VLN↻BERT: A Recurrent Vision-and-Language BERT for Navigation

Yicong Hong[1] Qi Wu[2] Yuankai Qi[2] Cristian Rodriguez-Opazo[1,2] Stephen Gould[1]
[1]The Australian National University [2]The University of Adelaide
Australian Centre for Robotic Vision
{yicong.hong, cristian.rodriguez, stephen.gould}@anu.edu.au
qi.wu01@adelaide.edu.au, qykshr@gmail.com

Project URL: https://github.com/YicongHong/Recurrent-VLN-BERT

Abstract

Accuracy of many visiolinguistic tasks has benefited significantly from the application of vision-and-language (V&L) BERT. However, its application for the task of vision-and-language navigation (VLN) remains limited. One reason for this is the difficulty adapting the BERT architecture to the partially observable Markov decision process present in VLN, requiring history-dependent attention and decision making. In this paper we propose a recurrent BERT model that is time-aware for use in VLN. Specifically, we equip the BERT model with a recurrent function that maintains cross-modal state information for the agent. Through extensive experiments on R2R and REVERIE we demonstrate that our model can replace more complex encoder-decoder models to achieve state-of-the-art results. Moreover, our approach can be generalised to other transformer-based architectures, supports pre-training, and is capable of solving navigation and referring expression tasks simultaneously.

1. Introduction

Asking a robot to navigate in complex environments following human instructions has been a long-term goal in AI research. Recently, a great variety of vision-and-language navigation (VLN) setups [3, 44, 52] have been introduced for relevant studies and a large number of works explore different methods to leverage visual and language clues to assist navigation. For example, in the popular R2R navigation task [3], enhancing the learning of visual-textual correspondence is essential for the agent to correctly interpret the instruction and perceive the environment.

On the other hand, recent work on vision-and-language pre-training has achieved significant improvement over a wide range of visiolinguistic problems. Instead of designing complex and monolithic models for different tasks, those methods pre-train a multi-layer Transformer [53] on a large number of image-text pairs to learn generic cross-modal representations [7, 26, 28, 30, 33, 48, 50], known as V&L BERT (Bidirectional Encoder Representations from Transformers [10]). Such advances have inspired us to employ V&L BERT for VLN, replacing the complicated modules for modelling cross-modal relationships and allowing the learning of navigation to adequately benefit from the pre-trained visual-textual knowledge. Unlike recent works on VLN, which apply a pre-trained V&L BERT only for encoding language [14, 29] or for measuring the instruction-path compatibility [37], we propose to use existing V&L BERT models themselves for learning to navigate.

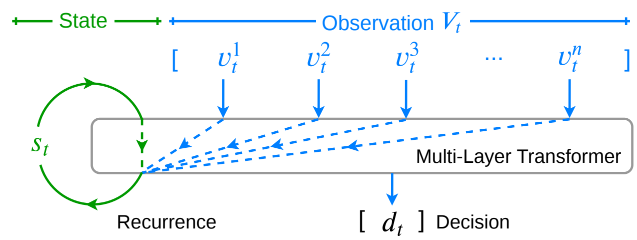

Figure 1. Recurrent multi-layer Transformer for addressing partially observable inputs. A state token is defined along with the input sequence. At each time step, a new state representation $\boldsymbol{s}_t$ will be generated based on the new observation. Meanwhile, the past information will help inferring a new decision d_t.

However, an essential difference between VLN and other vision-and-language tasks is that VLN can be considered as a partially observable Markov decision process, in which future observations are dependent on the agent's current state and action. Meanwhile, at each navigational step, the visual observation only corresponds to partial instruction, requiring the agent to keep track of the navigation progress and correctly localise the relevant sub-instruction to gain useful information for decision making. Another difficulty of applying V&L BERT for VLN is the high demand on computational power; since the navigational episode could be very long, performing self-attention on a long visual and textual sequence at each time step will cost an excessive amount of (GPU) memory during training.

To address the aforementioned problems, we propose a recurrent vision-and-language BERT for navigation, or simply VLN⟳BERT. Instead of employing large-scale datasets for pre-training which usually require thousands of GPU hours, the aim of this work is to allow the learning of VLN to adequately benefit from pre-trained V&L BERT. Based on the previously proposed V&L BERT models, we implement a recurrent function in their original architecture (Fig. 1) to model and leverage the history-dependent state representations, without explicitly defining a memory buffer [61] or applying any external recurrent modules such as an LSTM [16]. To reduce the memory consumption, we control the self-attention to consider the language tokens as keys and values but not queries during navigation, which is similar to the cross-modality encoder in LXMERT [50]. Such design greatly reduces the memory usage so that the entire model can be trained on a single GPU without performance degeneration. Furthermore, as in the original V&L BERT, our proposed model has the potential of multi-task learning, it is able to address other vision and language problems along with the navigation task.

We employ two datasets to evaluate the performance of our VLN⟳BERT, R2R [3] and REVERIE [44]. The chosen datasets are different in terms of the provided visual clues, the instructions and the goal. Our agent, initialised from a pre-trained V&L BERT and fine-tuned on the two datasets, achieves state-of-the-art results. We also initialise our model with the PREVALENT [14], a LXMERT-like model pre-trained for VLN. On the test split of R2R [3], it improves the Success Rate absolutely by 8% and achieves 57% Success weighted by Path Length (SPL). For the remote referring expression task in REVERIE [44], our agent obtains 23.99% navigation SPL and 13.51% Remote Grounding SPL. These results indicate the strong generalisation ability of our proposed VLN⟳BERT as well as the potential of using it for merging the learning of VLN with other vision and language tasks.

2. Related Work

Vision-and-Language Navigation Learning navigation with visual-linguistic clues has drawn significant research interests. The recent R2R [3] and Touchdown [6] datasets introduce human natural language as guidance and apply photo-realistic environments for navigation. Following these work, dialog-based navigation such as CVDN [52], VNLA [41] and HANNA [40], navigation for localising a remote object such as REVERIE [44], VLN in continuous environment [22], and multilingual navigation with spatial-temporal grounding such as RxR [24] have been proposed for further research.

One crucial challenge in VLN is to understand the visual-textual correspondence for decision making. To achieve this, Self-Monitoring [36] and RCM [55] adopt cross-modal attention to highlight the relevant observations and instruction at each step. Speaker-Follower [11] and EnvDrop [51] learn on the augmented training data via a self-supervised manner. FAST [19] resorts self-correction navigation, while APS [12] samples adversarial paths for training to enhance the model's ability to generalise. AuxRN [60] applies several auxiliary losses to learn comprehensive representations, Qi *et al*. [43] and Wang *et al*. [54] also design loss functions to encourage the agent to follow the instructions to take the shortest paths. More recently, Hong *et al*. [17] propose a graph network to model the intra- and inter-modal relationships among the contextual and visual clues. The great improvements achieved by these methods encourage researchers to explore simpler and more powerful visiolinguistic learning network for VLN.

Visual BERT Pre-Training Following the success of pre-trained BERT on a wide range of natural language processing tasks [10], the model has been extended to process visual tokens and to pre-train on large-scale image/video-text pairs for learning generic visual-linguistic representations. Previous research introduce two-stream BERT models which encode texts and images separately, and fuse the two modalities in a later stage [33, 50], as well as one-stream BERT models which directly perform inter-modal grounding [7, 26, 28, 30, 48]. Although video BERT approaches have been proposed to learn the correspondence between texts and video frames [27, 35, 49, 57], we are the first to integrate recurrence into BERT to learn partially-observable and temporal-dependent inputs. In terms of VLN pre-training, PRESS fine tunes a pre-trained language BERT to encode instructions [29], PREVALENT trains a V&L BERT on a large amount of image-text-action triplets from scratch to learn navigation-oriented textual representations [14], and VLN-BERT [37] fine-tunes a ViLBERT [33] on instruction-trajectory pairs to measure their compatibility in beam search setting. Unlike all previous work, our VLN⟳BERT can augment various V&L BERT models with recurrent function, it is a navigator network by itself that can be directly trained for navigation.

V&L Multi-Task Learning Instead of building a monolithic model for different V&L tasks, numbers of previous work explore multi-task learning with a unified model for utilising the common and the complementary knowledge to reduce the domain gap [31, 39, 42, 47, 33]. Very recently, 12-in-1 [34] trains a single ViLBERT [33] on 12 different datasets across four categories of V&L tasks, including visual question answering, referring expressions, multi-modal verification and caption-based image retrieval. In Vision-and-Language Navigation, Wang *et al*. [56] propose a multi-task navigation model to address the R2R navigation [3] and

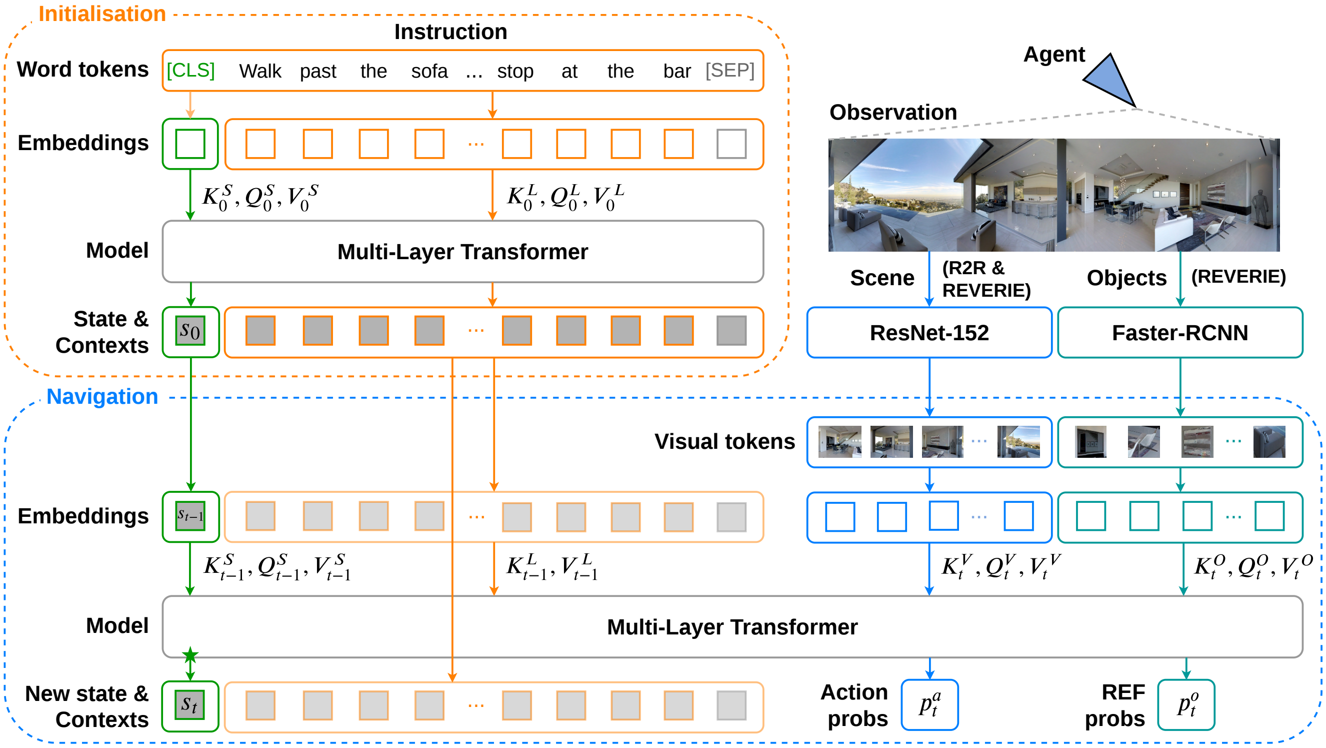

Figure 2. Schematics of the Recurrent Vision-and-Language BERT. At the initialisation stage, the entire instruction is encoded by a multi-layer Transformer, where the output feature of the `[CLS]` token serves as the initial state representation of the agent. During navigation, the concatenated sequence of state, encoded language and new visual observation is fed to the same Transformer to obtain the updated state and decision probabilities. The updated state and the language encoding from initialisation will be fused and applied as input at the next time step. The green star (★) indicates the *cross-modal matching* (Eq. 12) and the *past decision encoding* (Eq. 13) in *State Refinement*.

the Navigation from Dialog History (NDH) [52] problems. Comparing to previous methods, our VLN⟳BERT uses a single network to address the navigation task and the remote referring expression task seamlessly in REVERIE [44].

3. Proposed Model

In this section, we first define the vision-and-language navigation task, then we revisit the BERT model [10] and present the architecture of our proposed VLN⟳BERT .

3.1. VLN Background

The problem of VLN can be formulated as follows: Given a natural language instruction $\boldsymbol{U}$ which contains a sequence of words, at each time step t, the agent observes the environment and infers an action a_t that transfers the agent from state $\boldsymbol{s}_t$ to a new state $\boldsymbol{s}_{t+1}$. The state consists of the navigational history and the current spatial position defined by a triplet $\langle \boldsymbol{C}_t, \theta_t, \phi_t \rangle$, where $\boldsymbol{C}_t$ is a viewpoint on the pre-defined connectivity graph of the environment [3], and θ_t and ϕ_t are the angles of heading and elevation, respectively. The agent needs to execute a sequence of actions to navigate on the connectivity graph and eventually decides to `stop` at the target position to complete the task.

3.2. Revisit BERT

Bidirectional Encoder Representations from Transformers (BERT) [10] is a multi-layer Transformer architecture [53] designed to pre-train deep bidirectional language representations. Each layer of the Transformer encodes the language features from the previous layer $\boldsymbol{X}_{l-1} \in \mathbb{R}^{I \times hd_h}$ with multi-head self-attention to capture the dependencies among the I words in the sentence, and applies a residual feed-forward network to process the output features.

Formally, the k-th attention head at the l-th layer performs self-attention over $\boldsymbol{X}_{l-1}$ as

$$\boldsymbol{Q} = \boldsymbol{X}_{l-1}\boldsymbol{W}_{l,k}^{Q}, \boldsymbol{K} = \boldsymbol{X}_{l-1}\boldsymbol{W}_{l,k}^{K}, \boldsymbol{V} = \boldsymbol{X}_{l-1}\boldsymbol{W}_{l,k}^{V} \quad (1)$$

$$\boldsymbol{H}_{l,k} = \text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\top}}{\sqrt{d_h}}\right)\boldsymbol{V} \quad (2)$$

where $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$ and $\boldsymbol{W}^V \in \mathbb{R}^{hd_h \times d_h}$ are learnable linear projections[1] specifically for queries, keys and values; d_h is the hidden dimension of the network. The outputs from all the attention heads will be concatenated and projected onto the same dimension as the input as

$$\boldsymbol{H}_l = [\boldsymbol{H}_{l,1}; ...; \boldsymbol{H}_{l,h}]\, \boldsymbol{W}_l^{O} \quad (3)$$

[1] All $\boldsymbol{W}$ in this section denotes learnable linear projections

where h is the total number of heads, $[;]$ denotes concatenation and $\boldsymbol{W}^O \in \mathbb{R}^{hd_h \times hd_h}$ is a learned linear projection. Finally, the output of layer l is formulated by

$$\boldsymbol{H}'_l = \text{LayerNorm}(\boldsymbol{H}_l + \boldsymbol{X}_{l-1}) \tag{4}$$

$$\boldsymbol{X}'_l = \text{ReLU}(\boldsymbol{H}'_l \boldsymbol{W}_l^{F_1})\boldsymbol{W}_l^{F_2} \tag{5}$$

$$\boldsymbol{X}_l = \text{LayerNorm}(\boldsymbol{H}'_l + \boldsymbol{X}'_l) \tag{6}$$

where ReLU is the Rectified Linear Unit activation function and LayerNorm is layer normalisation [4].

Based on this architecture, BERT has been extended to V&L BERT [7, 26, 28, 30, 33, 48, 50], which takes the concatenation of language tokens and visual tokens as input, and pre-trains on image-text corpus to learn generic visiolinguistic representations.

3.3. Recurrent VLN BERT

The idea of our VLN↻BERT can be adapted to a wide range of Transformer-based networks. In this section we apply the recently proposed one-stream V&L BERT model OSCAR [30] for demonstration. We modify the model to enable the learning of navigation and the associated referring expression (REF) task. As shown in Fig. 2, at each time step, the network takes four sets of tokens as input; the previous state token $\boldsymbol{s}_{t-1}$, the language tokens $\boldsymbol{X}$, the visual tokens $\boldsymbol{V}_t$ for scene, and the visual tokens $\boldsymbol{O}_t$ for objects (only in REVERIE [44]). Then, it performs self-attention over these cross-modal tokens to capture the textual-visual correspondence for inferring the action probabilities $\boldsymbol{p}_t^a$ and the object grounding probabilities $\boldsymbol{p}_t^o$ (only in REVERIE [44]):

$$\boldsymbol{s}_t, \boldsymbol{p}_t^a, \boldsymbol{p}_t^o = \text{VLN↻BERT}(\boldsymbol{s}_{t-1}, \boldsymbol{X}, \boldsymbol{V}_t, \boldsymbol{O}_t) \tag{7}$$

Language Processing At initialisation (t=0), a sequence of words consisting of the classification token `[CLS]`, the language tokens of the instruction $\boldsymbol{U}$ and the separation token `[SEP]` will be fed into VLN↻BERT, where `[CLS]` and `[SEP]` are pre-defined in BERT models. In the pre-training of OSCAR [30], the `[CLS]` token is applied for aggregating relevant visiolinguistic clues from the input sequence for contrastive learning. Here, we defined the embedded `[CLS]` token as the initial state representation $\boldsymbol{s}_0$, to inherit such function — initialise an agent's state which is aware of the entire navigation task.

$$\boldsymbol{s}_0, \boldsymbol{X} = \text{VLN↻BERT}(\texttt{[CLS]}, \boldsymbol{U}, \texttt{[SEP]}) \tag{8}$$

During navigation steps (t>0), unlike the state token $\boldsymbol{s}_t$ or the visual tokens $\boldsymbol{V}_t$ and $\boldsymbol{O}_t$ which performs self-attention with respect to the entire input sequence, the language tokens $\boldsymbol{X}$ only serve as the keys and values in the Transformer. We consider the language tokens produced by the model at the initialisation step as a deep representation of the instruction which does not need to be further encoded in later steps. Not updating the language features also save a huge amount of computational resources since the instruction and the trajectory can be long in VLN problems.

Vision Processing At each navigation step (t>0), the agent makes new visual observation in the environment and uses the visual clues to assist navigation. To process the visual clues, the network first projects the image features of views at the navigable directions $\boldsymbol{I}_t^v$ to the same space as the BERT token as $\boldsymbol{V}_t = \boldsymbol{I}_t^v \boldsymbol{W}^{I^v}$. Then, the visual tokens will be concatenated with the state token and the language tokens, and fed into the model.

In terms of the remote REF task [44], we simply consider the object features $\boldsymbol{I}_t^o$ as additional visual tokens in the input sequence. Similarly, the features will be projected onto the token space as $\boldsymbol{O}_t = \boldsymbol{I}_t^o \boldsymbol{W}^{I^o}$ and fed into the model. The object clues can provide valuable information about the important landmarks on the path, which could be very helpful to the navigation with high-level instructions [44].

State Representation We formulate the agent's state at each time step $\boldsymbol{s}_t$ as the summary of all textual and visual clues that the agent collects, as well as all decisions that the agent makes until the current viewpoint. Instead of explicitly defining a memory buffer [61] or implementing an additional recurrent network [16] to store the past experiences, our model relies on BERT's original architecture to recognise time-dependent inputs, and recurrently updates $\boldsymbol{s}_0$ from initialisation to represent the state. At each navigation step, the state representation is used as the leading input token of the entire textual-visual sequence. It then performs inter-modal self-attention in VLN↻BERT with other tokens to update its content and becomes the leading token of the input at the next step, in an autoregressive way.

State Refinement Unlike most of the V&L BERT models which apply the output feature of the `[CLS]` token for classification, our state is not directly used for inferring a decision (see following *Decision Making* subsection), which means, the vanilla state representation is not explicitly enforced to capture the most important language and visual features. To address this issue, our model matches the raw textual and visual tokens, and feeds the output to the state representation. Formally, let $\boldsymbol{Q}_{l,k}^s$ and $\boldsymbol{K}_{l,k}^x$ be the state and textual tokens at head k of the final (l=12) layer of VLN↻BERT, the attention scores over the textual tokens can be expressed as:

$$\boldsymbol{A}_{l,k}^{s,x} = \frac{\boldsymbol{Q}_{l,k}^s \boldsymbol{K}_{l,k}^{x\,\top}}{\sqrt{d_h}} \tag{9}$$

Then, we average the scores over all the attention heads (K=12) and apply a Softmax function to get the overall state-language attention weights as:

$$\tilde{\boldsymbol{A}}_l^{s,x} = \text{Softmax}(\bar{\boldsymbol{A}}_l^{s,x}) = \text{Softmax}\left(\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{A}_{l,k}^{s,x}\right) \tag{10}$$

Similarly, the visual attention scores $\boldsymbol{A}_{l,k}^{s,v}$ and weights $\widetilde{\boldsymbol{A}}_l^{s,v}$ can be obtained. Now, we perform a weighted sum over the input textual tokens and visual tokens respectively to obtain the weighted raw features as:

$$\boldsymbol{F}_t^x = \widetilde{\boldsymbol{A}}_l^{s,x} \boldsymbol{X} \quad \text{and} \quad \boldsymbol{F}_t^v = \widetilde{\boldsymbol{A}}_l^{s,v} \boldsymbol{V}_t \tag{11}$$

We then enforce a **cross-modal matching** between the raw textual and visual features via element-wise product and send such information to the agent's state as:

$$\boldsymbol{s}_t^f = [\boldsymbol{s}_t^r; \boldsymbol{F}_t^x \odot \boldsymbol{F}_t^v] \boldsymbol{W}^r \tag{12}$$

where $\boldsymbol{s}_t^r$ is the output state features at the final layer. Notice that in REVERIE [44], only the visual features is sent to the state representation, i.e., Eq. 12 becomes $\boldsymbol{s}_t^f = [\boldsymbol{s}_t^r; \boldsymbol{F}_t^v] \boldsymbol{W}^r$. This is because the navigational instructions in REVERIE are high-level, hence performing step-wise matching between the raw textual and visual features is less valuable.

Finally, past decisions are important for the agent to keep track of the navigation progress, our network records the new decision by feeding the directional features of the selected action $\boldsymbol{a}_t$ into the state token as:

$$\boldsymbol{s}_t = \left[\boldsymbol{s}_t^f; \boldsymbol{a}_t\right] \boldsymbol{W}^s \tag{13}$$

where $\boldsymbol{s}_t$ is the new representation of the agent's state at time step t.

Decision Making Many previous VLN agents apply an inner product between the state representation and the visual features at candidate directions to evaluate the state-vision correspondence, and choose a direction with the highest matching score to navigate [36, 51]. We find that the BERT network can nicely perform such scoring because it is fully built upon the inner product based soft-attention. Inspired by the method of predicting alignment between regions and phrases in VisualBERT [28], we directly apply the mean attention weights of the visual tokens over all the attention heads in the last layer, with respect to the state, as the action probabilities, simply $\boldsymbol{p}_t^a = \widetilde{\boldsymbol{A}}_l^{s,v}$ (as defined in Eq. 10). As for the remote referring expression task [44], our agent uses the same method to select an object. The selection probabilities can be expressed as $\boldsymbol{p}_t^o = \widetilde{\boldsymbol{A}}_l^{s,o}$, where $\widetilde{\boldsymbol{A}}_l^{s,o}$ is the mean attention weights for all candidate objects. We refer the *Appendix* §B.2 for more details.

3.4. Training

We train our network with a mixture of reinforcement learning (RL) and imitation learning (IL) objectives. We apply A2C [38] for RL, in which the agent samples an action according to $\boldsymbol{p}_t^a$ and measures the advantage $\boldsymbol{A}_t$ at each step (we refer the *Appendix* §B.5 for more details about RL.). In IL, our agent navigates on the ground-truth trajectory by following teacher actions and calculates a cross-entropy loss for each decision. Formally, we minimise the navigation loss function, expressed for each given sample, as

$$\mathcal{L} = -\sum_t a_t^s \log(p_t^a) A_t - \lambda \sum_t a_t^* \log(p_t^a) \tag{14}$$

where a_t^s is the sampled action and a_t^* is the teacher action. Here λ is a coefficient for weighting the IL loss. In REVERIE [44], we applied an additional cross-entropy term $\sum_t o_t^* \log(p_t^o)$ to learn object grounding.

3.5. Adaptation

We initialise the parameters of VLN↻BERT from OSCAR [30] pre-trained without object tags. Although OSCAR is trained on regional features, we find that it is also compatible with the grid features of the entire scene. When adapting to the LXMERT-like [50] model in PREVALENT [14], we remove the language branch in the cross-modality encoder and concatenate the state token with the visual tokens for self-attention (see *Appendix* §B.3 for schematics). We also remove the entire downstream network EnvDrop [51], including the Speaker and the environmental dropout, and then directly fine-tune the model pre-trained by PREVALENT for navigation.

4. Experiments

Implementation Details All experiments are conducted on a single NVIDIA 2080Ti GPU, the learning rate is fixed to 10^{-5} throughout the training and AdamW optimiser [32] is applied. For R2R, we train the agent directly on the mixture of the original training data and the augmented data from PREVALENT [14], the batch size[2] is set to 16 and the network is trained for 300,000 iterations. For REVERIE, we use batch size[2] 8 and train the agent for 200,000 iterations. Images in the environments are encoded by a ResNet-152 [15] pre-trained on Places365 [59], and objects are encoded by a Faster-RCNN [46] pre-trained on the Visual Genome [23]. Early stopping is applied when the training saturates, the model which achieves the highest SPL in validation unseen split is adopted for testing.

Evaluation Metrics We apply the standard metrics employed by previous works to evaluate the performance.

R2R [3] considers Trajectory Length (TL): the average path length in meters, Navigation Error (NE): the average distance between agent's final position and the target in meters, Success Rate (SR): the ratio of stopping within 3 meters to the target, and Success weighted by the normalised inverse of the Path Length (SPL) [2].

[2]Half for RL and half for IL in each iteration, corresponding to the first and the second term in Eq. 14, respectively.

Methods	R2R Validation Seen				R2R Validation Unseen				R2R Test Unseen			
	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
Random	9.58	9.45	16	-	9.77	9.23	16	-	9.89	9.79	13	12
Human	-	-	-	-	-	-	-	-	11.85	1.61	86	76
Seq2Seq-SF [3]	11.33	6.01	39	-	8.39	7.81	22	-	8.13	7.85	20	18
Speaker-Follower [11]	-	3.36	66	-	-	6.62	35	-	14.82	6.62	35	28
SMNA [36]	-	3.22	67	58	-	5.52	45	32	18.04	5.67	48	35
RCM+SIL (train) [55]	10.65	3.53	67	-	11.46	6.09	43	-	11.97	6.12	43	38
PRESS [29] †	10.57	4.39	58	55	10.36	5.28	49	45	10.77	5.49	49	45
FAST-Short [19]	-	-	-	-	21.17	4.97	56	43	22.08	5.14	54	41
EnvDrop [51]	11.00	3.99	62	59	10.70	5.22	52	48	11.66	5.23	51	47
AuxRN [60]	-	3.33	70	**67**	-	5.28	55	50	-	5.15	55	51
PREVALENT [14] †	10.32	3.67	69	65	10.19	4.71	58	**53**	10.51	5.30	54	51
RelGraph [17]	10.13	3.47	67	65	9.99	4.73	57	**53**	10.29	4.75	55	52
Ours (no init. OSCAR)	9.78	3.92	62	59	10.31	5.10	50	46	11.15	5.45	51	47
Ours (init. OSCAR)	10.79	**3.11**	**71**	**67**	11.86	**4.29**	**59**	**53**	12.34	**4.59**	**57**	**53**
Ours (init. PREVALENT)	11.13	**2.90**	**72**	**68**	12.01	**3.93**	**63**	**57**	12.35	**4.09**	**63**	**57**

Table 1. Comparison of agent performance on R2R in single-run setting. † work that applies pre-trained BERT for language encoding.

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		
Random	2.74	8.92	1.91	11.99	1.97	1.31	1.76	11.93	1.01	10.76	0.96	0.56	2.30	8.88	1.44	10.34	1.18	0.78
Human	–	–	–	–	–	–	–	–	–	–	–	–	81.51	86.83	53.66	21.18	77.84	51.44
Seq2Seq-SF [3]	29.59	35.70	24.01	12.88	18.97	14.96	4.20	8.07	2.84	11.07	2.16	1.63	3.99	6.88	3.09	10.89	2.00	1.58
RCM [55]	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 [36]	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 [19]	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 [44]	**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
Ours (no init. OSCAR)	39.56	42.09	36.29	12.06	21.15	19.39	**25.76**	**29.28**	**22.16**	14.52	11.62	9.87	18.52	20.18	15.47	14.09	8.80	7.29
Ours (init. OSCAR)	39.85	41.32	35.86	12.85	24.46	22.28	25.53	27.66	21.06	14.35	**14.20**	**12.00**	**24.62**	26.67	**19.48**	14.88	**12.65**	**10.00**
Ours (init. PREVALENT)	**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**

Table 2. Comparison of agent performance of navigation and remote referring expression on REVERIE.

REVERIE [44] defines Success Rate (SR) as the ratio of stopping at a viewpoint where the target object is visible (in panorama), and considers the corresponding SPL. It also employ Oracle Success Rate (OSR): the ratio of having a viewpoint along the trajectory where the target object is visible, Remote Grounding Success Rate (RGS): the ratio of grounding to the correct objects when stopped, and RGSPL, which weights RGS by the trajectory length.

4.1. Main Results

Comparison with SoTA Results in Table 1 compare the single-run (greedy search, no pre-exploration [55]) performance of different agents on the R2R benchmark. Our proposed VLN↻BERT initialised from OSCAR [30] (init. OSCAR) performs better than previous methods across all the dataset splits. Comparing to a randomly initialised network (no init. OSCAR), the large performance degeneration suggests that the pre-trained general vision-linguistic knowledge significantly benefits the learning of navigation. The model initialised from PREVALENT [14], pre-trained especially for VLN, further improves the agent's performance, achieving 63% SR (+8%) and 57% SPL (+5%) on the test unseen split[3]. Comparing to PRESS [29] and PREVALENT [14] which only fine-tune a pre-trained BERT for extracting language features, adding recurrence into V&L BERT and using the model directly as the navigator network allows the VLN learning to adequately benefit from the pre-trained knowledge. Such performance gain cannot be achieved by using pre-trained V&L BERT only as a feature extractor, as will be shown in §4.2 *Ablation Study*. Moreover, the large gain in SR with a slight increase in TL suggests that the agent is able to navigate both accurately and efficiently. Comparing to previous methods, we can see that the performance gap between the validation unseen and the test unseen splits is greatly reduced, which means our agent has a stronger generalisation ability to novel instructions and environments.

In REVERIE [44] (Table 2), our VLN↻BERT (init. OSCAR) generalises much better to unseen data. On the validation unseen split, the SR of navigation and object grounding has been absolutely improved by 11.13% and 6.36% respectively. On the test unseen split[3], our method ob-

[3]R2R Leaderboard: https://evalai.cloudcv.org/web/challenges/challenge-page/97/overview, REVERIE Leaderboard: https://eval.ai/web/challenges/challenge-page/606/overview

tains 24.62% SR and 19.48% SPL for navigation, as well as 12.65% RGS and 10.00% RGSPL for REF, achieving a better performance than the previous best [44] which applies SoTA navigator FAST [19] for navigation and pointer MATTN [58] for object grounding. Compare our model with OSCAR initialisation to without, the navigation results on the validation splits are similar, but the navigation on the test split and the object grounding across all the data splits are largely improved. This result also suggests that it is possible to apply a BERT-based model for VLN and REF multi-task learning. Although the previous method has higher OSR, it is likely due to longer searching (long TL), the lower SR suggests that the agent does not know where to stop correctly. In Table 2, we also present the performance of VLN↻BERT initialised from PREVALENT [14], which achieves the best result across almost all of the metrics in all dataset splits. It is very interesting to see that although PREVALENT is pre-trained on low-level R2R instructions [3] without other V&L knowledge, it significantly boosts the navigation with high-level instructions as well as the object grounding in REVERIE. We hypothesis that the pre-trained knowledge provides the model with some structural priors, while the learning of REF is strongly influenced by navigation, especially at the early training stage where the target object is rarely observable by the agent.

Visualisation of Language Attention To demonstrate that our VLN↻BERT (init. OSCAR) is able to trace the navigation progress, we visualise the changes of language attention weights at the final Transformer layer over all instructions during navigation (Fig. 3). As the agent moves forward, the attention weights with respect to state shifts from the beginning of the instructions to the end. Since the sub-instructions and sub-paths for each sample in R2R is monotonically aligned [18], our results indicate that the state nicely records the partial instruction that has been completed. In terms of the attention weights with respect to the visual token at the select direction, it follows a similar pattern meaning that the most relevant part of the instruction is used for guiding the action selection.

4.2. Ablation Study

Network Components Table 3 shows comprehensive ablation experiments on the influence of using V&L BERT (init. OSCAR) to replace or to add the key network components in the baseline model (EnvDrop [51]). The baseline model consists of a language encoder, a visual encoder, a state LSTM and a decision making module, corresponding to the columns of *Language*, *Vision*, *State* and *Decision* in Table 3, respectively. E.g., Model #3 replaces the language encoder and visual encoder in baseline with a V&L BERT. For fair comparison, all models in the table are trained with

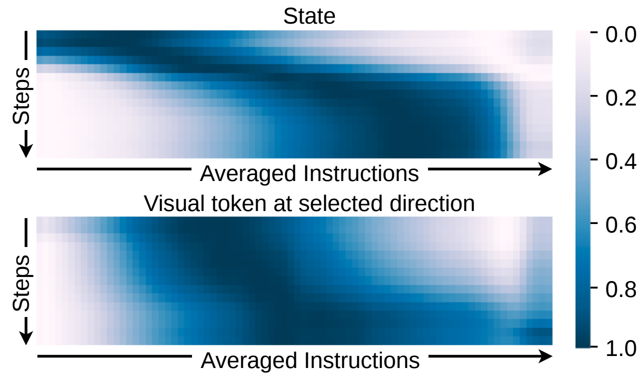

Figure 3. Averaged attention weights over all instructions in validation unseen split during navigation. *State*: Attention weights with respect to the state representation. *Selected Action*: Attention weights with respect to the visual token at the selected direction.

the same data and training strategy as our VLN↻BERT. As the results suggested, our proposed method is a multi-functional framework, the more network components it covers, the larger performance gain it achieves. Comparing the baseline with Model #1 and #2, we can see that employing a pre-trained BERT as language encoder improves the performance only if the BERT is fine-tuned for navigation. This finding is also supported by using the V&L BERT to encode both the textual and visual signals (Model #3 and #4). However, simply using pre-trained V&L BERT as text and image encoders does not fully utilise its power; model #5 indicates that relying on the original architecture of BERT to learn recurrence is feasible and it is able to achieve better results. Moreover, using the averaged visual attention weights of the final layer of the Transformer as the action probabilities (Model #6) and enhancing the state representation with visual-textual matching as defined in Eq. 12 (full model) further improves the agent's performance.

Self-Attended Language Features Due to the long instructions and episodes, high memory cost during training is one of the key issues that prevents previous research to apply BERT for self-attention at every time step. To demonstrate the influence of self-attending textual features during navigation, we compare the agent's performance and the training time GPU memory consumption (constrained to a single 11GB memory GPU) of re-attending the language at each step. As shown in Table 4, training for *Emb-Attn*, *Init-Attn* and *Re-Attn* consume much more memory for each sample than performing language self-attention only at initialisation (*Ours*), and their performances are worse than *Ours*. The results of *Re-Attn* degenerates significantly because at each time step the output language features aggregate the most relevant visual-textual clues at a certain viewpoint, which suppresses the valuable information in other part of the instruction for the future steps. We refer *Appendix* §C.2 for experiment on language self-attention with larger batch size by applying gradient accumulation.

Models	V&L BERT (init. OSCAR)						R2R Validation Seen				R2R Validation Unseen			
	Language	Vision	State	Decision	Matching	Train	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
Baseline [51]							11.84	4.44	57.79	52.85	12.50	5.26	49.81	43.45
1	✓						10.81	4.98	49.95	46.19	11.34	5.68	44.15	39.64
2	✓					✓	11.73	4.18	59.26	54.12	12.59	5.00	52.11	45.75
3	✓	✓					9.26	6.85	34.77	33.33	8.92	7.43	30.74	29.05
4	✓	✓				✓	11.37	3.50	67.97	63.94	12.98	4.73	54.75	48.31
5	✓	✓	✓			✓	11.10	3.81	65.52	61.24	12.20	4.62	55.21	49.72
6	✓	✓	✓	✓		✓	10.70	3.21	70.32	66.45	11.46	4.48	57.22	52.57
Full model	✓	✓	✓	✓	✓	✓	10.79	**3.11**	**71.11**	**67.23**	11.86	**4.29**	**58.71**	**53.41**

Table 3. Ablation experiments on the effect of applying V&L BERT for learning navigation. Checkmarks indicate using V&L BERT to replace or to add the corresponding network component in the baseline model. *Matching* indicates the *cross-modal matching* (Eq. 12), and *Train* with checkmark means the V&L BERT is fine-tuned for navigation.

Models	R2R Validation Seen				R2R Validation Unseen				Batch Size	Memory
	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑		
Emb-Attn	10.84	3.40	67.19	63.19	11.31	4.64	55.60	51.00	6	11.0GB
Init-Attn	10.21	3.93	61.80	58.06	10.39	4.59	53.47	49.05	6	11.0GB
Re-Attn	8.99	5.81	40.84	39.44	8.85	6.22	37.76	35.78	6	11.0GB
Ours	10.79	**3.11**	**71.11**	**67.23**	11.86	**4.29**	**58.71**	**53.41**	**16**	**9.2GB**

Table 4. Comparison of performing language self-attention: On the raw word embeddings at each step (*Emb-Attn*), on the initialised language features at each step (*Init-Attn*), on the output language features from previous step (*Re-Attn*), or only at initialisation (*Ours*). *Memory* is the training time GPU memory cost.

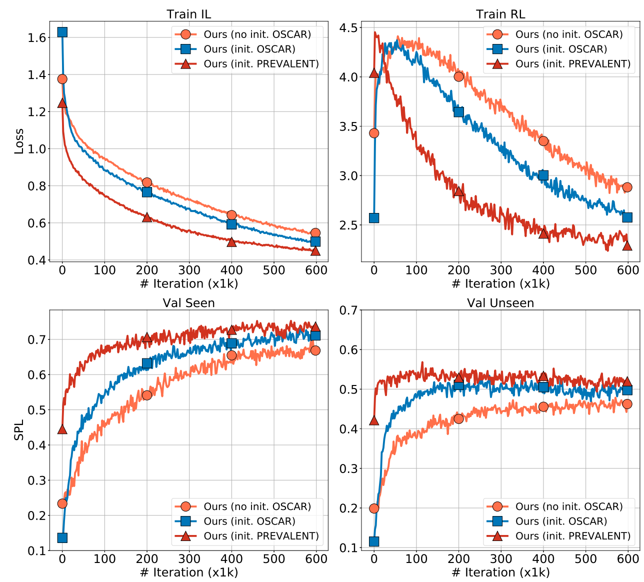

Figure 4. Comparison of the learning curves. *no init.* means randomly initialised network parameters.

Learning Curves As shown in Fig. 4, we compare the learning curves of VLN↻BERT initialised from different models. The training losses of our method initialised from pre-trained OSCAR [30] converges faster than a randomly initialised model, and it reaches much higher SPL in both validation seen and unseen environments. Moreover, our model initialised from PREVALENT [14] learns significantly faster than the other two methods and it is able to achieve a much better performance within much fewer iterations. In terms of training in real time, the model init. OSCAR takes about 7 days[4] to complete 600,000 iterations of training (best result achieved in 3.5 days), while the model init. PREVALENT takes about 4.5 days[4] (best result achieved in 1 day). Using wall-clock time in Fig. 4 as the x-axis will enhance the discrepancy between the three models. These results suggest that the pre-trained generic visiolinguistic knowledge is beneficial to the learning of VLN, and pre-training especially for navigation skills allows the agent to learn better in fine-tuning.

5. Conclusion

In this paper, we introduce recurrence into Vision-and-Language BERT and rely on its original architecture to recognise time-dependent inputs. Such innovation allows V&L BERT to address problems with a partially observable Markov decision process, and allows the learning of downstream tasks to adequately benefit from the pre-trained generic V&L knowledge. For VLN, our proposed VLN↻BERT applies BERT itself as the navigator network, which achieves SoTA performance in R2R [3] and REVERIE [44]. Moreover, results suggest that V&L BERT with recurrence is capable of VLN and REF multi-tasking.

Future Work As suggested by the significant improvement on R2R [3] achieved by our VLN↻BERT, we expect that the model can improve the performance of other navigation settings such as street navigation [6] and navigation in continuous environments [22]. In this paper, we only apply our recurrent BERT for VLN. However, we believe that it has a huge potential in addressing other tasks which require sequential interactions/decisions, such as language and visual dialog [5, 8, 21, 25, 45], dialog navigation [9, 40, 41] and action anticipation for reactive robot response [1, 13, 20].

[4]Time includes evaluation on the validation splits every 2,000 iterations. Matterport 3D Simulator v0.1 is applied, whereas the latest version supports batches of agents so it is much more efficient.

References

[1] Mohammad Sadegh Aliakbarian, Fatemeh Sadat Saleh, Mathieu Salzmann, Basura Fernando, Lars Petersson, and Lars Andersson. Viena2: A driving anticipation dataset. In *Asian Conference on Computer Vision*, pages 449–466. Springer, 2018. 8

[2] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. 5

[3] 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: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018. 1, 2, 3, 5, 6, 7, 8

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[5] Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4516–4525, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. 8

[6] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12538–12547, 2019. 2, 8

[7] 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 *European Conference on Computer Vision*, 2020. 1, 2, 4

[8] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 326–335, 2017. 8

[9] Harm de Vries, Kurt Shuster, Dhruv Batra, Devi Parikh, Jason Weston, and Douwe Kiela. Talk the walk: Navigating new york city through grounded dialogue. *arXiv preprint arXiv:1807.03367*, 2018. 8

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 1, 2, 3

[11] 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 *Advances in Neural Information Processing Systems*, pages 3314–3325, 2018. 2, 6

[12] Tsu-Jui Fu, Xin Eric Wang, Matthew Peterson, Scott Grafton, Miguel Eckstein, and William Yang Wang. Counterfactual vision-and-language navigation via adversarial path sampler. In *European Conference on Computer Vision (ECCV)*, 2020. 2

[13] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6252–6261, 2019. 8

[14] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13137–13146, 2020. 1, 2, 5, 6, 7, 8

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2, 4

[17] Yicong Hong, Cristian Rodriguez, Yuankai Qi, Qi Wu, and Stephen Gould. Language and visual entity relationship graph for agent navigation. *Advances in Neural Information Processing Systems*, 33, 2020. 2, 6

[18] Yicong Hong, Cristian Rodriguez, Qi Wu, and Stephen Gould. Sub-instruction aware vision-and-language navigation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3360–3376, Online, Nov. 2020. Association for Computational Linguistics. 7

[19] Liyiming Ke, Xiujun Li, Yonatan Bisk, Ari Holtzman, Zhe Gan, Jingjing Liu, Jianfeng Gao, Yejin Choi, and Siddhartha Srinivasa. Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6741–6749, 2019. 2, 6, 7

[20] Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):14–29, 2015. 8

[21] Satwik Kottur, José M. F. Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. CLEVR-dialog: A diagnostic dataset for multi-round reasoning in visual dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 582–595, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. 8

[22] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language

navigation in continuous environments. In *European Conference on Computer Vision*, 2020. 2, 8

[23] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 5

[24] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412, 2020. 2

[25] S Lee, H Schulz, A Atkinson, J Gao, K Suleman, L El Asri, M Adada, M Huang, S Sharma, W Tay, et al. Multi-domain task-completion dialog challenge. *Dialog System Technology Challenges*, 8, 2019. 8

[26] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Association for the Advancement of Artificial Intelligence*, pages 11336–11344, 2020. 1, 2, 4

[27] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. HERO: Hierarchical encoder for Video+Language omni-representation pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2046–2065, Online, Nov. 2020. Association for Computational Linguistics. 2

[28] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 1, 2, 4, 5

[29] Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah A Smith, and Yejin Choi. Robust navigation with language pretraining and stochastic sampling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1494–1499, 2019. 1, 2, 6

[30] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. 1, 2, 4, 5, 6, 8

[31] Yikang Li, Nan Duan, Bolei Zhou, Xiao Chu, Wanli Ouyang, Xiaogang Wang, and Ming Zhou. Visual question generation as dual task of visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6116–6124, 2018. 2

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 5

[33] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23, 2019. 1, 2, 4

[34] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10437–10446, 2020. 2

[35] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. Univilm: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. 2

[36] 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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. 2, 5, 6

[37] 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 Proceedings of the European Conference on Computer Vision*, 2020. 1, 2

[38] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016. 5

[39] Duy-Kien Nguyen and Takayuki Okatani. Multi-task learning of hierarchical vision-language representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10492–10501, 2019. 2

[40] Khanh Nguyen and Hal Daumé III. Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 684–695, 2019. 2, 8

[41] Khanh Nguyen, Debadeepta Dey, Chris Brockett, and Bill Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12527–12537, 2019. 2, 8

[42] Subhojeet Pramanik, Priyanka Agrawal, and Aman Hussain. Omninet: A unified architecture for multi-modal multi-task learning. *arXiv preprint arXiv:1907.07804*, 2019. 2

[43] Yuankai Qi, Zizheng Pan, Shengping Zhang, Anton van den Hengel, and Qi Wu. Object-and-action aware model for visual language navigation. In *European Conference on Computer Vision*, 2020. 2

[44] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[45] Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. Towards scalable multi-domain

conversational agents: The schema-guided dialogue dataset. In *AAAI*, 2020. 8

[46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 5

[47] Kurt Shuster, Da Ju, Stephen Roller, Emily Dinan, Y-Lan Boureau, and Jason Weston. The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2453–2470, Online, July 2020. Association for Computational Linguistics. 2

[48] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2019. 1, 2, 4

[49] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7464–7473, 2019. 2

[50] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5103–5114, 2019. 1, 2, 4, 5

[51] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of NAACL-HLT*, pages 2610–2621, 2019. 2, 5, 6, 7, 8

[52] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406, 2020. 1, 2, 3

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 3

[54] Hu Wang, Qi Wu, and Chunhua Shen. Soft expert reward learning for vision-and-language navigation. In *European Conference on Computer Vision*, 2020. 2

[55] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6629–6638, 2019. 2, 6

[56] Xin Wang, Vihan Jain, Eugene Ie, William Yang Wang, Zornitsa Kozareva, and Sujith Ravi. Environment-agnostic multitask learning for natural language grounded navigation. In *European Conference on Computer Vision*, 2020. 2

[57] Zekun Yang, Noa Garcia, Chenhui Chu, Mayu Otani, Yuta Nakashima, and Haruo Takemura. Bert representations for video question answering. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1556–1565, 2020. 2

[58] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1307–1315, 2018. 7

[59] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 5

[60] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10012–10022, 2020. 2, 6

[61] Wang Zhu, Hexiang Hu, Jiacheng Chen, Zhiwei Deng, Vihan Jain, Eugene Ie, and Fei Sha. BabyWalk: Going farther in vision-and-language navigation by taking baby steps. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2539–2556. Association for Computational Linguistics, 2020. 2, 4