 5

[28] Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *CVPR*, pages 12527–12537. Computer Vision Foundation / IEEE, 2019. 1

[29] Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *EMNLP*, pages 684–695. Association for Computational Linguistics, 2019. 1

[30] Yuankai Qi, Zizheng Pan, Yicong Hong, Ming-Hsuan Yang, Anton van den Hengel, and Qi Wu. The road to know-where: An object-and-room informed sequential bert for indoor vision-language navigation. In *ICCV*, pages 1655–1664, 2021. 2, 6, 7

[31] Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. Object-and-action aware model for visual language navigation. In *ECCV*, pages 303–317, 2020. 6

[32] 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 9979–9988, 2020. 1, 2, 5, 7

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 3

[34] Hao Tan and Mohit Bansal. LXMERT: learning cross-modality encoder representations from transformers. In *EMNLP*, pages 5099–5110, 2019. 2, 3

[35] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NAACL-HLT*, pages 2610–2621, 2019. 2, 4, 6, 7

[36] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *CoRL*, pages 394–406, 2019. 1, 2, 5, 6

[37] Hanqing Wang, Wenguan Wang, Tianmin Shu, Wei Liang, and Jianbing Shen. Active visual information gathering for vision-language navigation. In *ECCV*, 2020. 6

[38] Xin Wang, Qiuyuan Huang, Asli Çelikyilmaz, 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. 1, 2, 6, 7

[39] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV*, pages 69–85, 2016. 1, 2

[40] 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. 1

[41] Yi Zhu, Fengda Zhu, Zhaohuan Zhan, Bingqian Lin, Jianbin Jiao, Xiaojun Chang, and Xiaodan Liang. Vision-dialog navigation by exploring cross-modal memory. In *CVPR*, pages 10727–10736, 2020. 6