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.

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$\diamondsuit$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 CPU 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$\diamondsuit$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$\diamondsuit$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$\diamondsuit$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.

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\textsuperscript{C}BERT.

3.1. VLN Background

The problem of VLN can be formulated as follows: Given a natural language instruction $U$ 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 $s_t$ to a new state $s_{t+1}$. The state consists of the navigational history and the current spatial position defined by a triplet $(C_t, \theta_t, \phi_t)$, where $C_t$ is a viewpoint on the pre-defined connectivity graph of the environment [3], and $\theta_t$ and $\phi_t$ 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 $X_{t-1} \in \mathbb{R}^{I \times hd_h}$ 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 $X_{t-1}$ as

$$Q = X_{t-1}W^Q_{l,k}, K = X_{t-1}W^K_{l,k}, V = X_{t-1}W^V_{l,k}$$

$$H_{l,k} = \text{Softmax} \left( \frac{QK^T}{\sqrt{d_h}} \right) V$$

where $W^Q$, $W^K$ and $W^V \in \mathbb{R}^{hd_h \times d_h}$ 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

$$H_l = [H_{l,1}; \ldots; H_{l,h}]W^O_l$$

\(^1\)All $W$ in this section denotes learnable linear projections.
where $h$ is the total number of heads, $[;]$ denotes concatenation and $W^O \in \mathbb{R}^{h \times d_x \times d_y}$ is a learned linear projection. Finally, the output of layer $l$ is formulated by

$$H^l_i = \text{LayerNorm}(H^l_i + X_{i-1})$$ (4)

$$X' = \text{ReLU}(H^l_i W^F_i) W^F_i$$ (5)

$$X_i = \text{LayerNorm}(H^l_i + X'_i)$$ (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 $s_{t-1}$, the language tokens $X$, the visual tokens $V_t$, for scene, and the visual tokens $O_t$ for objects (only in REVERSE [44]). Then, it performs self-attention over these cross-modal tokens to capture the textual-visual correspondence for inferring the action probabilities $p^a_t$ and the object grounding probabilities $p^o_t$ (only in REVERSE [44]):

$$s_t, p^a_t, p^o_t = \text{VLN} \odot \text{BERT}(s_{t-1}, X, V_t, O_t)$$ (7)

**Language Processing** At initialisation ($t=0$), a sequence of words consisting of the classification token $[CLS]$, the language tokens of the instruction $U$ 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 $s_0$, to inherit such function — initialise an agent’s state which is aware of the entire navigation task.

$$s_0, X = \text{VLN} \odot \text{BERT}([CLS], U, [SEP])$$ (8)

During navigation steps ($t>0$), unlike the state token $s_t$ or the visual tokens $V_t$ and $O_t$, which performs self-attention with respect to the entire input sequence, the language tokens $X$ 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 the instruction which does not need to be further encoded in the model at the initialisation step as a deep representation of the object features $I^o_t$ to the same space as the BERT token as $V_t = I^o_t W^g$. 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 $I^o_t$ as additional visual tokens in the input sequence. Similarly, the features will be projected onto the token space as $O_t = I^o_t W^g$ and fed into the model. The object clues can provide valuable information about the important land marks 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 $s_t$ 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 [6] 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 $s_t$ 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 $Q^{t}_{l,k}$ and $K^{t}_{l,k}$ 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:

$$A_{l,k}^{s,t} = \frac{Q^{t}_{l,k} K^{t}_{l,k} \top}{\sqrt{d_h}}$$ (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:

$$A^{s,t}_l = \text{Softmax}(A^{s,t}_l) = \text{Softmax} \left( \frac{1}{K} \sum_{k=1}^{K} A_{l,k}^{s,t} \right)$$ (10)
Similarly, the visual attention scores $A_t^{s,v}$ and weights $\tilde{A}_t^{s,v}$ 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:

$$F_t^x = \tilde{A}_t^{s,x} X \quad \text{and} \quad F_t^v = \tilde{A}_t^{s,v} V_t$$

(11)

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

$$s_t^f = [s_t^f; F_t^x \odot F_t^v] W^v$$

(12)

where $s_t^f$ 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 $s_t^f = [s_t^f; F_t^v] W^v$. 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 $a_t$ into the state token as:

$$s_t = [s_t^f; a_t] W^s$$

(13)

where $s_t$ 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-visibility 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 $p_t^{\alpha} = \tilde{A}_t^{s,\alpha}$ (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 $p_t^{\alpha} = \tilde{A}_t^{s,\alpha}$, where $\tilde{A}_t^{s,\alpha}$ 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 $p_t^\alpha$ and measures the advantage $A_t$ 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

$$L = - \sum_t a_t^\alpha \log(p_t^\alpha) A_t - \lambda \sum_t a_t^\alpha \log(p_t^\alpha)$$

(14)

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

3.5. Adaptation

We initialise the parameters of VLN$\odot$BERT from OS-CAR [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 is set to 16 and the network is trained for 300,000 iterations. For REVERIE, we use batch size 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].

References

[1] Half for RL and half for IL in each iteration, corresponding to the first and the second term in Eq. 14, respectively.
### 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. 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, our method obtained...
maintains 24.62% SR and 19.48% SPL for navigation, as well as 12.65% RGS and 10.00% RGSP 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 selected 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 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.
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.

<table>
<thead>
<tr>
<th>Models</th>
<th>V&amp;L BERT (init. OSCAR)</th>
<th>R2R Validation Seen</th>
<th>R2R Validation Unseen</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Language</td>
<td>Vision</td>
<td>State</td>
</tr>
<tr>
<td>Baseline [51]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Full model</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

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.

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 days⁴ to complete 600,000 iterations of training (best result achieved in 3.5 days), while the model init. PREVALENT takes about 4.5 days⁴ (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 observ-

---

⁴Time 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


conversational agents: The schema-guided dialogue dataset. In AAAI, 2020. 8
[48] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vi-bert: Pre-training of generic visual-linguistic representations. In International Conference on Learning Representations, 2019. 1, 2, 4