Multimodal Transformer with Variable-length Memory for Vision-and-Language Navigation

Chuang Lin[1] *, Yi Jiang[2], Jianfei Cai[1], Lizhen Qu[1], Gholamreza Haffari[1], and Zehuan Yuan[2]

[1] Monash University [2] ByteDance

Abstract. Vision-and-Language Navigation (VLN) is a task that an agent is required to follow a language instruction to navigate to the goal position, which relies on the ongoing interactions with the environment during moving. Recent Transformer-based VLN methods have made great progress benefiting from the direct connections between visual observations and language instructions via the multimodal cross-attention mechanism. However, these methods usually represent temporal context as a fixed-length vector by using an LSTM decoder or using manually designed hidden states to build a recurrent Transformer. Considering a single fixed-length vector is often insufficient to capture long-term temporal context, in this paper, we introduce Multimodal Transformer with Variable-length Memory (MTVM) for visually-grounded natural language navigation by modeling the temporal context explicitly. Specifically, MTVM enables the agent to keep track of the navigation trajectory by directly storing activations in the previous time step in a memory bank. To further boost the performance, we propose a memory-aware consistency loss to help learn a better joint representation of temporal context with random masked instructions. We evaluate MTVM on popular R2R and CVDN datasets. Our model improves Success Rate on R2R test set by 2% and reduces Goal Process by 1.5m on CVDN test set. Code is available at: `https://github.com/clin1223/MTVM`.

Keywords: Vision-and-language Navigation, Multimodal Transformer

1 Introduction

Enabling robots to assist humans in real world has been desired so long in AI [4,41,11]. To achieve it, one crucial capability of robots is to be able to follow human instructions to navigate their environments. Vision-and-Language Navigation (VLN) is the task where an embodied agent is required to follow language instructions to navigate to a goal position. Specifically, the agent is given a detailed instruction, like *"Head a bit ahead and towards the double doors on the left towards the kitchen. Stop upon reaching the counter."* At each step, then, the agent observes the panorama view of its surrounding environment and

* This work was performed while Chuang Lin worked as an intern at ByteDance.

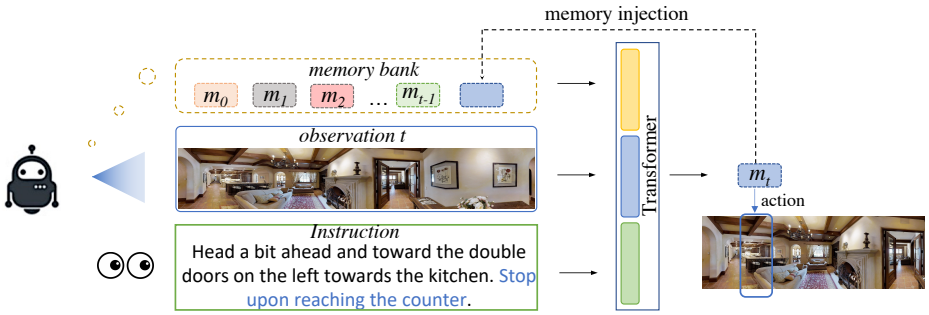

Fig. 1. In contrast to most existing methods that utilize a fixed-length vector to represent temporal context, we equip the agent with the capability to model long-term dependency. At each step t, MTVM takes all the tokens stored in the memory bank as the temporal context input. After making a decision, it adds a memory token m_t by simply reusing the output activation corresponding to the action at step t.

makes a decision for the direction to move in the next step, until it reaches the desired goal position.

Recently, many methods [3,19,24,34,30,9,27,15,40,50,44,46] have been proposed for the VLN task. Most of the literature adopts the encoder-decoder framework to encode the instruction and visual observations, and then decode the action sequence. Recent VLN studies [36,23,31,13,12] have shown great performance by directly modeling cross-modal vision-language modelling with Transformer. Different from other vision and language tasks, e.g. VQA and image captioning that learn relationships between each individual image and its corresponding text, VLN aims to learn the joint representation between each instruction and a series of observations by interacting with the environment. Thus, taking the temporal context into account is the key to ground the instruction onto the observations, figuring out what has been completed, what is next, and where to go. A straightforward way is to directly encode all the past observations [35], which however misses record cross-modal history and also increases the training cost as the path grows. Further, [16] employs the recurrent hidden state to inject temporal information into Transformer and [36,13] use the encoder-decoder structure with an additional LSTM to encode the temporal context. Nevertheless, a single hidden state vector is not expressive enough to encode the whole history of interactions with environment in Transformer. It is very challenging to align such hidden state at time t with the corresponding sub-instruction for decision making.

To address this challenge, we propose a Multimodal Transformer with Variable-length Memory (MTVM) framework for VLN. Instead of using hidden states or an LSTM to encode temporal context, we find that it is simple and effective to directly reuse the cross-modal Transformer activations obtained in the previous steps. Storing past activations in an explicit memory bank allows

to explicitly model the cross-modal history. Moreover, the Transformer architecture naturally accommodates variable-length memory token inputs. In this way, the agent is able to easily update the temporal context by adding the current output activation m_t, corresponding to the action at step t, into the memory bank, as shown in Figure 1.

Thanks to the explicit cross-modal memory bank, we further design a memory-aware consistency loss to boost the navigation performance. The consistency loss aims to help cross-modal alignment by learning the relations between the previous activations and the language instruction. Specifically, we randomly mask out some instruction words and force the model output distribution to be consistent with that of the original unmasked instruction. In this way, the model avoids overfitting to the language modality with the help of the explicit memory bank.

Our contributions can be summarized as follows:

1. We propose MTVM that allows the agent to capture temporal context without distance dependency by simply reusing the previous cross-model activations corresponding to the actions.
2. We design a memory-aware consistency loss to learn strong relations between instruction and temporal context to further boost the navigation performance.
3. We conduct extensive experiments on R2R and CVDN datasets, improving Success Rate by 2% on R2R and reducing Goal Progress by 1.5m on CVDN compared to strong baselines.

2 Related Work

Vision-and-Language Navigation. VLN [1] is a task that requires an agent to follow a nature-language instruction to navigate in a photo-realistic environment to a goal location. In this process, the given instruction describes the trajectory in detail and the embodied agent needs to move through the scene with first person views as observations. Following [1], several navigation tasks [5,42,33,34,37] have been further proposed for interactions with surrounding environments. In particular, different from [1] collecting data from an indoor environment, [5] extends the navigation environment to real-life visual urban streets. [42] introduces navigating according to several question-answering pairs in a dialog history. [20] further extends the dialog navigating task by taking the full dialogue and the whole navigation path as one instance. [34] and [33] consider object-finding tasks [38,47,26,25] by requesting and interpreting simulated human assistants. [37] requires the agent to navigate to an appropriate location and identify the target object. [21] proposes a multilingual datasets for VLN, which including more visual entities and avoiding language bias.

As a practical task in real-world applications, VLN has made incredible progress in recent years. [24] uses adversarial attacking to capture key information from long instructions for a robust navigation. The progress monitor in [30] aims to estimate the navigation progress explicitly as a multi-task learning,

supervised by the normalized distance to the goal. RCM [44] enforces cross-modal grounding both locally and globally via a matching critic providing rewards for reinforcement learning.

In vision-and-language navigation setting, it is difficult to collect enough annotated data due to the large navigation space. [9] synthesizes new instructions where the speaker model helps the agent by additional route-instruction pairs to expand the limited training data. To make further advances, [46] proposes an instruction-trajectory compatibility model to improve the instruction evaluation. [40] proposes an environmental dropout method based on the view consistency to mimic novel and diverse environments. From a different perspective, REM [27] reconnect the seen scenes to generate augmented data via mixing up environments. To further understand the relations between the instructions and scenes, [15] and [36] take the objects in scenes and the corresponding words in instructions as the minimal units of encoding. AuxRN [50] introduces additional training signals including explaining actions, predicting next orientation, etc. , to help acquire semantic knowledge. In contrast, our method focuses on modeling the temporal context to help the alignment between language and observations.

Multi-Modal Transformers. The Transformer [43] architecture has shown great effectiveness in vision and language tasks [39,29,7,22,18,49,10,45]. Most of the vision-and-language tasks focus on the joint embedding learning with individual pairs of an image and its corresponding language, such as VQA, image captioning, and text-to-image retrieval. Different from these tasks, VLN is a Markov Decision Process, which learns the joint representation between the instruction and a series of observations along the corresponding trajectory. Inspired by the success of BERT [8], PRESS [23] first introduces a large-scale pre-trained language model to VLN for text representations. As cross-modal joint learning is the key for VLN task, VLN-BERT [31] and PREVALENT [13] develop Transformer-based model in a self-supervised manner on image-text pairs from the web and image-text-action triplets from R2R dataset [1], respectively. [16] and [36] adapt pre-trained V&L BERT to VLN task by leveraging the hidden state representations with the learned linear projection or LSTM. Recently, HAMT [6] and Episodic Transformer [35] also propose to model the history information explicitly by directly encoding all past observations and the actions. Our key insight is: *only explicitly modelling the history observations is not good enough; instead, explicitly modelling the history interactions between observations and the instruction is more critical since it helps figure out the progress of the navigation trajectory.*

3 Methods

3.1 Overview

Formally, at the beginning of each episode, the agent is given a nature language instruction $x = \langle x_1, x_2, \ldots, x_L \rangle$, where L is the length of the instruction and x_i denotes a word. VLN task requires the agent to follow the instruction to navigate from a start position to the goal location. At each step t, the agent is able to

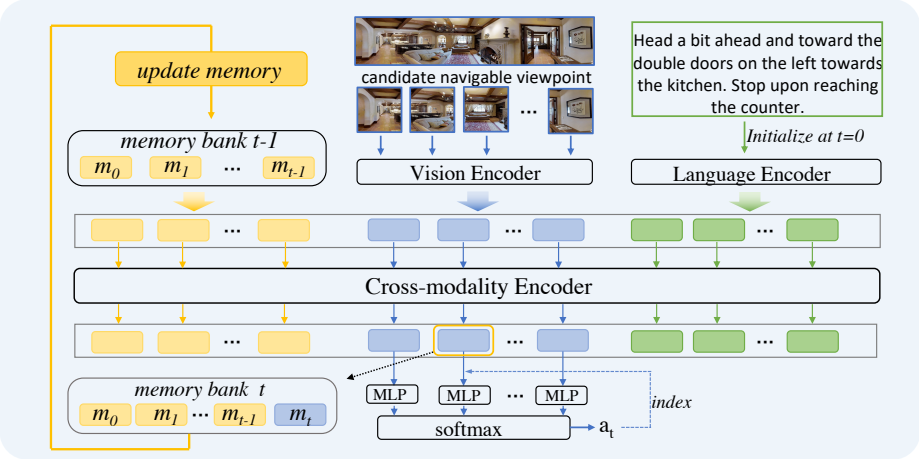

Fig. 2. The general framework of our proposed MTVM framework. At each step, we concatenate temporal context in the memory bank, together with visual features and language features as input. After making decision, we update the memory bank by storing the output activation that corresponding to the action.

observe the surrounding environment in a panoramic view $o_t = \langle o_t^1, o_t^2, \ldots, o_t^{36} \rangle$ comprised by 36 single view images. Figure 2 gives an overview of our proposed Multimodal Transformer with Variable-length Memory (MTVM). At each step, our MTVM directly interacts with visual information, language information, and history information to make the action decision. After that, we update the memory bank by reusing the activation of the Transformer output according to the action decision. Moreover, a consistency loss is introduced to measure the distance between the output distributions of the full instruction and a randomly masked instruction to help the cross-modal alignment. Note that the instruction masking is only used in training but not in inference.

3.2 Memory-based Multimodal Transformer

As VLN is a Markov decision process [1], an embodied agent needs to pay attention to the temporal context information during its navigation. The general Transformer is not enough to model the instruction and the observations due to the lack of the temporal context. At each navigation step, an agent needs to ground an instruction to which part has finished and which part is the next.

MTVM learns the cross-modal alignment to encourage matching the completed part of the instructions with the past trajectory. Our memory bank enables the agent to be aware of the navigation process by directly interacting with the previous actions so that it can ground the sub-instructions as guidance. In this way, it becomes easier for the agent to locate the sub-instruction to gain

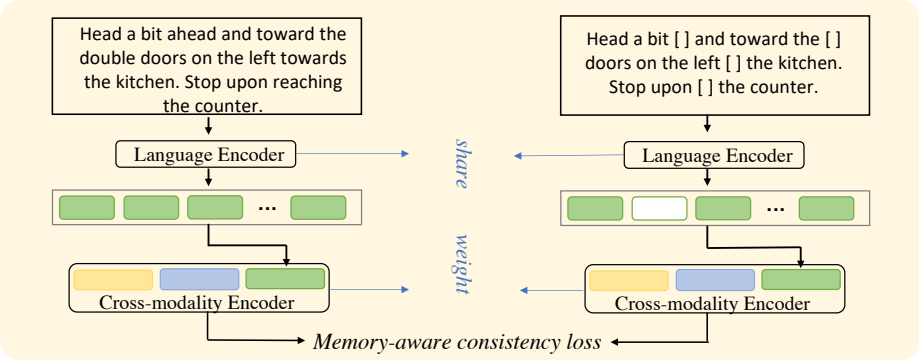

Fig. 3. The proposed memory-aware consistency loss. During training, we randomly mask out some words to help the alignment between language and temporal context, avoiding model overfitting to the language modality.

useful information to select the candidate direction from the current-step observation. We construct our model following the vision and language pretrained work [39,13], which consists of a language encoder, a vision encoder and a cross-modality encoder.

Language Encoder. The language encoder is a standard multi-layer transformer with self-attention. At the beginning of an episode, we feed the instruction to the language encoder $\mathcal{S}$ to get the language representation $X = \mathcal{S}(x)$.

Vision Encoder. The vision encoder is a convolution network to encode each single view image o_t^i to a 2048-dimensional visual feature v_t^i. A 128-dimensional directional feature d_t^i by repeating the trigonometric function representation [9] is concatenated with the visual feature v_t^i to represent the orientation of each single view $V_t^i = [v_t^i; d_t^i]$. For each step, we have $V_t = \{V_t^1, V_t^2, \ldots, V_t^K\}$ as the visual representation, where K is the number of candidate directions.

Cross-modality Encoder. In order to learn cross-modality representations, the cross-modality encoder $\mathcal{C}$ is composed of self-attention layers and cross-attention layers, where cross-attention layers treat one modality as query and the other as key and value to exchange the information and align the entities between the two modalities. In particular, we feed language representation X, vision representation V_t, and previous activations M_t to the cross-modality encoder $\mathcal{C}$ as

$$\widehat{X}, \widehat{M_t}, \widehat{V_t} = \mathcal{C}(X, [M_t; V_t]), \tag{1}$$

where $[;]$ denotes concatenation. Then, the action prediction head takes the output $\widehat{V_t}$ to make the action decision for this step: $a_t = MLP(\widehat{V_t})$.

At the end of each step, we update the memory bank by reusing the output activations $\widehat{V_t}^k$ according to the current agent action decision as

$$M_t \leftarrow (M_{t-1}, \left[\widehat{V_t}^k; d_t^k\right]) \tag{2}$$

where k is the index of the selected vision output and d_t^k is the corresponding directional feature of t step action.

3.3 Memory-aware Consistency Loss

As aforementioned, the key challenge in VLN is that the embodied agent needs to be aware of the progress of the navigating trajectory by learning the cross-modal representation. However, the existing studies [17,1] show that the agent tends to overfit the instructions, which could be due to large variations in the visual modality. In order to avoid the model from overfitting a single modality, we design a memory-aware consistency loss. By randomly dropping some words in the instruction, we force the model to learn strong representations among language, vision, and temporal context from the cross-modality encoder.

Specifically, given an instruction x, we random drop some words with a fixed probability and obtain

$$x' = RandomDrop(x). \tag{3}$$

Both x and x' are then encoded by language encoder $\mathcal{S}$ to produce the instruction representations X and X', respectively. Same as the instruction feature X, X' is also fed through the cross-modality encoder $\mathcal{C}$ with the same history and vision representations as Eq. (1):

$$\widehat{X'}, \widehat{M_t'}, \widehat{V_t'} = \mathcal{C}(X', [M_t; V_t]), \tag{4}$$

Although some words are discarded, we expect the similarities between the instruction features X and X' and their corresponding outputs are preserved. Concretely, we generate the probability vectors for the outputs of the language encoder and cross-modality encoder respectively with the Softmax layer. By minimizing the bidirectional Kullback-Leibler (KL) divergence between the outputs of the full instruction and the randomly dropped instruction, the consistency loss is defined as

$$\begin{aligned} \mathcal{L}_{consis} = \lambda_s(&\mathcal{D}_{KL}(X \| X') + \mathcal{D}_{KL}(X' \| X)) \\ +\lambda_m(&\mathcal{D}_{KL}(\widehat{X'}, \widehat{M_t'}, \widehat{V_t'} \| \widehat{X}, \widehat{M_t}, \widehat{V_t}) + \\ &\mathcal{D}_{KL}(\widehat{X}, \widehat{M_t}, \widehat{V_t} \| \widehat{X'}, \widehat{M_t'}, \widehat{V_t'})), \end{aligned} \tag{5}$$

where λ_s and λ_m are the weights to balance the distance losses. The first term aims to prevent the agent from overfitting the special words (such as route words), while the second term aims to avoid overfitting the language modality.

3.4 Training

Following the existing VLN works, we apply the mixture of Imitation Learning (IL) and Reinforcement Learning (RL) strategies [44,40]. In IL, the agent learns to follow the teacher action a_t^* of the ground-truth path at each step t by minimizing the negative log probability loss function. In RL, the agent learns from

rewards by using A2C algorithm [32], where sampling the action a_t^s from the agent's action distribution a_t, the agent will get rewards if successfully arriving at the target within 3m ($t = T$) or reducing the distance to the target after taking the action ($t < T$). Besides, we consider the similarity of the agent path and the ground-truth path as a reward to encourage the agent follow the instruction to move closer to the target. The overall loss function can be written as:

$$\begin{aligned}\mathcal{L} &= \lambda_l \mathcal{L}_{IL} + \mathcal{L}_{RL} + \mathcal{L}_{consis} \\ &= \lambda_l \sum_{t=0}^{T-1} -a_t^* log(a_t) + \sum_{t=0}^{T-1} -a_t^s log(a_t) A_t + \mathcal{L}_{consis}\end{aligned} \tag{6}$$

where λ_l is a trade-off weight for IL loss, T is the length of the navigation path, and A_t is the advantage calculated by A2C algorithm [32]. We alternately train the agent with IL and RL strategies while applying the consistency loss in both.

4 Experiments

4.1 Setup

Datasets: We evaluate MTVM on the Room-to-Room dataset (R2R) [1] and Cooperative Vision-and-Dialog Navigation dataset (CVDN) [42] in 3D environments based on Matterport3D Simulator [2]. The simulated environments include 90 different housing scenes. R2R dataset provides fully specified instructions describing the steps necessary to reach the goal, while CVDN dataset provides an ambiguous and underspecified goal location and human-human dialogs to guide the agent. R2R splits the dataset into the training set consisting of 61 environments with 14,025 instructions, the seen validation set consisting of the same 61 environments with 1,020 instructions, and the unseen validation consisting of another 11 environments with 2,349 instructions, while the test consists of the remaining 18 environments with 4,173 instructions. CVDN contains 4742 training, 382 seen validation, 907 unseen validation, and 1384 unseen test instances.

Evaluation Metrics: For R2R, we use its three standard metrics: Navigation Error (NE) defined as the distance (in meters) from the stop viewpoint to the goal position, Success Rate (SR), and Success rate weighted by Path Length (SPL), where SPL is regarded as the primary metric. For CVDN, following [42], we evaluate the performance on the navigation from dialog history (NDH) task by Goal Progress, which measures how much reduction in meters the agent makes towards the goal. There are three settings depending on the supervised strategy. *Oracle* indicates the agent regarding the shortest path as ground truth and *Navigator* indicates learning from the navigator path (maybe not be the optimal navigation). *Mixed* supervision means to learn from the navigator path if it reaches the goal point; otherwise learn from the shortest path.

Implementation Details: To leverage vision and language pre-trained models, we initialize the language encoder and the cross-modality encoder by a pre-train VLN model PREVALENT [13]. Following PREVALENT [13] and

Table 1. Comparisons of the VLN performance on R2R dataset in a single-run setting. The best results are in bold font. The set of methods at the bottom are Transformer based solutions, whose model parameters are initialized by the pre-trained vision-and-language BERT. The set of methods in the middle are non-Transformer based solutions.

Methods	Validation Seen			Validation Unseen			Test		
	NE↓	SR↑	SPL↑	NE↓	SR↑	SPL↑	NE↓	SR↑	SPL↑
Random	9.45	16	-	9.23	16	-	9.79	13	12
Human	-	-	-	-	-	-	1.61	86	76
Speaker-Follower [9]	3.36	66	-	6.62	35	-	6.62	35	28
Self-monitoring [30]	3.22	67	58	5.52	45	32	5.67	48	35
RCM [44]	3.53	67	-	6.09	43	-	6.12	43	38
FAST-Short [19]	-	-	-	4.97	56	43	5.14	54	41
EnvDrop[40]	3.99	62	59	5.22	52	48	5.23	51	47
DR-Attacker [24]	3.52	70	67	4.99	53	48	5.53	52	49
AuxRN [50]	3.33	70	67	5.28	55	50	5.15	55	51
RelGraph [15]	3.47	67	65	4.73	57	53	4.75	55	52
PRESS [23]	4.39	58	55	5.28	49	45	5.49	49	45
PREVALENT [13]	3.67	69	65	4.71	58	53	5.30	54	51
ORIST [36]	-	-	-	4.72	57	51	5.10	57	52
VLN⟳BERT [16]	2.90	72	68	3.93	63	57	4.09	63	57
Ours	**2.67**	**74**	**69**	**3.73**	**66**	**59**	**3.85**	**65**	**59**

VLN⟳BERT [16], we train the agent on the original training data and the augmented data provided by [13]. The vision encoder is a fixed ResNet-152 [14] pre-trained on Place365 [48] provided by R2R dataset. The experiments are conducted on 3 V100 GPUs. We train the model 10,000 iterations and adopt the early stopping strategy when the model achieves the best performance on the evaluation metric. The learning rate is fixed to 5e−6 with an AdamW optimiser [28]. The parameters λ_s and λ_m are respectively set to 0.6 and 0.2 and λ_{IL} is set to 0.2. We find different levels of dropping words are all helpful, and we fix the word dropping probability to 0.5.

4.2 Comparisons with SoTA

Table 1 shows the performance comparisons of different VLN methods on R2R dataset in a single-run setting. It can be seen that our model performs the best on all the metrics under both unseen validation and test sets, suggesting the good generalizing ability. Compared with other transformer-based methods including PRESS [23], ORIST [36] and VLN⟳BERT [16] which also initialize their models using the pre-trained ones [13,7], our method is at least 2% higher in terms of SPL or SR under both test and validation unseen scenarios. In addition, the lowest navigation error achieved by our model indicates that we can make the agent move closer to the target.

Table 2 shows the performance comparisons in terms of Goal Progress on CVDN dataset under the three different settings. Again, our method achieves the

Table 2. Comparisons with state-of-the-art methods in terms of Goal Progress (m) on the navigation from dialog history (NDH) task on CVDN dataset [42]. 'Ora', 'Nav' and 'mix' denote the three settings, 'Oracle', 'Navigator' and 'Mixed', respectively.

Methods	Validation Unseen			Test		
	Ora	Nav	Mix	Ora	Nav	Mix
Random	1.09	1.09	1.09	0.83	0.83	0.83
Shortest Path	8.36	7.99	9.58	8.06	8.48	9.76
Seq-to-seq [42]	1.23	1.98	2.10	1.25	2.11	2.35
PREVALENT [13]	2.58	2.99	3.15	1.67	2.39	2.44
CMN [52]	2.68	2.28	2.97	2.69	2.26	2.95
ORIST [36]	3.30	3.29	3.55	2.78	3.17	3.15
SCoA [51]	1.94	2.91	2.85	2.49	3.37	3.31
DR-Attacker [24]	3.27	4.00	4.18	2.77	2.95	3.26
Ours	**4.57**	**4.80**	**5.15**	**4.23**	**4.46**	**4.82**

best performance with significant gains on both unseen validation and test sets, demonstrating the effectiveness of handling a variety of language instructions. Note that the Shortest Path Agent takes the shortest path to the supervision goal at inference, which represents the upper bound navigation performance for an agent.

4.3 Ablation studies

Memory bank size. Recall that our method stores the activations at each step as history information in a memory bank. Here, we evaluate the model performance with different memory bank sizes. When the memory bank size is n, we only record the last n step activations; when the size is variable, it means we record every step. Note that the paths in R2R dataset are all around four to six steps. The results are shown in Figure 4. In general, a larger memory size helps, and the variable-length memory gives the best performance, suggesting the importance of explicitly storing the history information. In addition, we also show the performance of PREVALENT as our baseline (dashed lines) since our model is initialized from it. It can be seen that our model under most of the fixed-length memory banks outperforms the baseline.

Comparison of history encoding methods. We next evaluate the advantage of proposed variable-length memory bank to other baselines, including visual-only [35,6] and cross-modal interaction [16] as history encoding methods. [35] encodes oriented observations (one view of full observations) and the actions as the history information. [6] proposes a hierarchical observations and actions encoding method which is able to learning intra-panorama and inter-panorama visual information for temporal context. The experiments are under the R2R validation seen and unseen setting and measured by Success Rate (SR) and Success rate weighted by Path Length (SPL). As [35] was proposed for the ALFRED benchmark which is for household action learning from instructions

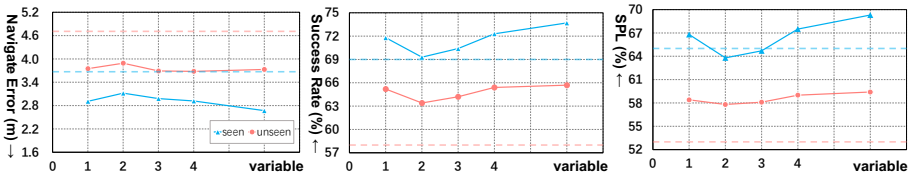

Fig. 4. Impacts of the memory bank size on seen and unseen validation sets of R2R dataset in terms of NE, SR and SPL. Solid lines are our results with different memory bank sizes, and dashed lines are the results of PREVALENT [13] from which our model is initialized.

and egocentric vision, we reproduce it by simply replacing our history encoding with the corresponding methods.

As shown in Table3, we have the following observations from the results: (1) The visual-only method [35] that only encodes the past oriented views and actions obtains the worst performance. This is obvious, because only recording the oriented views may ignore significant information in the trajectory. For instance, "Go straight passing the fridge", "fridge" might not be in the oriented visual observations, which is essential for the agent to record history. (2) Compared with visual-only methods, the cross-modal history encoding methods achieves better performance in most settings, which demonstrates the effectiveness of considering the multimodal interactions as history for VLN. Only modelling history observations provides the visual information in temporal context, but is insufficient to record vision and language navigation progress. (3) Our MTVM achieves the highest SR and SPL among all history encoding methods, because of the proposed variable length memory and the memory consistency loss. Compared to the typically used recurrent state, we found that cross-modal history can be better captured by simply reusing the previous cross-model activations corresponding to the actions, which is simple but effective and non-trivial. Note that for HAMT [6], we report its results with Resnet-152 as the vision encoder for fair comparison.

Table 3. Comparison of different history encoding methods in R2R setting. "Visual-only" indicates methods that encoding past observations and actions as history. "Cross-modal" indicates methods considering cross-modal interactions and actions as history.

History Encoding Methods		Val Seen		Val Unseen	
		SR↑	SPL↑	SR↑	SPL↑
Visual-only	E.T. [35] *Oriented observations*	68.1	63.6	59.0	54.5
	HAMT [6] *Hierarchical observations*	69.3	64.8	63.5	57.5
Cross-modal	VLN⟳BERT [16] *Recurrent state*	72	68	63	57
	Ours	**73.7**	**69.3**	**65.7**	**59.4**

Table 4. Impacts of our proposed memory-aware consistency loss and random word dropping. "word dropping" refers to our model without using the consistency loss but with random word dropping in language instructions for data augmentation.

Methods			Validation Unseen	
memory bank	consistency	word dropping	SR(%)↑	SPL(%)↑
✓			64.0	58.6
✓	✓		$65.7_{\uparrow 1.7}$	$59.4_{\uparrow 0.8}$
✓		✓	$64.5_{\uparrow 0.5}$	$57.8_{\downarrow 0.8}$

Table 5. Comparisons of training memory and computation cost on R2R dataset. We produce MTVM†* with the same cross-attention strategy as VLN↻BERT, where language is used as keys and values but not as queries. † indicates MTVM without the consistency loss. The best results are in bold and the second best results are underlined.

Methods	Params#	Memory	Validation Unseen	
			SR(%)↑	SPL(%)↑
VLN↻BERT	41.9M	8.6GB	63.3	57.5
MTVM†*	41.6M	8.4GB	<u>63.6</u>	<u>58.2</u>
MTVM†	68.4M	17.9GB	**64.0**	**58.6**

Impacts of consistency loss and random word dropping. Table 4 compares the results with and without our proposed consistency loss. For our MTVM model, we can see that the consistency loss significantly improves the performance. The consistency loss is designed to encourage the model to pay more attention to our explicitly modelled history tokens. Although some words are dropped during training, a lot of vision-language alignments have already been captured in the memory. It improves 1.7% and 0.8% on R2R validation unseen setting with SR and SPL metric, indicating that the agent with the memory consistency loss achieves better generalize ability.

Note that our word-drop strategy for the consistency loss is similar to conventional random word dropping used for data augmentation. Thus, we make a comparison with direct word dropping for data augmentation (denoted as "memory bank" + "word dropping") in Table 4, where we fix the word dropping rate to 0.5 in all methods. It can be seen that direct word dropping as data augmentation is not as effective as ours.

We further investigate the effect of different word dropping rates on SR and SPL in both seen and unseen validation sets of R2R dataset. Here we conduct experiments by varying the word dropping rate in $\{0.1, 0.3, 0.5, 0.7\}$. As shown in Fig 5 a), we can see that a small dropping rate (e.g., 0.1) does not perform as good as a large one (e.g., 0.5), while a too large dropping rate (e.g. 0.7) also hurts the performance. Thus, the best choice is 0.5.

Hyper-parameter sensitivity. We analyze the sensitivity of the hyper-parameters to SPL metric on R2R unseen validation set by using λ_s and λ_m in Eq. (5) as examples. The results are reported in Figure 5 b). From these re-

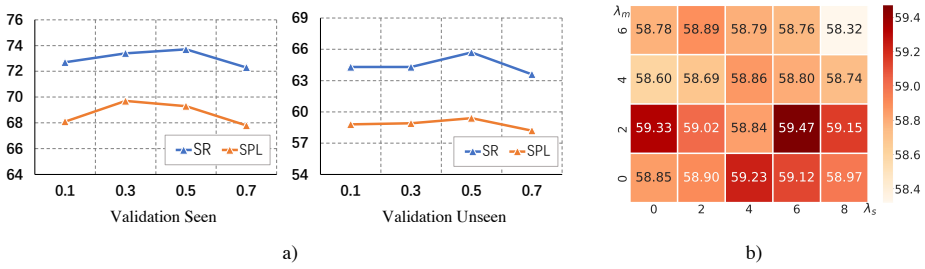

Fig. 5. a) Impact of different random word dropping rates on SR and SPL on both seen and unseen validation sets of R2R dataset. b) Sensitivity examples of the hyper-parameters in Eq. (5) to SPL metric on R2R unseen validation set. The darker the color, the better the performance.

sults, we can see that SPL is not very sensitive to the variations of λ_s and λ_m in a range around $2 \sim 8$ and we find that it is a good choice to set $\lambda_s = 6, \lambda_m = 2$.

Memory and computation Cost. Following most of the cross-modal Transformer methods [39,13], our MTVM facilitates vision-and-language interactions by bi-directional cross-attention sub-layers, where language is used as query attending to vision and vice versa. To compare with single-direction cross-modal Transformer method VLN⟳BERT [16], which only considers language tokens as keys and values but not as queries, we also develop a similar version, MTVM†*. The comparison results of VLN⟳BERT, MTVM†* and MTVM† in terms of Parameters and GPU Memory Cost are shown in Table 5. For a fair comparison with VLN⟳BERT, all the experiments are conducted on a single V100 GPU with batch size 16. With the same cross-attention strategy, compared with VLN⟳BERT, our MTVM†* archives better performance but with lower memory and computation cost. This is because VLN⟳BERT needs an additional small network to encode update its hidden states for temporal context while our MTVM†* directly reuses the previous activations. This demonstrates the efficiency and effectiveness of our proposed memory bank based Transformer design.

4.4 Visualization

To demonstrate the proposed consistency loss, we give a few visualization examples of panoramic views and language attention weights in Fig. 6. In R2R dataset, the agent needs to navigate following the instruction from the beginning to the end. Sub-figures (a) and (b) in Fig. 6 show that our MTVM model with the consistency loss achieves better navigation performance with a much shorter trajectory. In sub-figures (c) and (d), we observe that our model with the consistency loss is able to better ground the sub-instructions while MTVM without the consistency loss fails to focus on the action word at each step.

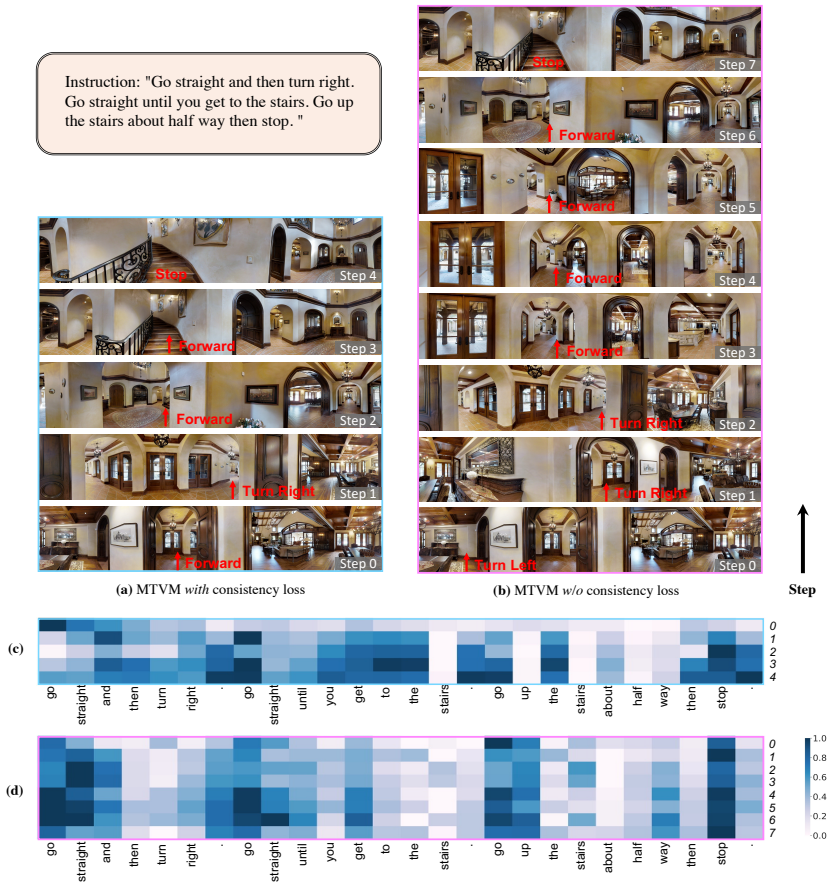

Fig. 6. Visualization examples of panoramic views and language attention weights. From sub-figures (a) and (b), it can be seen that without the consistency loss, MTVM took a longer path to reach "stairs". (c) and (d) are the language attention weights at the final layer of the cross-modality encoder corresponding to (a) and (b) at each step.

5 Conclusion

We have proposed the framework of Multimodal Transformer with Variable-length Memory (MTVM), which enables the agent explicitly model the history information in a simple and effective way. We have also designed the memory-aware consistency loss to improve the generalization ability of our model. Our MTVM has demonstrated strong performance, outperforming almost all the existing works on both R2R and CVDN dataset. We see the benefit of allowing long-range dependency for VLN task and we hope this idea can benefit other vision and language interaction tasks.

References

1. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., Van Den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3674–3683 (2018)
2. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158 (2017)
3. Chattopadhyay, P., Hoffman, J., Mottaghi, R., Kembhavi, A.: Robustnav: Towards benchmarking robustness in embodied navigation. arXiv preprint arXiv:2106.04531 (2021)
4. Chen, D., Mooney, R.: Learning to interpret natural language navigation instructions from observations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 25 (2011)
5. Chen, H., Suhr, A., Misra, D., Snavely, N., Artzi, Y.: Touchdown: Natural language navigation and spatial reasoning in visual street environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12538–12547 (2019)
6. Chen, S., Guhur, P.L., Schmid, C., Laptev, I.: History aware multimodal transformer for vision-and-language navigation. Advances in Neural Information Processing Systems **34** (2021)
7. Chen, Y.C., Li, L., Yu, L., Kholy, A.E., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: Uniter: Universal image-text representation learning. In: ECCV (2020)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., Darrell, T.: Speaker-follower models for vision-and-language navigation. arXiv preprint arXiv:1806.02724 (2018)
10. Gan, Z., Chen, Y.C., Li, L., Zhu, C., Cheng, Y., Liu, J.: Large-scale adversarial training for vision-and-language representation learning. arXiv preprint arXiv:2006.06195 (2020)
11. Guadarrama, S., Riano, L., Golland, D., Go, D., Jia, Y., Klein, D., Abbeel, P., Darrell, T., et al.: Grounding spatial relations for human-robot interaction. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 1640–1647. IEEE (2013)
12. Guhur, P.L., Tapaswi, M., Chen, S., Laptev, I., Schmid, C.: Airbert: In-domain pretraining for vision-and-language navigation. arXiv preprint arXiv:2108.09105 (2021)
13. Hao, W., Li, C., Li, X., Carin, L., Gao, J.: 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. pp. 13137–13146 (2020)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
15. Hong, Y., Rodriguez-Opazo, C., Qi, Y., Wu, Q., Gould, S.: Language and visual entity relationship graph for agent navigation. arXiv preprint arXiv:2010.09304 (2020)

16. Hong, Y., Wu, Q., Qi, Y., Rodriguez-Opazo, C., Gould, S.: Vln bert: A recurrent vision-and-language bert for navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1643–1653 (2021)
17. Hu, R., Fried, D., Rohrbach, A., Klein, D., Darrell, T., Saenko, K.: Are you looking? grounding to multiple modalities in vision-and-language navigation. arXiv preprint arXiv:1906.00347 (2019)
18. Huang, Z., Zeng, Z., Liu, B., Fu, D., Fu, J.: Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. arXiv preprint arXiv:2004.00849 (2020)
19. Ke, L., Li, X., Bisk, Y., Holtzman, A., Gan, Z., Liu, J., Gao, J., Choi, Y., Srinivasa, S.: Tactical rewind: Self-correction via backtracking in vision-and-language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6741–6749 (2019)
20. Kim, H., Li, J., Bansal, M.: Ndh-full: Learning and evaluating navigational agents on full-length dialogue. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 6432–6442 (2021)
21. Ku, A., Anderson, P., Patel, R., Ie, E., Baldridge, J.: Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. arXiv preprint arXiv:2010.07954 (2020)
22. Li, G., Duan, N., Fang, Y., Gong, M., Jiang, D.: Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 11336–11344 (2020)
23. Li, X., Li, C., Xia, Q., Bisk, Y., Celikyilmaz, A., Gao, J., Smith, N., Choi, Y.: Robust navigation with language pretraining and stochastic sampling. arXiv preprint arXiv:1909.02244 (2019)
24. Lin, B., Zhu, Y., Long, Y., Liang, X., Ye, Q., Lin, L.: Adversarial reinforced instruction attacker for robust vision-language navigation. arXiv preprint arXiv:2107.11252 (2021)
25. Lin, C., Yuan, Z., Zhao, S., Sun, P., Wang, C., Cai, J.: Domain-invariant disentangled network for generalizable object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8771–8780 (2021)
26. Lin, C., Zhao, S., Meng, L., Chua, T.S.: Multi-source domain adaptation for visual sentiment classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 2661–2668 (2020)
27. Liu, C., Zhu, F., Chang, X., Liang, X., Shen, Y.D.: Vision-language navigation with random environmental mixup. arXiv preprint arXiv:2106.07876 (2021)
28. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
29. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. arXiv preprint arXiv:1908.02265 (2019)
30. Ma, C.Y., Lu, J., Wu, Z., AlRegib, G., Kira, Z., Socher, R., Xiong, C.: Self-monitoring navigation agent via auxiliary progress estimation. arXiv preprint arXiv:1901.03035 (2019)
31. Majumdar, A., Shrivastava, A., Lee, S., Anderson, P., Parikh, D., Batra, D.: Improving vision-and-language navigation with image-text pairs from the web. In: European Conference on Computer Vision. pp. 259–274. Springer (2020)
32. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. pp. 1928–1937. PMLR (2016)

33. Nguyen, K., Daumé III, H.: Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning. arXiv preprint arXiv:1909.01871 (2019)
34. Nguyen, K., Dey, D., Brockett, C., Dolan, B.: Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12527–12537 (2019)
35. Pashevich, A., Schmid, C., Sun, C.: Episodic transformer for vision-and-language navigation. arXiv preprint arXiv:2105.06453 (2021)
36. Qi, Y., Pan, Z., Hong, Y., Yang, M.H., Hengel, A.v.d., Wu, Q.: Know what and know where: An object-and-room informed sequential bert for indoor vision-language navigation. arXiv preprint arXiv:2104.04167 (2021)
37. Qi, Y., Wu, Q., Anderson, P., Wang, X., Wang, W.Y., Shen, C., Hengel, A.v.d.: Reverie: Remote embodied visual referring expression in real indoor environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9982–9991 (2020)
38. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
39. Tan, H., Bansal, M.: Lxmert: Learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490 (2019)
40. Tan, H., Yu, L., Bansal, M.: Learning to navigate unseen environments: Back translation with environmental dropout. arXiv preprint arXiv:1904.04195 (2019)
41. Tellex, S., Kollar, T., Dickerson, S., Walter, M., Banerjee, A., Teller, S., Roy, N.: Understanding natural language commands for robotic navigation and mobile manipulation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 25 (2011)
42. Thomason, J., Murray, M., Cakmak, M., Zettlemoyer, L.: Vision-and-dialog navigation. In: Conference on Robot Learning. pp. 394–406. PMLR (2020)
43. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
44. Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.F., Wang, W.Y., Zhang, L.: Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6629–6638 (2019)
45. Wu, J., Jiang, Y., Sun, P., Yuan, Z., Luo, P.: Language as queries for referring video object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4974–4984 (2022)
46. Zhao, M., Anderson, P., Jain, V., Wang, S., Ku, A., Baldridge, J., Ie, E.: On the evaluation of vision-and-language navigation instructions. arXiv preprint arXiv:2101.10504 (2021)
47. Zhao, S., Lin, C., Xu, P., Zhao, S., Guo, Y., Krishna, R., Ding, G., Keutzer, K.: Cycleemotiongan: Emotional semantic consistency preserved cyclegan for adapting image emotions. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 2620–2627 (2019)
48. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1452–1464 (2017)

49. Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J., Gao, J.: Unified vision-language pre-training for image captioning and vqa. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 13041–13049 (2020)
50. Zhu, F., Zhu, Y., Chang, X., Liang, X.: Vision-language navigation with self-supervised auxiliary reasoning tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10012–10022 (2020)
51. Zhu, Y., Weng, Y., Zhu, F., Liang, X., Ye, Q., Lu, Y., Jiao, J.: Self-motivated communication agent for real-world vision-dialog navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1594–1603 (2021)
52. Zhu, Y., Zhu, F., Zhan, Z., Lin, B., Jiao, J., Chang, X., Liang, X.: Vision-dialog navigation by exploring cross-modal memory. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10730–10739 (2020)