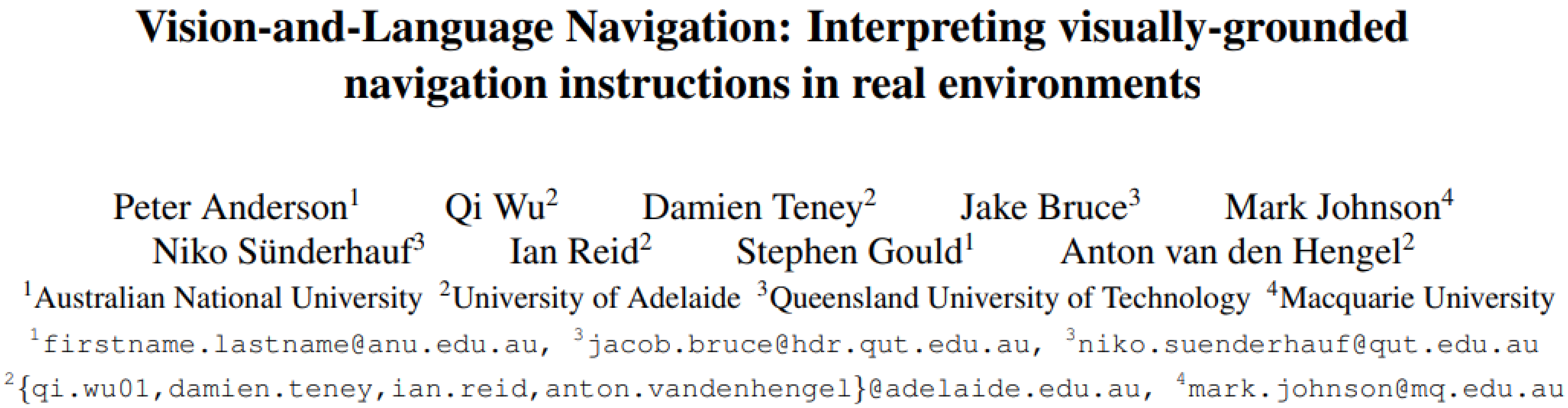


Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments

Peter Anderson[1] Qi Wu[2] Damien Teney[2] Jake Bruce[3] Mark Johnson[4]
Niko Sünderhauf[3] Ian Reid[2] Stephen Gould[1] Anton van den Hengel[2]
[1]Australian National University [2]University of Adelaide [3]Queensland University of Technology [4]Macquarie University
[1]firstname.lastname@anu.edu.au, [3]jacob.bruce@hdr.qut.edu.au, [3]niko.suenderhauf@qut.edu.au
[2]{qi.wu01,damien.teney,ian.reid,anton.vandenhengel}@adelaide.edu.au, [4]mark.johnson@mq.edu.au

CVPR 2018

Motivation

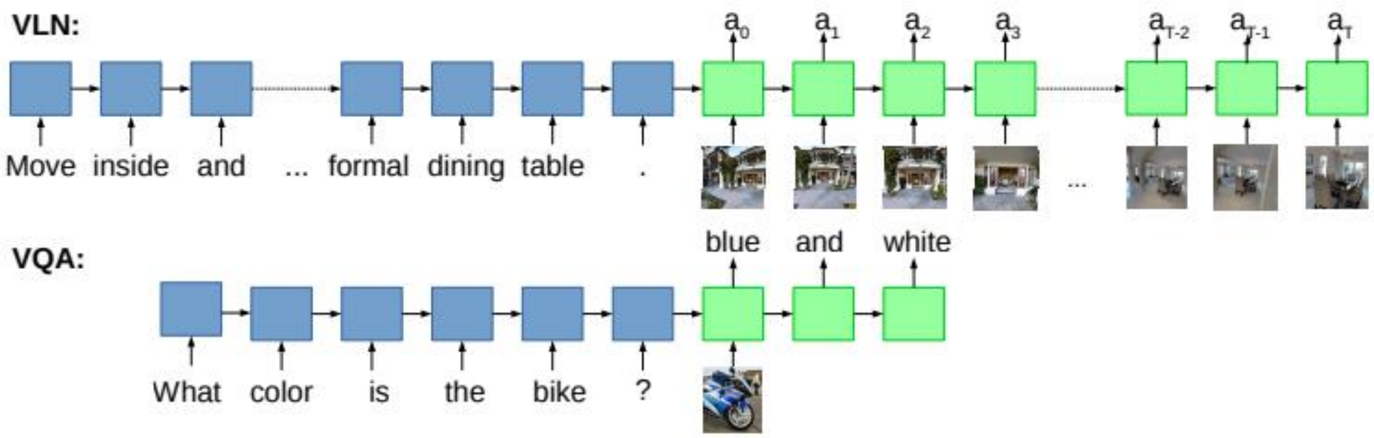


Figure 2. Differences between Vision-and-Language Navigation (VLN) and Visual Question Answering (VQA). Both tasks can be formulated as visually grounded sequence-to-sequence transcoding problems. However, VLN sequences are much longer and, uniquely among vision and language benchmark tasks using real images, the model outputs actions $\langle a_0, a_1, \ldots a_T \rangle$ that manipulate the camera viewpoint.

- 由COCO数据集联想能不能再加一个模态“动作”.

Dataset and Environment

```
{
  "distance": 11.66,
  "scan": "VLzqgDo317F",     // Which building the agent is in
  "path_id": 6250,
  "path": [
    "af3af33b0120469c9a00daa0d0b36799",
    "5be145994f974347850a48cecd04cdcd",
    "79aedad1206b4eea9c4b639ea2182eb7",       // Viewpoint identifier ,观察点编号
    "1c91ed40af2246f2b126dd0f661970df",
    "385019f5d018430fa233d483b253076c",
    "fd263d778b534f798d0e1ae48886e5f3"
  ],
  "heading": 3.751,           // Agent‘s current camera heading in radians  取值范围：［0，2Π）
  "instructions": [
    "Walk down one flight of stairs and stop on the landing. ",
    "Walk between the columns and make a sharp turn right. Walk down the steps and stop on the landing. ",
    "walk forward then turn right at the stairs then go down the stairs. "
  ]
},
```

Instruction: Head upstairs and walk past the piano through an archway directly in front. Turn right when the hallway ends at pictures and table. Wait by the moose antlers hanging on the wall.

- “elevation” : 0,   // Agent‘s current camera elevation in radians (缺省) 取值范围：［-Π/2，Π/2］
- “viewIndex”   // Index of the agent's current viewing angle [0-35] (only valid with discretized viewing angles)
  // [0-11] is looking down, [12-23] is looking at horizon, is [24-35] looking up.

Dataset and Environment

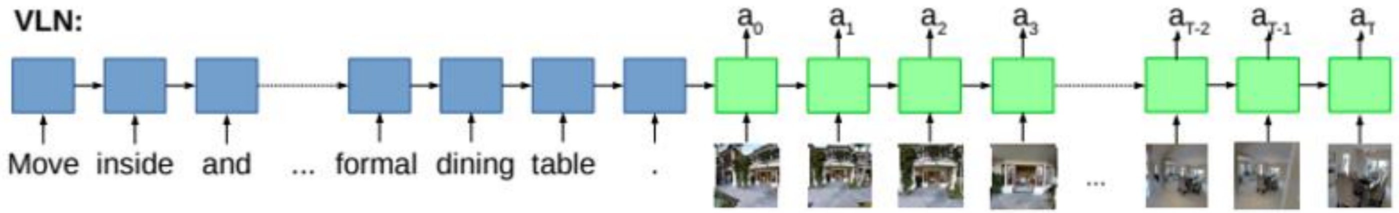
- Dataset Splits: 61 scenes , 14,025 (training) / 1,020 (validation seen); 11 scenes, 2,349 (validation unseen); 18 scenes， 4,173(test unseen). Total: 90 scenes, 21,567 instructions.

- Model action space: left, right, up, down, forward and stop. (The forward action is defined to always move to the reachable viewpoint that is closest to the centre of the agent’s visual field.)

- Image feature: For each image observation, we use a ResNet-152 CNN pretrained on ImageNet to extract a mean-pooled feature vector. 全景图维度：(36, 2048)

- Navigation graphs: Dijkstra for 90 scenes according to the position and connectivity of viewpoints.

Model



- Sequence-to-sequence: Resnet152 + LSTM + attention.

- Language instruction encoding:

$$h_i = \text{LSTM}_{enc}\left(x_i, h_{i-1}\right) \qquad \bar{h} = \{h_1, h_2, \ldots, h_L\}$$

- Image and action embedding:
  - The encoded image and previous action features are then concatenated together to form a single vector q_t.

$$h_t^{'} = \text{LSTM}_{dec}\left(q_t, h_{t-1}^{'}\right)$$

- Action prediction with attention mechanism:

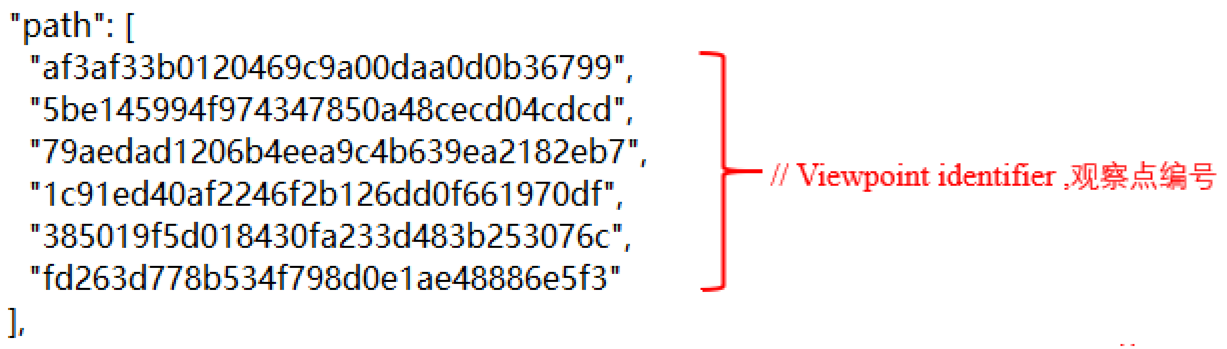
$$c_t = f(h_t^{'}, \bar{h})$$

$$\tilde{h}_t = \tanh\left(W_c[c_t; h_t^{'}]\right).$$

$$a_t = \text{softmax}\left(\tilde{h}_t\right)$$

Training

- Teacher-forcing: Under the ‘teacher-forcing’ approach, at each step during training the ground-truth target action is selected, to be conditioned on for the prediction of later outputs.
  - Goal Teaching: Determine next action on the shortest path to goal, for supervised training.
  - Next Viewpoint Teaching: Determine next action based on the ‘path’ of the given data, for supervised training.

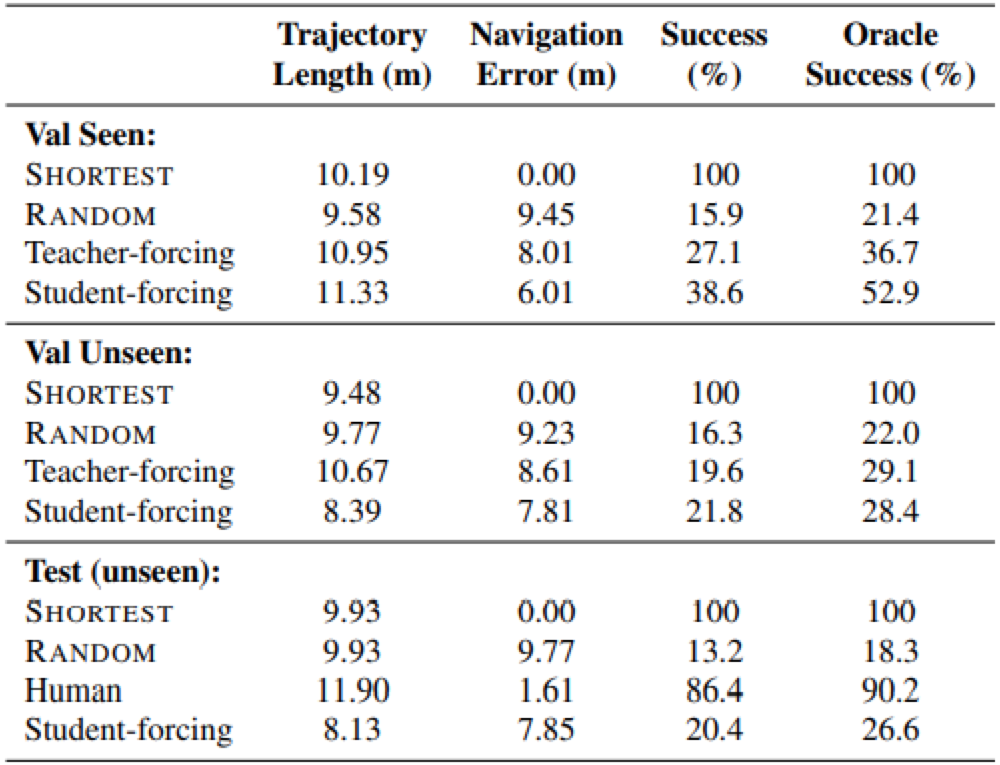


```
"path": [
  "af3af33b0120469c9a00daa0d0b36799",
  "5be145994f974347850a48cecd04cdcd",
  "79aedad1206b4eea9c4b639ea2182eb7",
  "1c91ed40af2246f2b126dd0f661970df",
  "385019f5d018430fa233d483b253076c",
  "fd263d778b534f798d0e1ae48886e5f3"
],
```

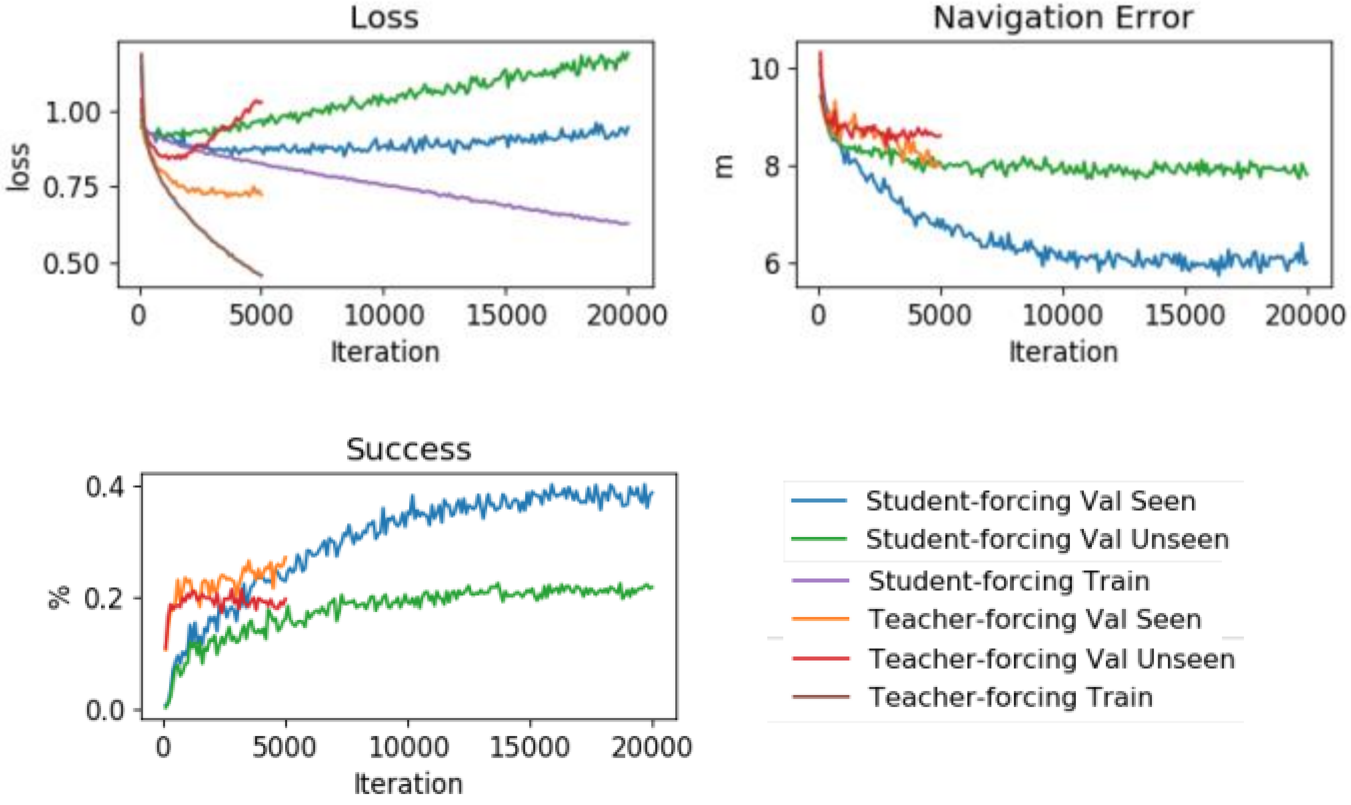
// Viewpoint identifier ,观察点编号

- Student-forcing: The next action is sampled from the agent’s output probability distribution.

Experiments



	Trajectory Length (m)	Navigation Error (m)	Success (%)	Oracle Success (%)
Val Seen:				
SHORTEST	10.19	0.00	100	100
RANDOM	9.58	9.45	15.9	21.4
Teacher-forcing	10.95	8.01	27.1	36.7
Student-forcing	11.33	6.01	38.6	52.9
Val Unseen:				
SHORTEST	9.48	0.00	100	100
RANDOM	9.77	9.23	16.3	22.0
Teacher-forcing	10.67	8.61	19.6	29.1
Student-forcing	8.39	7.81	21.8	28.4
Test (unseen):				
SHORTEST	9.93	0.00	100	100
RANDOM	9.93	9.77	13.2	18.3
Human	11.90	1.61	86.4	90.2
Student-forcing	8.13	7.85	20.4	26.6



Navigation Error: The shortest path distance in the navigation graph between the agent’s final position and the goal location.

Success: The navigation error is less than 3m.

Experiments suggest that neural network approaches can strongly overfit to training environments. This makes generalizing to unseen environments challenging.

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

CVPR 2021

Motivation

- Many visiolinguistic tasks has benefited significantly from the application of vision-and-language (V&L) BERT.

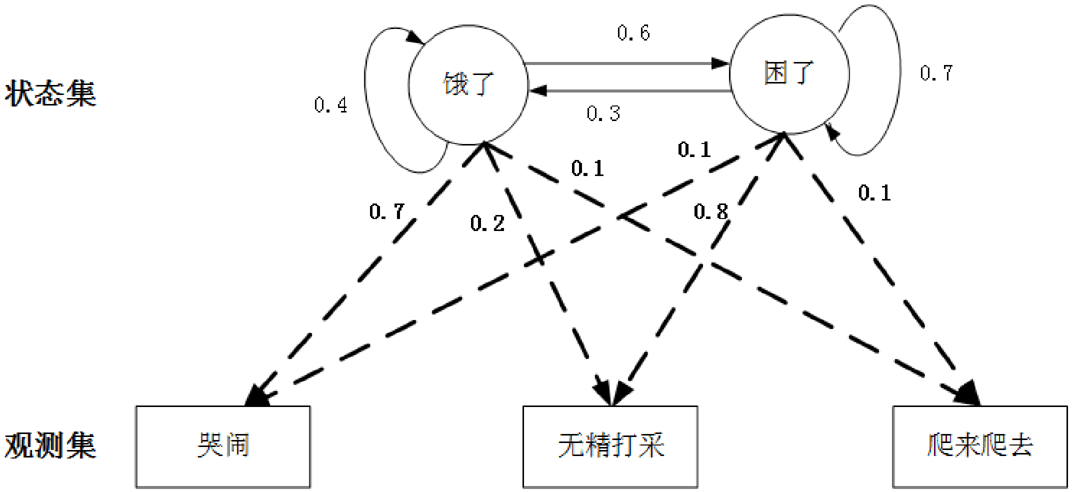
- VLN can be considered as a partially observable Markov decision process, in which future observations are dependent on the agent’s current state and action.

Zhen He 作者

嗯，其实你可以先了解下HMM，POMDP就是HMM和MDP的结合物。

2015-04-09

回复 3



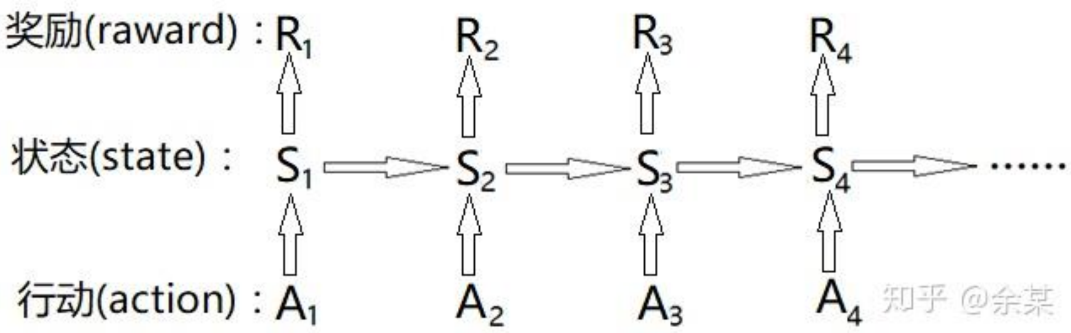
- Real State: $s_t = \langle v_t, \psi_t, \theta_t \rangle$ and navigation history

  3D position $v \in V$

  heading $\psi \in [0, 2\pi)$

  elevation $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$

MDP:


奖励(raward)：R_1 R_2 R_3 R_4
状态(state)：S_1 S_2 S_3 S_4 ......
行动(action)：A_1 A_2 A_3 A_4
知乎 @余某

Motivation

- 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.

- 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. (Applying V&L BERT)

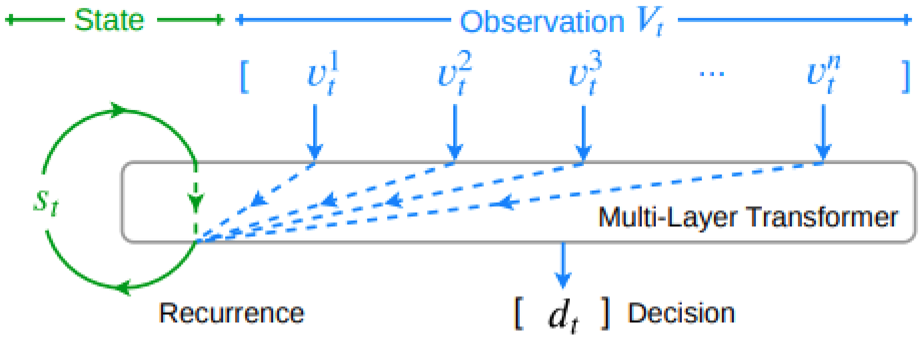
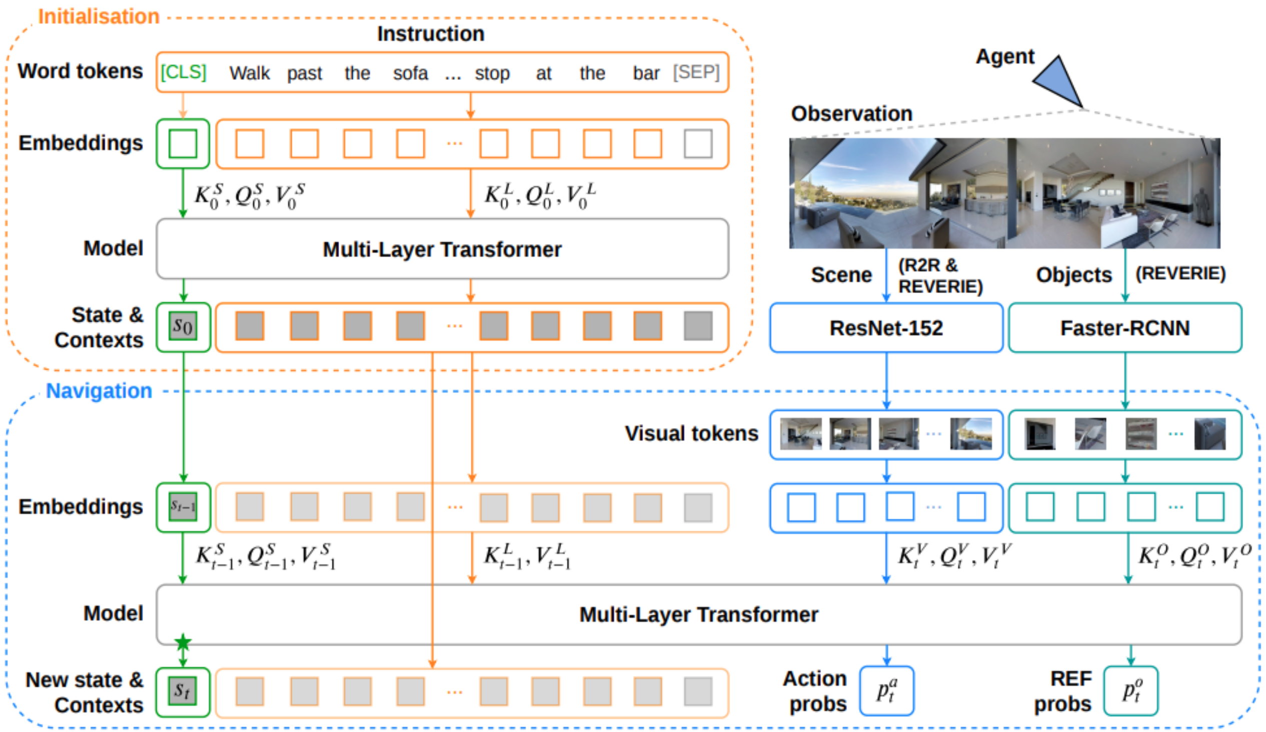


Figure 1. Recurrent multi-layer Transformer for addressing partially observable inputs. A state token is defined along with the input sequence. At each time step, a new state representation $\boldsymbol{s}_t$ will be generated based on the new observation. Meanwhile, the past information will help inferring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

- Using the pre-trained V&L BERT for initialization, combined with a recurrent function to model and leverage the history-dependent state representations. (Applying VLN↻BERT)

- Making the language tokens as keys and values but not queries during navigation, which could reduce the memory usage.

Method



- Language Processing:

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

- Inherite from the pretrained model.
- Define the embedded [CLS] token as the initial state representation s_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.

- Vision Processing:

$$V_t = I_t^v W^{I^v} \quad O_t = I_t^o W^{I^o}$$

  - Project the image features to the same space as the BERT token.

- State Refinement:
  - Attention scores:

$$A_{l,k}^{s,x} = \frac{Q_{l,k}^{s} K_{l,k}^{x\top}}{\sqrt{d_h}} \quad \text{(l:12, k:head, s:state, x:text)}$$

  - State-language attention weights:

$$\widetilde{A}_l^{s,x} = \text{Softmax}(\bar{A}_l^{s,x}) = \text{Softmax}\left(\frac{1}{K}\sum_{k=1}^{K} A_{l,k}^{s,x}\right)$$

  - Weighted raw features:

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

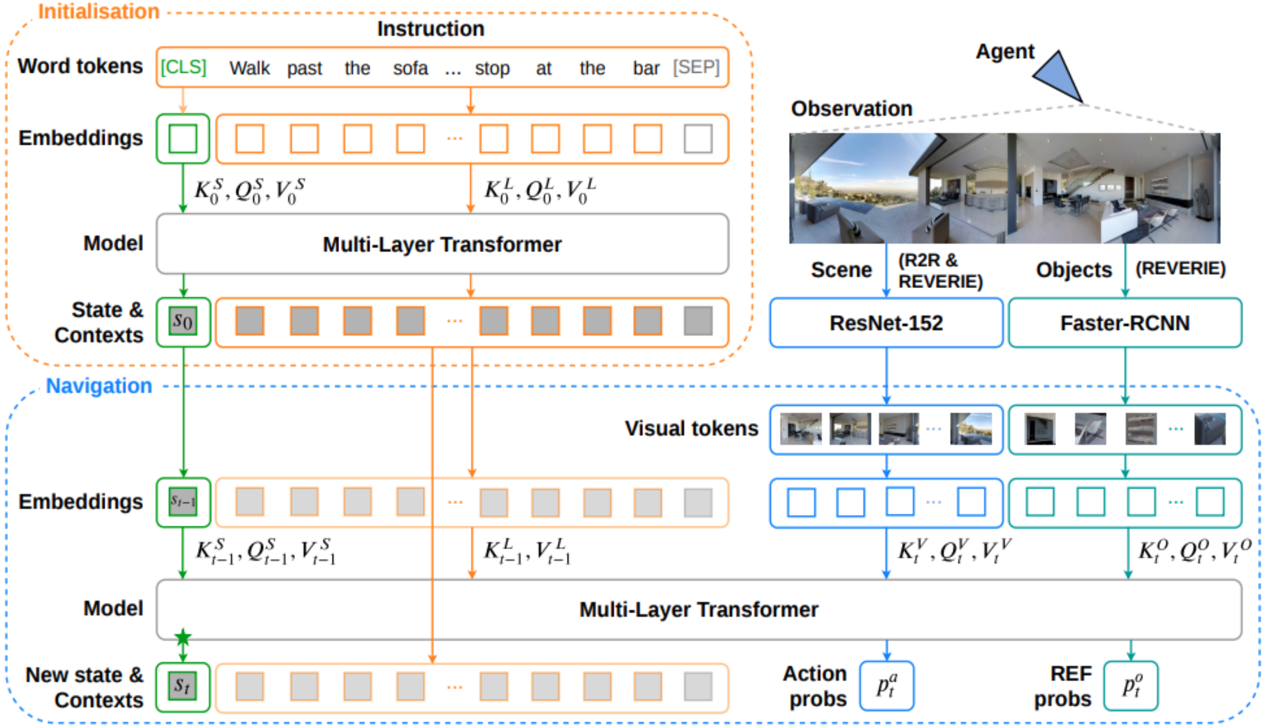
  - Cross-modal matching :

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

  - Feeding action feature:

$$s_t = \left[s_t^f; a_t\right] W^s$$

Method



- Decision Making:

$$p_t^a = \widetilde{\bar{A}}_l^{s,v} \qquad p_t^o = \widetilde{\bar{A}}_l^{s,o}$$

- 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.

- Training :

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

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 [47], we applied an additional cross-entropy term $\sum_t o_t^* \log(p_t^o)$ to learn object grounding.

- Progress Reward :

$$\Delta D_t = D_t - D_{t-1}$$

$$r_t^{D,step} = \begin{cases} +1.0, & \Delta D_t > 0.0 \\ -1.0, & \text{otherwise} \end{cases} \quad (a_t \neq \texttt{stop})$$

$$r_t^{D,final} = \begin{cases} +2.0, & D_t < 3.0 \\ -2.0, & \text{otherwise} \end{cases} \quad (a_t = \texttt{stop})$$

- Path Fidelity Rewards :

$$\Delta P_t = P_t - P_{t-1}$$

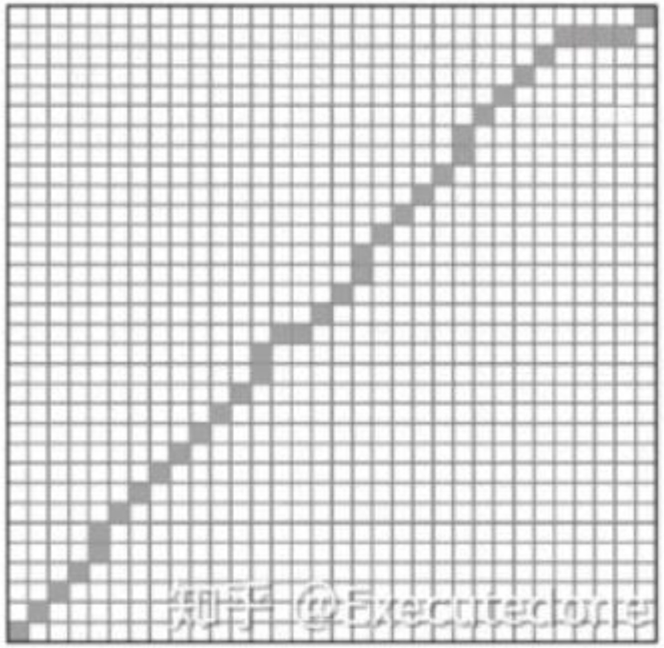
$$r_t^{P,step} = \Delta P_t \quad a_t \neq \texttt{stop}$$

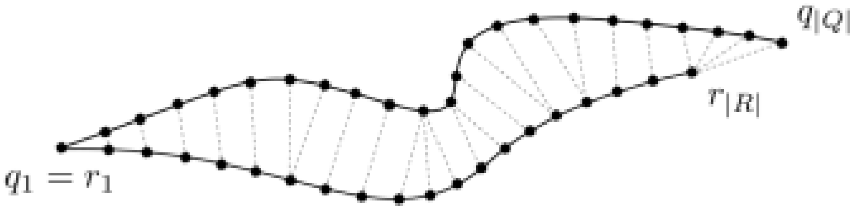
$$r_t^{P,final} = \begin{cases} +2.0 P_t, & D_t < 3.0 \\ +0.0, & \text{otherwise} \end{cases} \quad a_t = \texttt{stop}$$

$$r_t^S = -2.0 \times (1.0 - D_{t-1}) \;\; D_{t-1} \leqslant 1.0 \text{ and } \Delta D_t > 0.0$$

DTW

我们有两个序列C和Q， $C_n=\{c_1,c_2,\cdots,c_n\}$ ， $Q_m=\{q_1,q_2,\cdots,q_m\}$ ，要计算两者之间的距离，先画一个 $m\times n$ 的二维数组，数组中的每个点 $w(i,j)$ 代表 Q_i 与 C_j 的距离 $\sqrt{(Q_i-C_j)^2}$ ，DTW的核心思想是在这样的一个距离矩阵中从两个序列的起点找到通往两个序列终点（即对角线的一端到另一端）的最小距离路径（如图B中灰色方块，可通过动态规划求解，下文有具体例子介绍），但是在寻找路径的过程中，必须满足一些约束条件：





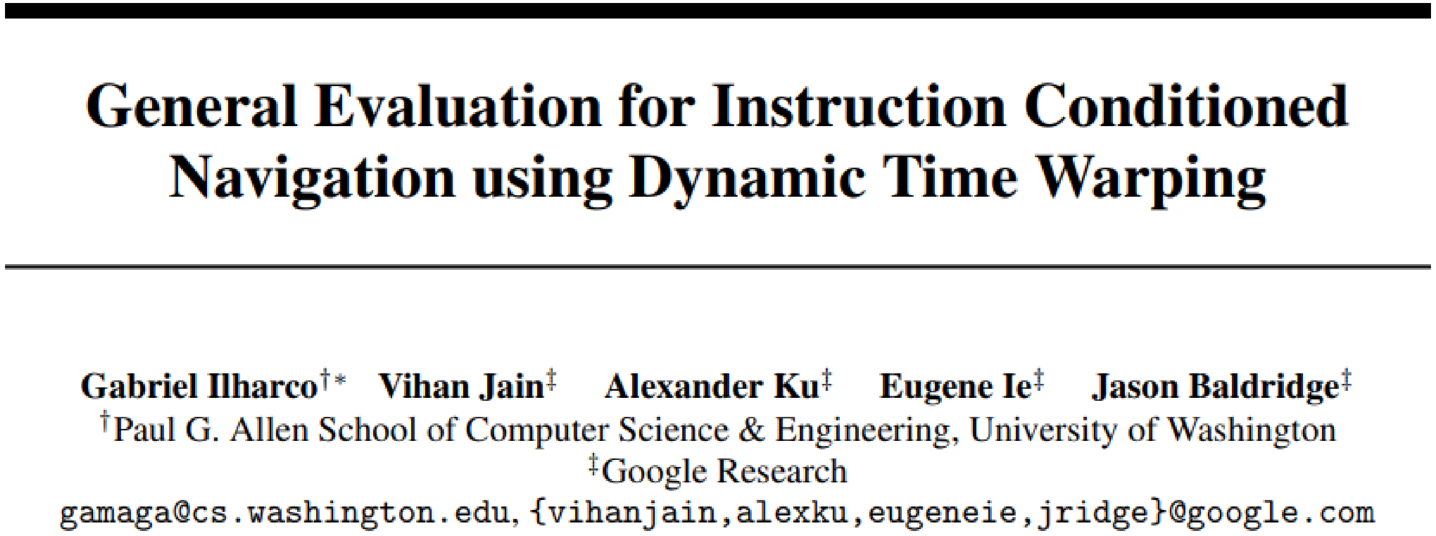
1、边界条件：起点必须是 $w(1,1)$ ，终点必须是 $w(m,n)$ ，要有始有终；

2、连续性：意思是下一个满足条件的灰色方块一定是在当前灰色方块的周围一圈；

3、单调性：下一个满足条件的灰色方块一定在当前灰色方块的右上方，不能回头；

$$dp(i,j)=min(dp(i-1,j-1),dp(i-1,j),dp(i,j-1))+d(i,j)$$

其中 $d(i,j)$ 为 P_i 与 Q_j 的距离.



General Evaluation for Instruction Conditioned Navigation using Dynamic Time Warping

Gabriel Ilharco[†*] **Vihan Jain**[‡] **Alexander Ku**[‡] **Eugene Ie**[‡] **Jason Baldridge**[‡]
[†]Paul G. Allen School of Computer Science & Engineering, University of Washington
[‡]Google Research
`gamaga@cs.washington.edu`, `{vihanjain,alexku,eugeneie,jridge}@google.com`

NIPS 2019

nDTW and SDTW

- normalized Dynamic Time Warping (nDTW):

$$\mathrm{nDTW}(R, Q) = \exp\left(-\frac{\mathrm{DTW}(R, Q)}{|R| \cdot d_{th}}\right) = \exp\left(-\frac{\min\limits_{W \in \mathcal{W}} \sum_{(i_k, j_k) \in W} d(r_{i_k}, q_{j_k})}{|R| \cdot d_{th}}\right)$$

d_{th} is a sampling rate invariant threshold distance

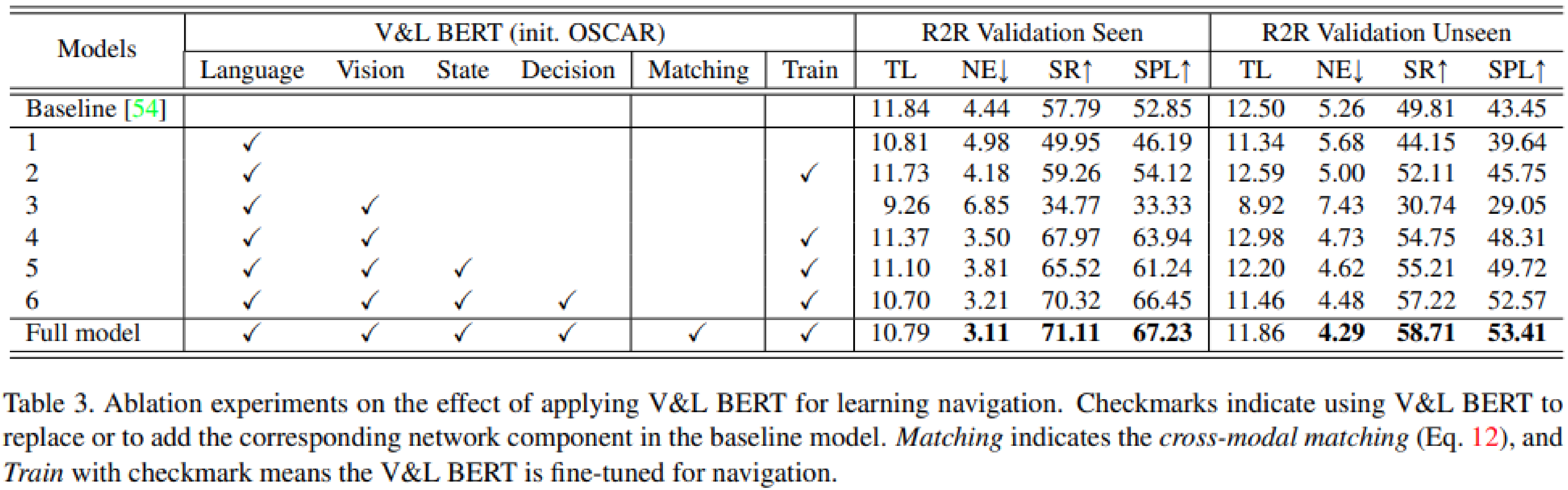
reference $(R = r_{1..|R|})$ query $(Q = q_{1..|Q|})$

- Success weighted by normalized Dynamic Time Warping (SDTW):

$$\mathrm{SDTW}(R, Q) = \mathrm{SR}(R, Q) \cdot \mathrm{nDTW}(R, Q)$$

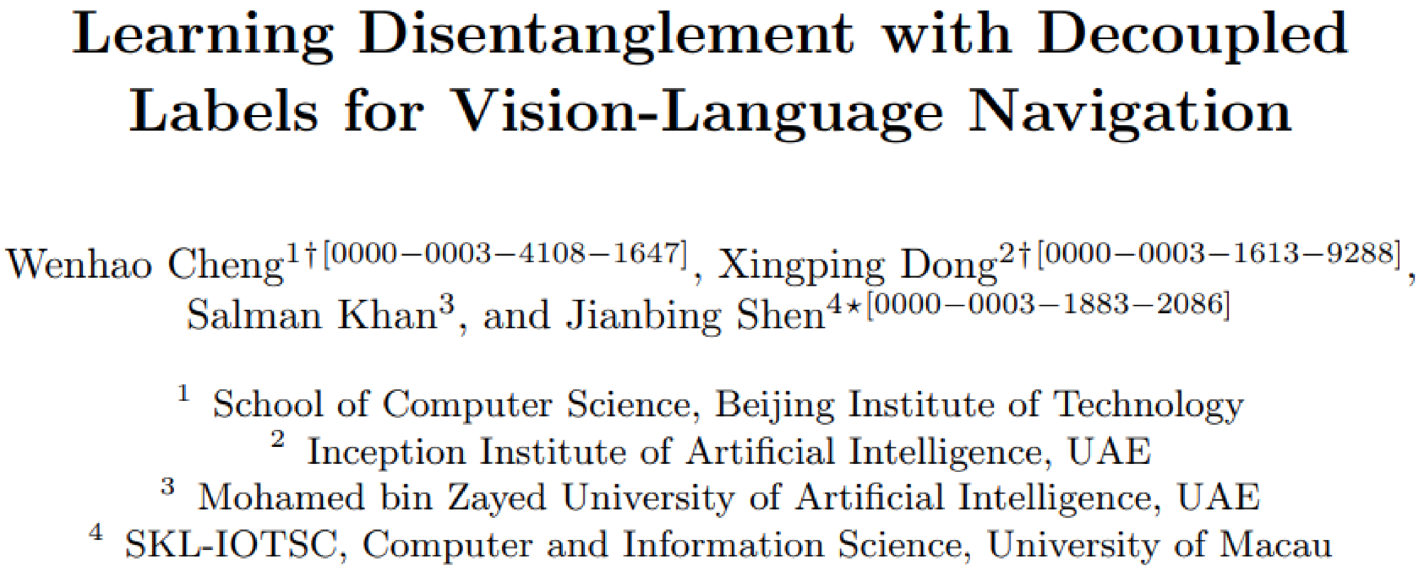
$\mathrm{SR}(R, Q)$ is one if the episode was successful and zero otherwise, commonly defined by the threshold distance d_{th}.

Experiments



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 [54]							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.



Learning Disentanglement with Decoupled Labels for Vision-Language Navigation

Wenhao Cheng[1†][0000−0003−4108−1647], Xingping Dong[2†][0000−0003−1613−9288],
Salman Khan[3], and Jianbing Shen[4⋆][0000−0003−1883−2086]

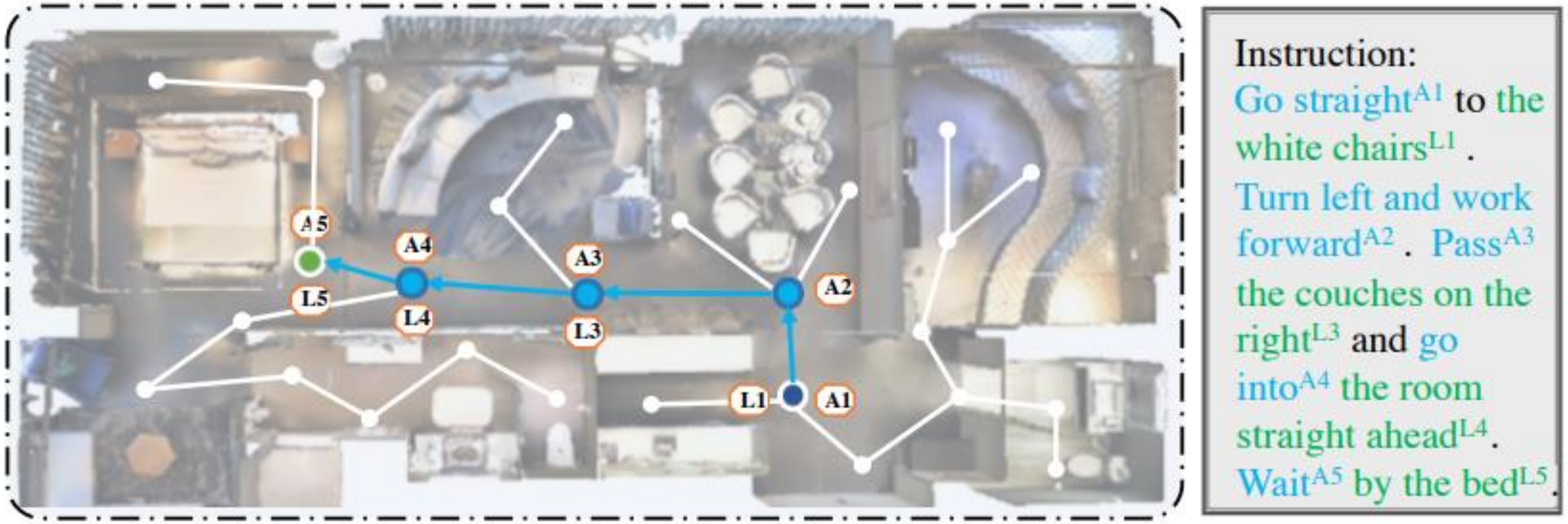
[1] School of Computer Science, Beijing Institute of Technology
[2] Inception Institute of Artificial Intelligence, UAE
[3] Mohamed bin Zayed University of Artificial Intelligence, UAE
[4] SKL-IOTSC, Computer and Information Science, University of Macau

ECCV 2022

Motivation

- The existing VLN dataset Room-to-Room only provides complex human instructions which contain information about several different attributes, e.g., objects, landmarks and actions.
- Such convoluted instructions make the agent's task more challenging.
- Disentangling these instructions can provide more accurate and clear input to improve the decisions taken by the agent.
- Human beings usually do orthogonal decomposition for cognition, i.e., divide something into different attributes to better understand and remember.

Motivation



Instruction:
Go straight^{A1} to the white chairs^{L1} . Turn left and work forward^{A2} . Pass^{A3} the couches on the right^{L3} and go into^{A4} the room straight ahead^{L4}. Wait^{A5} by the bed^{L5}.

Fig. 1. An illustration of decoupled labels providing intermediate supervision during navigation. The superscripts in the instruction denote the landmark and action labels for each viewpoint. The decoupled labels not only contain disentangled information, but help the alignment between vision and language modalities.

Method

- Disentangled Decoding Module

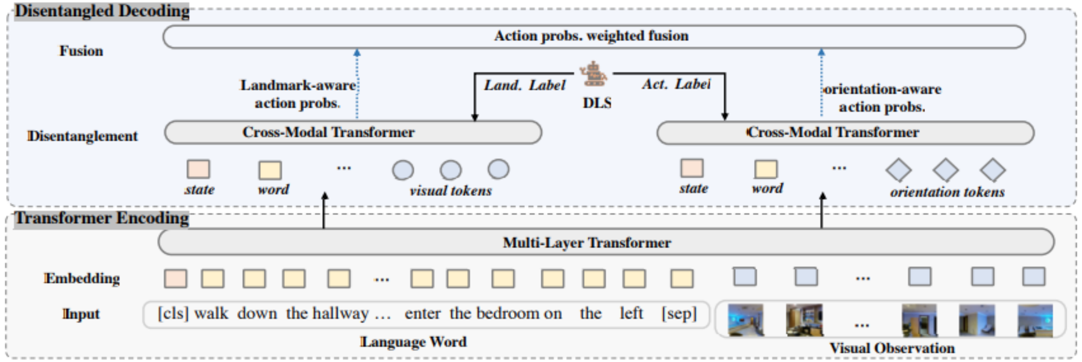


Fig. 3. Overview of the Transformer-based Disentangled Decoding Module. The model takes language words and visual observation as input. After the transformer encoding, two parallel cross-modal transformers are utilized to enable disentangled decoding, supervised by our decoupled labels via a language auxiliary loss. Then the output of two disentangled branches is fused to predict the final action of the agent. DLS represents the decoupled label speaker (see Sec. 3.5 for details).

$$\mathcal{L}_{lan} = -\frac{1}{TL}\sum_{t=1}^{T}\sum_{j=1}^{L} x_{t,j}\log(\gamma_{t,j}) + (1 - x_{t,j})\log(1 - \gamma_{t,j}),$$

$$\mathcal{L}_{act} = -\frac{1}{TL}\sum_{t=1}^{T}\sum_{j=1}^{L} y_{t,j}\log(\sigma_{t,j}) + (1 - y_{t,j})\log(1 - \sigma_{t,j}),$$

- Decoupled Label Speaker

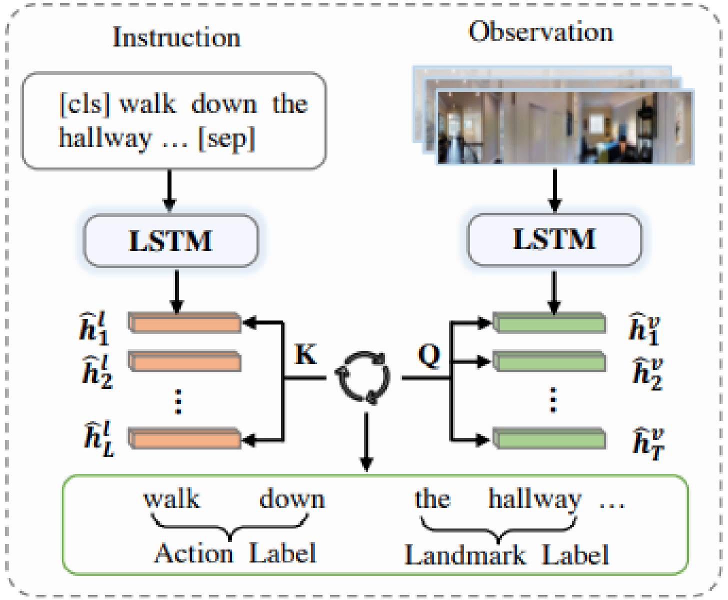


Fig. 4. Architecture of the proposed Decoupled Label Speaker (DLS). Taking the language instruction and visual observation as inputs, the DLS first encodes them using LSTMs and then employs cross-modal attention to predict landmark and action labels for each viewpoint.

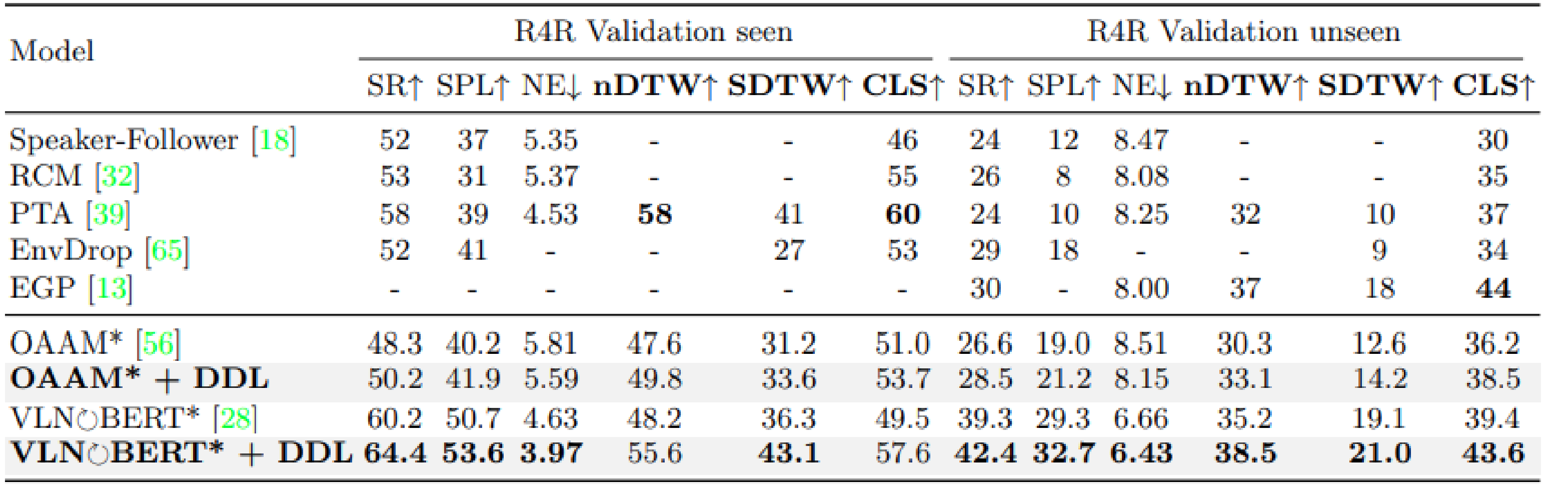
$$\tilde{\gamma}_{t,j} = \text{Sigmoid}\left((W_l \hat{h}_t^v)^T \hat{h}_j^l\right)$$

$$\tilde{\sigma}_{t,j} = \text{Sigmoid}\left((W_a \hat{h}_t^v)^T \hat{h}_j^l\right)$$

- Training

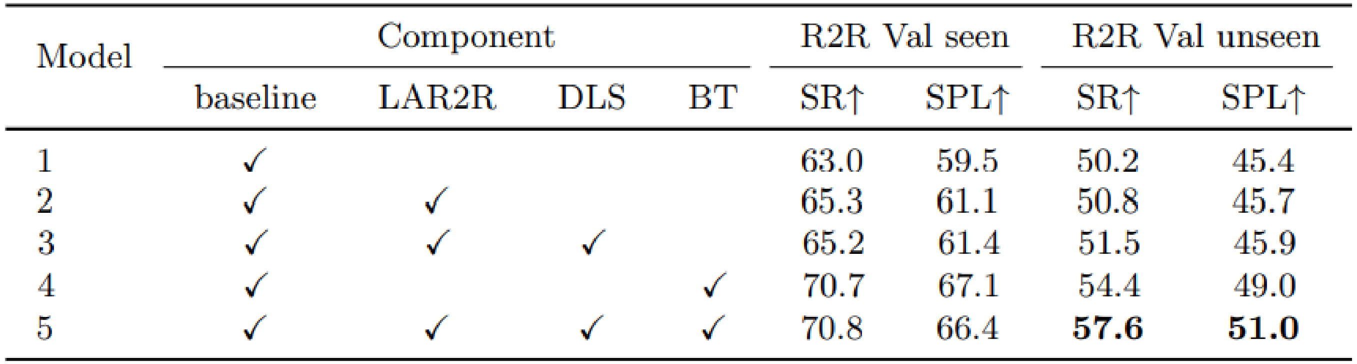
$$\mathcal{L}_{loss} = \mathcal{L}_{RL} + \lambda_1\mathcal{L}_{IL} + \lambda_2\mathcal{L}_{lan} + \lambda_3\mathcal{L}_{act}$$

Experiments



Model	R4R Validation seen						R4R Validation unseen					
	SR↑	SPL↑	NE↓	**nDTW↑**	**SDTW↑**	**CLS↑**	SR↑	SPL↑	NE↓	**nDTW↑**	**SDTW↑**	**CLS↑**
Speaker-Follower [18]	52	37	5.35	-	-	46	24	12	8.47	-	-	30
RCM [32]	53	31	5.37	-	-	55	26	8	8.08	-	-	35
PTA [39]	58	39	4.53	**58**	41	**60**	24	10	8.25	32	10	37
EnvDrop [65]	52	41	-	-	27	53	29	18	-	-	9	34
EGP [13]	-	-	-	-	-	-	30	-	8.00	37	18	**44**
OAAM* [56]	48.3	40.2	5.81	47.6	31.2	51.0	26.6	19.0	8.51	30.3	12.6	36.2
OAAM* + DDL	50.2	41.9	5.59	49.8	33.6	53.7	28.5	21.2	8.15	33.1	14.2	38.5
VLN↻BERT* [28]	60.2	50.7	4.63	48.2	36.3	49.5	39.3	29.3	6.66	35.2	19.1	39.4
VLN↻BERT* + DDL	**64.4**	**53.6**	**3.97**	55.6	**43.1**	57.6	**42.4**	**32.7**	**6.43**	**38.5**	**21.0**	**43.6**

Table 2. Comparison of single-run performance to the state-of-the-art methods on R4R [32]. *denotes our re-implementation. DDL provides consistent improvements.



Model	Component				R2R Val seen		R2R Val unseen	
	baseline	LAR2R	DLS	BT	SR↑	SPL↑	SR↑	SPL↑
1	✓				63.0	59.5	50.2	45.4
2	✓	✓			65.3	61.1	50.8	45.7
3	✓	✓	✓		65.2	61.4	51.5	45.9
4	✓			✓	70.7	67.1	54.4	49.0
5	✓	✓	✓	✓	70.8	66.4	**57.6**	**51.0**

Table 3. Ablation study with OAAM showing the effect of each component on R2R.

Experiments

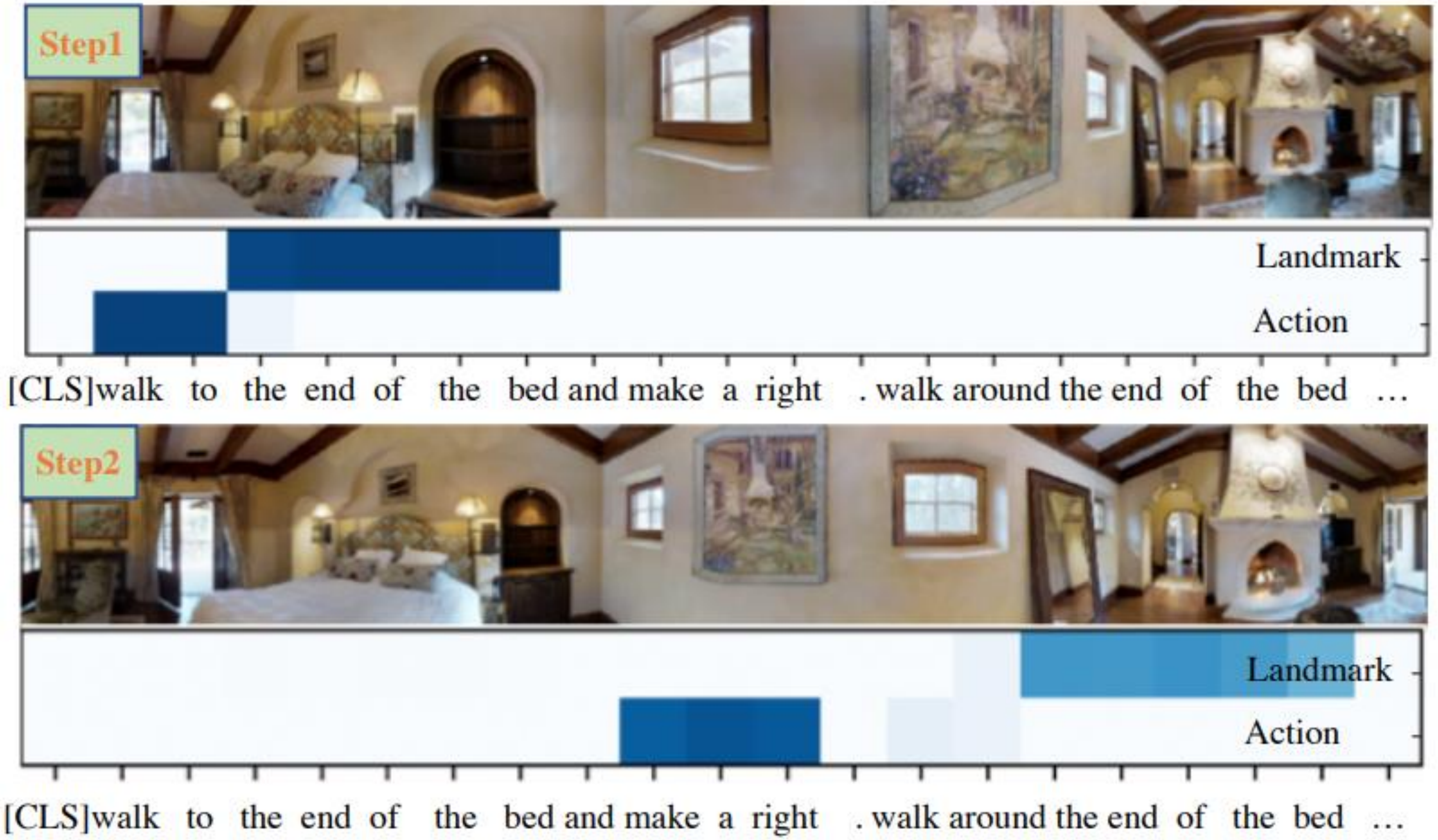
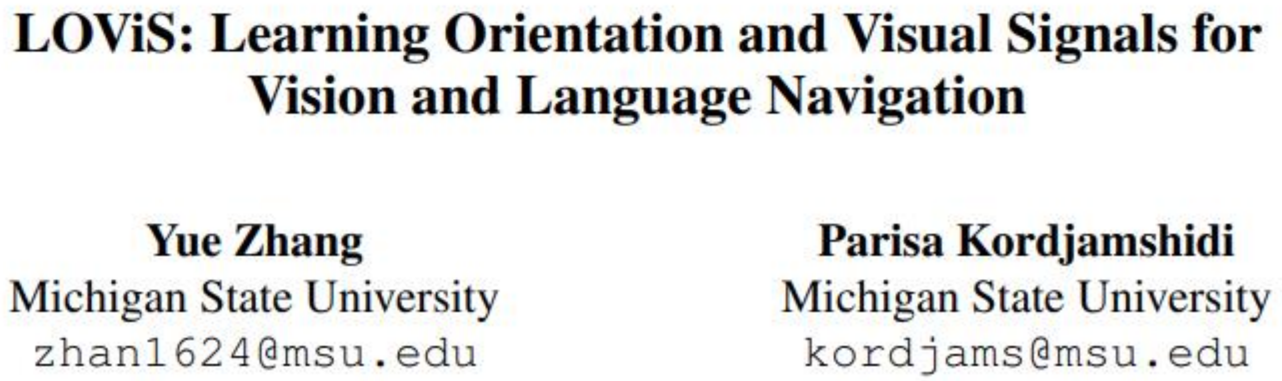


Fig. 5. Distribution of landmark and action attention weights predicted by the decoupled label speaker at the first two navigation steps in an unseen environment. Color shade represents the relative attention weight (darker is higher).



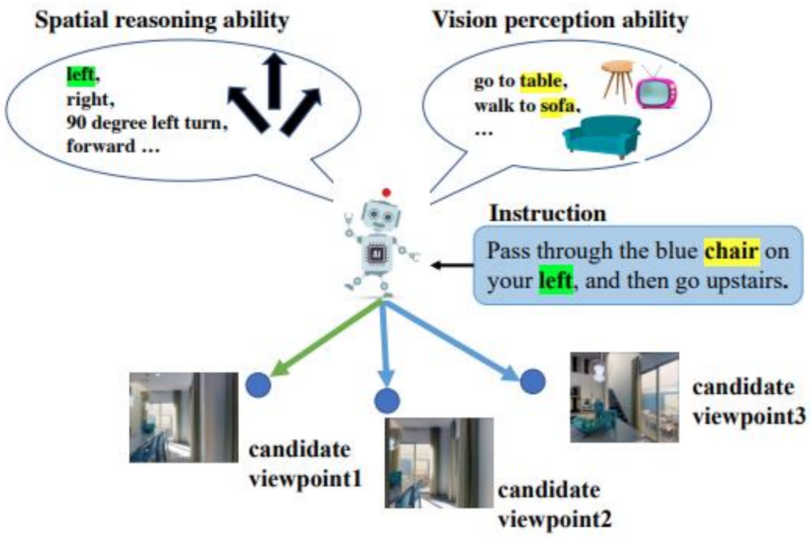
LOViS: Learning Orientation and Visual Signals for Vision and Language Navigation

Yue Zhang
Michigan State University
zhan1624@msu.edu

Parisa Kordjamshidi
Michigan State University
kordjams@msu.edu

COLING 2022

Motivation



- Two major abilities are important to the navigation agent: spatial reasoning and visual perception.

- Spatial reasoning guides the agent towards the correct direction when the instruction is “90-degree left-turn” or “on your 5746 right”, regardless of the surrounding visual scene or objects.

- Visual perception is sufficient to recognize the mentioned landmarks in the visual environment after receiving instructions without any auxiliary signals of orientation.

Model

- History Module

$$\hat{X}, \hat{s}_t, \hat{VO}_t = Cross\_Attn(X, [s_t; VO_t])$$

  - In cross-modality attention Transformer layers, one modality is used as a query and the other as the key to exchange information.

$$s_{t+1}, p_t^h = Self\_Attn([\hat{s}_t; \hat{VO}_t])$$

  - The refinement of the state representation only happens in the history module.

- Orientation Module

$$\hat{X}^o, \hat{s}_t^o, \hat{O}_t = Cross\_Attn(X, [s_t^o; \tilde{O}_t])$$

$$p_t^o = Self\_Attn([\hat{s}_t^o; \hat{O}_t])$$

- Vision Module

$$\hat{X}^v, \hat{s}_t^v, \hat{V}_t = Cross\_Attn(X, [s_t^v; \tilde{V}_t]),$$

$$p_t^v = Self\_Attn([\hat{s}_t^v; \hat{V}_t]),$$

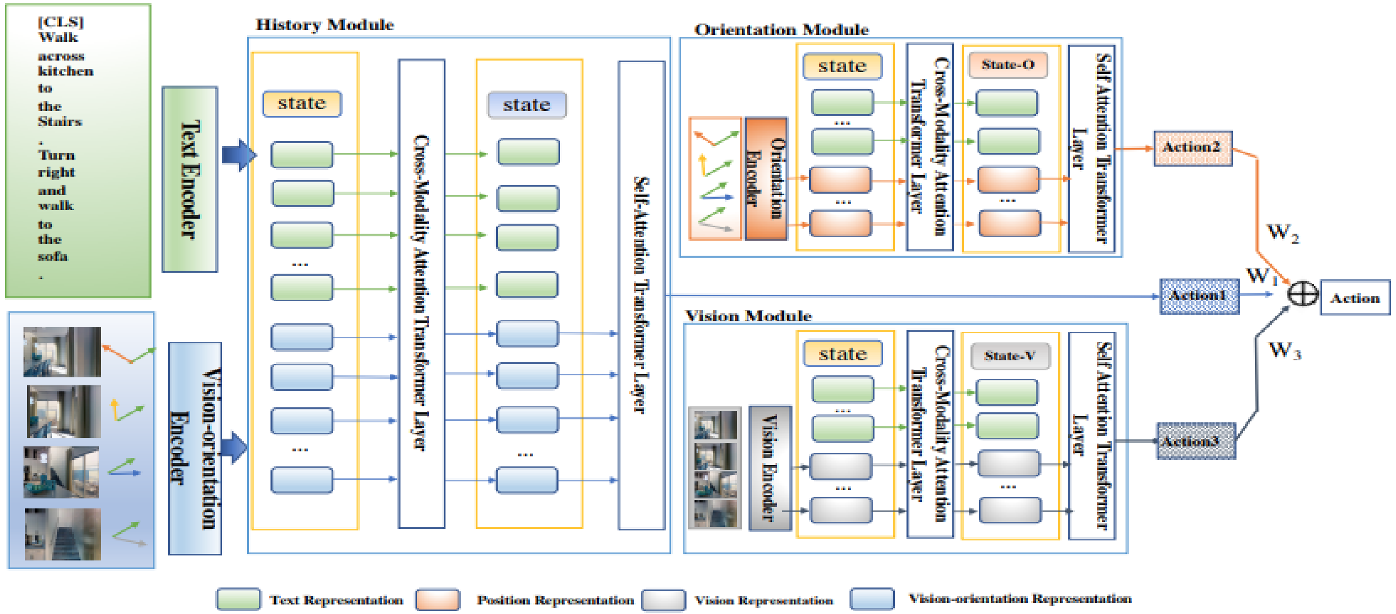

[CLS] Walk across kitchen to the Stairs . Turn right and walk to the sofa .
Text Encoder
Vision-orientation Encoder
History Module
state
Cross-Modality Attention Transformer Layer
state
Self-Attention Transformer Layer
Orientation Module
Orientation Encoder
state
Cross-Modality Attention Transformer Layer
State-O
Self Attention Transformer Layer
Action2
W2
W1
Action1
Action
W3
Vision Module
Vision Encoder
state
Cross-Modality Attention Transformer Layer
State-V
Self Attention Transformer Layer
Action3
Text Representation
Position Representation
Vision Representation
Vision-orientation Representation

Figure 2: Our proposed LOViS contains three modules: history module, orientation module and vision module. Each module can generate action decision based on different reasoning, then three actions are combined to determine the final action selection.

- Action Selection

$$p_t = Softmax(W_a[p_t^h; p_t^o; p_t^v])$$

- Action Selection

$$l = -\sum_t a_t^s log(p_t^a) - \lambda \sum_t a_t^* log(p_t^a)$$

Pre-training

- Masked Language Modeling

$$\mathcal{L}_{MLM} = -\mathbb{E}_{VO_p \text{-} P(\tau),(w,\tau) \text{-} D} \log P(w_m | w_{\backslash m}, VO_p)$$

  - The goal is to recover landmark or orientation tokens by reasoning over the surrounding words, and the orientation and visual observation at the each navigation step.

- Single Step Action Prediction

$$\mathcal{L}_{SSAP} = -\mathbb{E}_{OV_p \text{-} P(\tau),(w,\tau) \text{-} D} \log P(a | w_{[CLS]}, VO_p)$$

- Vision Matching

$$\mathcal{L}_{VM} = -\mathbb{E}_{v_p \text{-} \tau,(w,\tau) \text{-} D}[y \log P + (1 - y) \log P)]$$

- Orientation Matching

$$\mathcal{L}_{OM} = -\mathbb{E}_{o_p \text{-} \tau,(w,\tau) \text{-} D} \log p(O' | w_{[CLS]}, O_p)$$

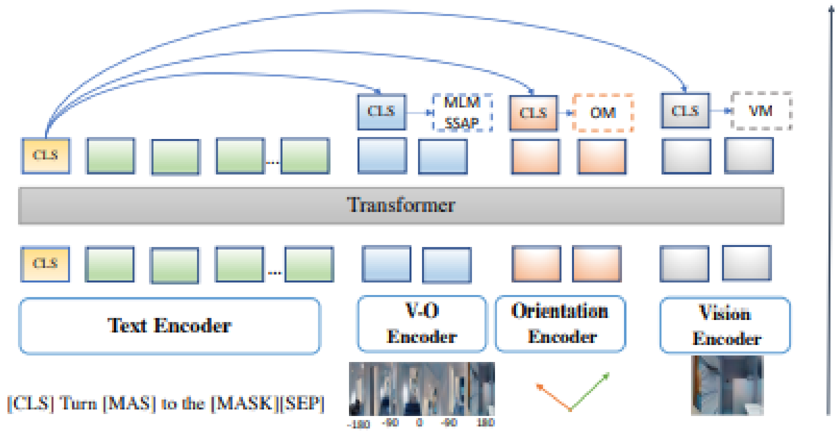
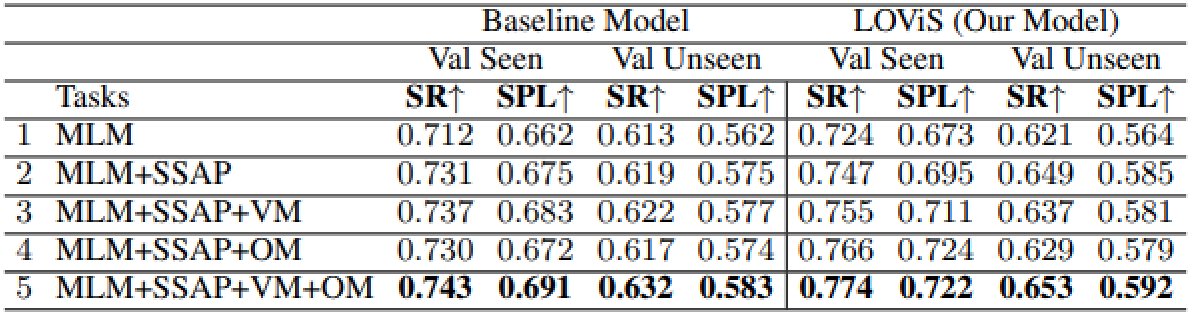


Figure 3: Pre-training Model with Specific Pre-training Tasks. V-O is the encoder considering both orientation and vision representations. MLM: Masked Language Modeling; SSAP: Singe Step Action Prediction; OM: Orientation Matching; VM:Vision Matching.

- Pre-training objective

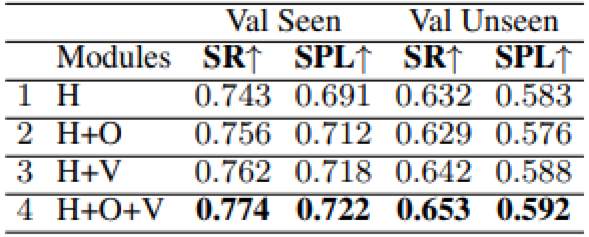
$$\mathcal{L}_{pre-train} = \mathcal{L}_{MLM} + \mathcal{L}_{SSAP} + \mathcal{L}_{VM} + \mathcal{L}_{OM}.$$

Experiments



		Baseline Model				LOViS (Our Model)			
		Val Seen		Val Unseen		Val Seen		Val Unseen	
	Tasks	**SR↑**	**SPL↑**	**SR↑**	**SPL↑**	**SR↑**	**SPL↑**	**SR↑**	**SPL↑**
1	MLM	0.712	0.662	0.613	0.562	0.724	0.673	0.621	0.564
2	MLM+SSAP	0.731	0.675	0.619	0.575	0.747	0.695	0.649	0.585
3	MLM+SSAP+VM	0.737	0.683	0.622	0.577	0.755	0.711	0.637	0.581
4	MLM+SSAP+OM	0.730	0.672	0.617	0.574	0.766	0.724	0.629	0.579
5	MLM+SSAP+VM+OM	**0.743**	**0.691**	**0.632**	**0.583**	**0.774**	**0.722**	**0.653**	**0.592**

Table 3: Ablation Study for Different Tasks of Pre-training on the Baseline and LOViS.



		Val Seen		Val Unseen	
	Modules	**SR↑**	**SPL↑**	**SR↑**	**SPL↑**
1	H	0.743	0.691	0.632	0.583
2	H+O	0.756	0.712	0.629	0.576
3	H+V	0.762	0.718	0.642	0.588
4	H+O+V	**0.774**	**0.722**	**0.653**	**0.592**

Table 4: Ablation Study for Different Modules in Model. H: History Module; O: Orientation Module; V: Vision Module.

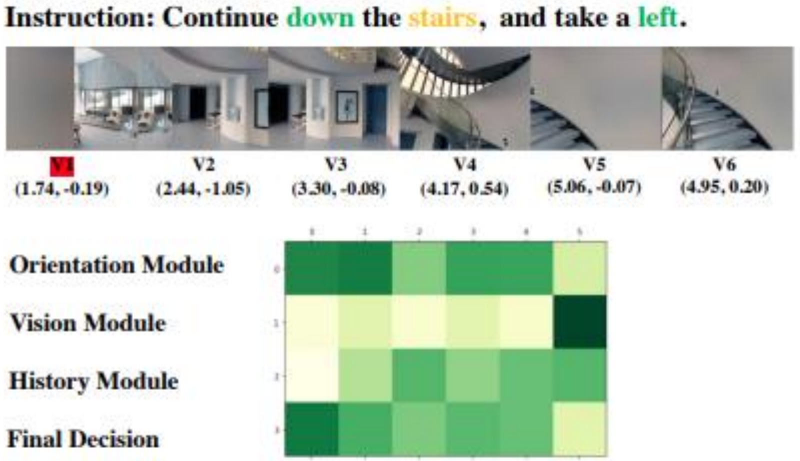


Figure 4: **Qualitative Example.** The ground-truth viewpoint is v1. The word "*down*" and "*left*" are the orientation signals. The word "*stairs*" is the vision signal. The attention map shows the score of different candidate viewpoints in each module. The darker color means the higher score. The numbers below each viewpoint show the orientation information with the format of <relative heading, relative elevation>. The lower value of each number means the orientation is more towards left and down respectively.

Thanks