Noname manuscript No.
(will be inserted by the editor)

Vital Information Matching in Vision-and-language Navigation

xx* [1] · xx [1] · xx*[1] · xx [1] · xx [1] · xx [1] · xx [1]

Received: date / Accepted: date

Abstract With the rapid growth of computer vision and natural language processing technologies, more and more researchers have paid attention on the Vision-and-Language Navigation which is one of the most important tasks in multimodal machine learning. It is crucial to integrate the multimodal intrinsic information. However, the existing model is only achieved through simple data enhancement or expansion, which is obviously far from extracting the internal connections between different modalities. In this paper, we propose a new multimodal matching method and a more flexible navigation self-tuning module to overcome these challenges, which is a brand new neural network model called Vital Information Matching Networks (VIM-Net). Our VIM-Net consists of two matching modules, the vision matching module (V-mat) and the trajectory matching module (T-mat). Specifically, V-mat is to match the target information identified from the vision with the entity information extracted from the instructions. T-mat is to match trajectory features to instructions. Benefiting from the collaborative learning of the V-mat and the T-mat, our VIM-Net has the ability to abstract the key points between different modalities. The performance of the proposed model on various operations is then experimentally demonstrated and compared with other models using the Matterport3D Simulator and room-to-room (R2R) benchmark dataset.

Keywords Vision-and-language Navigation · Multimodal Matching · Self-tuning Module · Vital Information Matching Network

1 Introduction

The Vision-and-Language Navigation(VLN) is to allow the agent to follow the natural language instructions to navigate. This task needs to understand the natural language instructions and the image information that can be seen in the perspective at the same time. Then the agent can ensure its position in the environment and take the corresponding action to reache the destination. In recent years, Anderson [1] first proposed a relatively simple sequence (Seq2Seq) type neural network model in which an action sequence is output from two inputs sequences. Then various data augmentation techniques were used to solve the problem of insufficienTt R2R data set for training the model, including Fried [2], Zhu [3], Jain [4]. Therefore, promoting the solution of this problem and focusing on improving the ability of perceptual vision-and-language inputs multimodal matching has become the focus of research in this field.

Improving the ability to perceive the multimodal matching of the inputs is very helpful for perfecting multimodal intelligence tasks in other fields. For example, visual question answering (VQA) [5], which generates answers to natural language questions on the presented images; image/video

xx*
E-mail: xx

xx
E-mail: xx

xx*
E-mail: xx

[1] Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China

subtitle generation, generates description text on the premise of understanding the inputs image or video content [6], etc. If there is no way to improve the quality of the inputs, and just import it into the overall network without processing, a lot of useful information will be lost. Therefore, improving the multimodal matching ability is of great significance to the optimization of the entire multimodal intelligence task.

It is challenging to improve the multimodal matching ability of perceptual information for two reasons [2,7,8]. First of all, the vital features cannot be accurately predicted of visual and textual information, including landmark objects in the scene or landmark direction words under instructions. Secondly, the internal connection between perceptual inputs may only be reflected in a little vital information for complex indoor scenes and if the connection is not dealt with, this will increase the challenges of the task.

In order to overcome these challenges, we propose a new multimodal matching method and navigation self-tuning module. The former is a new neural network model called Vital Information Matching Network(VIM-Net). Our VIM-Net consists of two matching modules, the vision matching module (V-mat) and the trajectory matching module (T-mat). Specifically, V-mat is to match the target information identified from the vision with the entity information extracted from the instructions. T-mat is to match movements identified from the trajectory with the direction information extracted from the instructions. In the original vision-and-language information, it focuses on the overall feature fusion between the modalities, thereby improving the overall multimodal matching ability is crucial. In view of the fact that the traditional processing method only improves the matching effect by increasing the data quantity, it greatly ignores the characteristics of the dataset itself. Therefore, we follow the way how humans to know their position and take the next step in the process of path finding, which will give priority to matching whether to arrive at the key position, so as to determine whether to complete the navigation at this stage and proceed to the next stage of trajectory planning. More than that, we also sample the path generated by the action predictor as the third type of data information to match the text instructions. In addition, T-mat is aimed at comparing the difference between the target trajectory and the actual path. Then, if the path after the matching score is lower than the threshold, it should be corrected and return to the previous location. The heuristic self-tuning module is applied to retrain and adjust the parameters to plan a more reliable path, thereby improving the accuracy of reaching the designated destination. In general, our contributions include:

(1) A new neural network model VIM-Net is developed, which uses the information after image target recognition to match the extracted features of language entities to optimize the utilization of dataset.
(2) The Self-tuning module based on heuristics makes the generated path match the text instruction, and retrains the parameters at the appropriate position to achieve the effect of optimizing the path and improving the robustness of the entire network.
(3) We have compared the result with baseline on the R2R dataset, and the experimental results show that the VIM-Net network can greatly improve the navigation accuracy of VLN tasks.

2 Motivation

In this section, we first explain that it is universal for the object matching of vision features and language features to be ignored. Then, introduce our preliminary idea. Finally, in response to the above ideas, the measures to be taken are proposed.

2.1 Current Situation

As Figure 1 shows, the perceptual visual information of the VLN task contains a large amount of environmental information [9], such as various tables, chairs, decorations, etc. in the vision. Natural language instructions include requiring the agent to perform high-level actions [10], such as "walk to the lamp and stop" and Various advanced operations including "turning before the green plants". Therefore, the VLN agent is very important for the detailed processing of the two types of information. It needs to acquire knowledge of its location in the current environment to determine the next action; it needs to identify the landmark in the input image mentioned in the instruction; and it needs to choose the low-level action to implements the high-level action in accordance with the instruction.

In particular, the agent should be able to perform alignment and grounding of multimodal input data to understand the natural language instructions coherently in relation to the real-time input images [11]. However, the previous work mainly focused on increasing the amount of

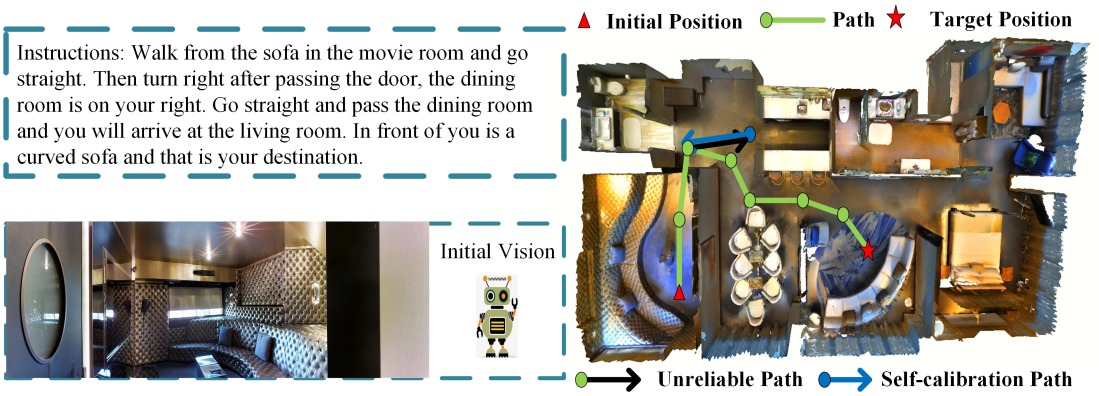


Fig. 1: Examples of vision and language navigation (VLN) tasks.

input data, ignoring the matching between the target information and the Vital Information of the text command under the panoramic vision. This leads to the fact that when the two features are matched, some useless features are taken into account too much, so that when input to the action prediction module, actions that cannot match the instructions are generated [12]. After serialization, the generated trajectory is even more deviated from the original predetermined route, resulting in the poor navigation accuracy.

2.2 Our Discovering

In response to the above-mentioned situation, we found that in the data set, the detailed matching information between perceptual vision and text instructions is often ignored. This phenomenon is not only in the R2R dataset, but also in other related fields, such as dialogue navigation, outdoor scene navigation, etc [13,14,15]. However, this information can be extracted and reasonably used. As shown in Figure 2, we extracted the keywords kitchen, hallway, sitting area, couch of the text instruction and the target features of the visual information have a certain corresponding relationship. The rest of the conjunctions, etc., have no corresponding goals in the visual information, and the most important keywords occupy a relatively small proportion in the entire text, which will not play a leading role in the entire model [16]. If in the model training process, we can train the dominant module as an auxiliary feature of the model, which will help improve the navigation accuracy of the entire VLN task.

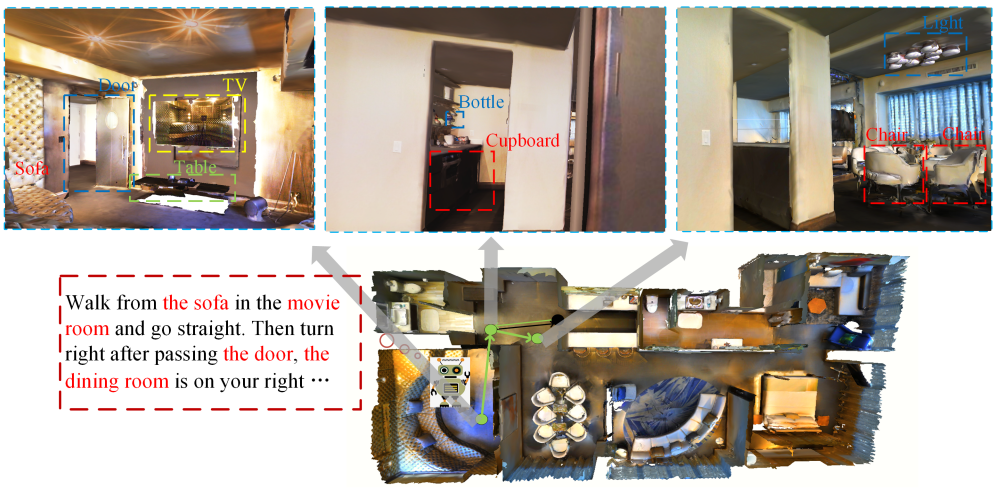


Fig. 2: Examples of the keywords kitchen, hallway, sitting area, couch of the text instruction and the target features of the visual information have a certain corresponding relationship.

2.3 Our Method Brief Introduction

Based on the above description, we found that alignment matching based on the target features of the perceptual vision and the keywords of the text instruction can improve the multimodal matching ability, and can optimize the limitation of simply expanding the amount of data to improve the network effect. This method makes full use of the subtle connections between the two modalities, matches directly from the details, grasps the impact of Vital Information in the entire scene, and greatly optimizes the data input quality of the entire network. Based on this phenomenon, we propose the VIM-Net network, a multimodal data fusion model based on Vital Information matching, to solve the problem that the existing methods cannot make full use of the internal characteristics of the dataset.

3 Model Design

Based on the above motivation, we propose the VIM-Net model, that matches the information after image target recognition with the features extracted from language entities. In this section, we will introduce the details of the VIM-Net network. The VIM-Net contains two modules: visual instruction matching module and trajectory instruction matching module. as the Figure 3 shows. In the visual instruction matching module, we use the Yolo algorithm to extract the object features after target recognition and the word features processed by the entity extraction component of the instruction, compare them and input them into the action prediction module to guide the generation of navigation actions [17].

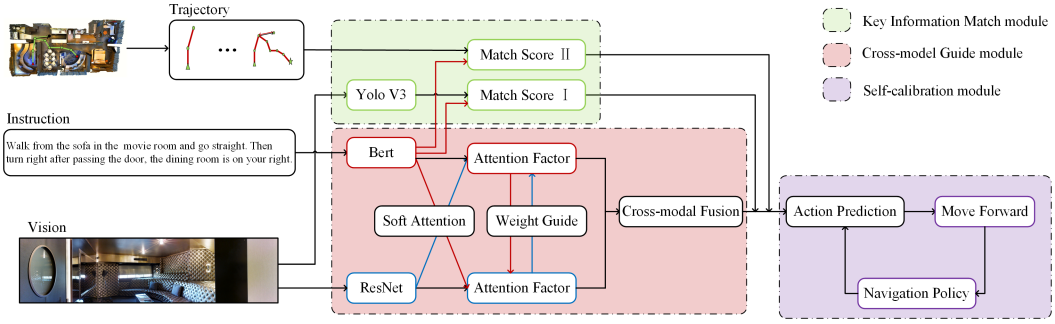


Fig. 3: Schema of the proposed architecture for VLN. The input instruction is vision, instruction and trajectory. The VIM-Net consists of Vital Information match module, Cross-model guide module and Self-tuning module.

In order to further improve the overall integrity of the navigation, we introduced a self-correcting trajectory module that matches the path and the instruction. In this module, we refer to the feedback ideas of Ke [18], improve the relevant evaluation score indicators, and increase the robustness of the entire system.

3.1 Review of Target Detection Component

The target detection algorithm has made a great breakthrough. The more popular algorithms can be divided into two categories, one is the R-CNN algorithm based on region proposal (Fast R-CNN and Faster R-CNN). They are two-stage and need to use heuristic methods or CNN network to generate region proposals, and then perform classification and regression on the region proposals [19, 20]. The other is one-stage algorithms such as Yolo and SSD , which only use a CNN network to directly predict the categories and positions of different targets. The first type of method is more accurate, but slower, but the second type of algorithm is faster, but the accuracy is lower.

as the Figure 4 shows, we use Yolo algorithm, which is detected by a CNN network. It is a single-pipe strategy, and its training and prediction are both end-to-end, so the Yolo algorithm is

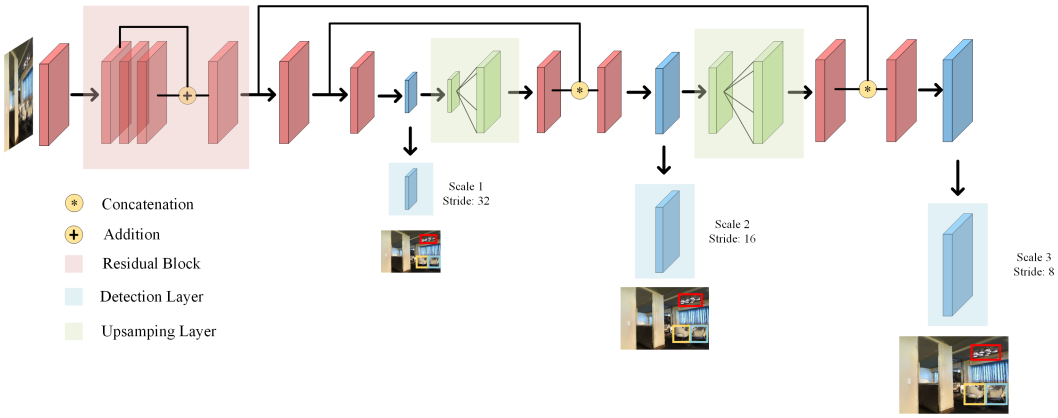


Fig. 4: Schema of the proposed architecture for Yolo.

relatively simple and fast. Then, since Yolo convolves the entire picture, it has a larger field of view in the detection target, and it is not easy to misjudge the background.

3.2 Review of Entity Abstraction Component

In order to further consider the degree of matching between the key target information and the instruction information, we introduced entity abstract components to process the characteristic information under the instruction information. We adopt a similar approach to Suhr [21], replacing phrases in the sentences which refer to previously unseen entities with variables. E.g.,"Walk from kitchen to sofa" turns into "Walk from X1 to Y1". Here X1 and Y1 are the feature points that match the perceived visual information. We use entity abstract components to extract different types of subjects (streets, restaurants, etc.) and number them in the order of occurrence of sentences [22]. as the Figure 5 shows, the number is reset after each instruction is completed, so the model is still a small number of Vital Information is saved, so that the matching efficiency of the overall model is significantly improved [23]. We use an encoder-decoder model with global attention, where the anonymized utterance is encoded using a bidirectional LSTM network.

$$c_i = \sum_{j=1}^{k} \alpha_{i,j} \cdot s_j \tag{1}$$

$$\alpha_{i,j} = \frac{exp(h_i^T F s_j)}{\sum_{j=1}^{k} exp(h_i^T F s_j)} \tag{2}$$

$$h_i, m_i = f(h_{i-1}, m_{i-1}, c_{i-1}) \tag{3}$$

The attention weights $\alpha_{i,j}$ are computed using an inner product between the decoder hidden state for the current timestep h_i, and the hidden representation of the source token s_j. Where F is a linear transformation. The decoder LSTM cell f computes the next hidden state h_i, and cell state m_i based on the previous hidden and cell states,h_{i-1},m_{i-1}, the context vector of the previous timestep, c_{i-1}.

3.3 Vision and Instruction Matching Module

In view of the fact that traditional processing methods only increase the matching effect by increasing the amount, only copy the data set, simply increase the number to improve the accuracy of the matching, or simply complicate the dataset (delete and modify the dataset or add other types of data), these methods still ignore the characteristics of the data set itself. Therefore, our visual instruction matching module is used to determine whether the agent has reached the landmark of the predetermined trajectory, which is the key matching part of the two types of information [24]. The image information obtained by the concrete vision is matched with the entity extracted from the language input. In the original visual and language information, the matching of entity information in the data stream is focused on, thereby improving the overall multimodal matching ability [25].

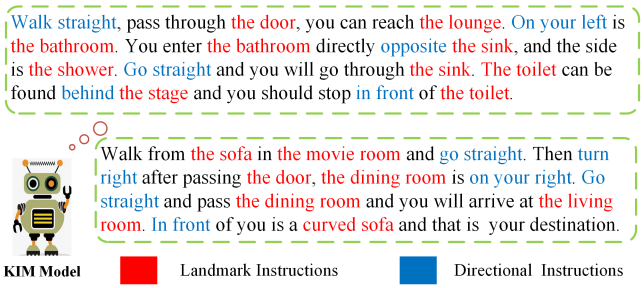


Fig. 5: Examples of instruction for navigation. We use entity abstract components to extract different types of subjects (streets, restaurants, etc.) and number them in the order of occurrence of sentences.

For panoramic images, the orientation features based on the pre-trained Reset-152 and the agent will be input into the visual features of the convolutional neural network (CNN). The institutional embedding dataset for each mode is input as the joint multimodal embedding module so that embedding features can be generated based on intermodal data exchange. Input the perceptual visual features processed by Yolo and the instruction keywords extracted by the entity extraction component to the visual instruction matching module, and set the controller of the scoring module as:

$$\Psi_t(s_t, h_{t-1}) \in (0, 1) \tag{4}$$

where 1 indicates that the aimed landmark is reached and 0 otherwise. Ψ_t is an Adaptive Computation Time (ACT) LSTM which allows the controller to learn to make decisions at variable time steps. h_{t-1} is the hidden state of the controller. In this work, Ψ_t learns to identify the landmarks with the variable number of intermediate navigation steps.

3.3.1 Track and Instruction Matching Module

The inspiration for trajectory instruction matching comes from the fact that when people find their way, they will confirm whether they have not deviated from the trajectory when they reach a landmark or need to turn. Some scholars have made bold attempts before. Gordon and others used external memory MT to clearly remember the traversal path of the agent from the newly visited landmark[24]. When the agent reaches the iconic location, the memory MT is reinitialized to store the traversed path from the most recently visited landmark. This re-initialization can be understood as focusing on the most recently traversed path in order to better locate and better match the relevant direction indication through the trajectory instruction matching module.

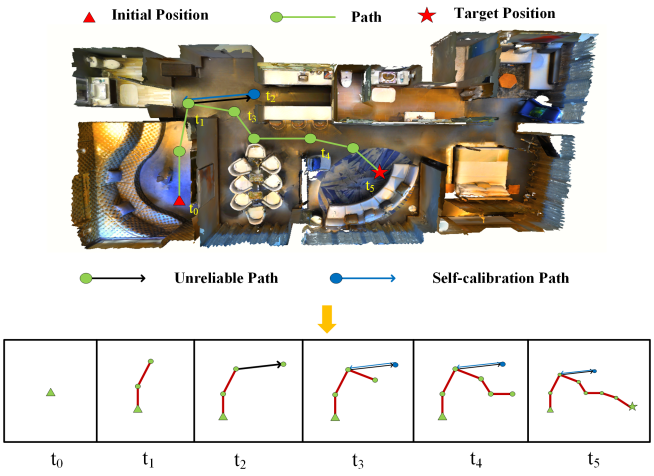


Fig. 6: Examples of navigation trajectory. The path is rasterized and written into the memory image.

As the Figure 6 shows, since the agent is in the navigation process, we set up a write module to write the traversed path into the memory and calculate it from the traversal path in the simulation environment. Our writing module tracks the path from the most recently visited key point to the current location. The path is rasterized and written into the memory image. In the image, the path is represented by the red line, the starting point is marked by the blue square. The write module always writes from the center of the memory image to ensure that there is space in all directions. Whenever the coordinates of a new rasterized pixel exceeds the image size, the module incrementally increases the proportion of the stored image until the new pixel is in the image.

By recording the characteristics of the trajectory and inputting the corresponding instructions to the trajectory instruction matching module, the controller of the scoring module is set as:

$$\varphi_t(m_t, h_{t-1}) \in (0, 1) \tag{5}$$

where 1 indicates that the aimed landmark is reached and 0 otherwise. φ_t is an Adaptive Computation Time (ACT) LSTM which allows the controller to learn to make decisions at variable time steps. h_{t-1} is the hidden state of the controller. In this work, φ_t learns to identify the landmarks with the variable number of intermediate navigation steps.

3.3.2 Action prediction module

The action at at time T is defined as the weighted average of Ψ_t and φ_t . For different parts of the trajectory, the input of the action predictor mainly depends on two inputs $\lambda\Psi_t + \varphi_t$. For example, when the next Vital Information is not visible, the prediction should rely on φ_t ; when the Vital Information is clearly identifiable, both inputs are input to the predictor. After the final data analysis, we determined that when the λ is equal to 0.75, the overall network effect is the best. The learned matching score will adaptively decide which predictions are trustworthy and how many are passed in each step. This adaptive fusion can be understood as a calibration system of two complementary subsystems for motion prediction. The calibration method needs time or situation dependent. In this case, the input of the action predictor is enriched to the greatest extent, more accurate navigation actions are trained, and then the navigation trajectory with less error is serialized.

3.4 Learning Detail

The model is trained in a supervised manner. We follow the student-forcing approach to train our models. In each step, the monitoring signal with motion in the direction of the next landmark trains the motion prediction module. We use cross-entropy loss to train the action module and the matching module because they are planned as classification tasks. The total loss is the sum of the losses of all modules:

$$Loss_{all} = Loss_{vision-matching} + Loss_{track-matching} + Loss_{action} \tag{6}$$

The loss of the two matching modules only takes effect at the landmarks of the landmarks, and these landmarks are more than the road nodes for calculating the motion loss and trajectory error loss. Therefore, we first train the matching networks separately for the matching task, and then integrate them with other components into overall training. We use $Loss_{all}$ to train the entire network.

4 Experiments

4.1 Experimental Settings

Dataset. We use the Room-to-Room (R2R) vision-and-language navigation dataset for our experimental evaluation[1]. In this task, the agent starts at a certain location in an environment and is provided with a human-generated navigation instruction, that describes a path to a goal location. The agent needs to follow the instruction by taking multiple discrete actions (e.g. turning, moving) to navigate to the goal location, and executing a "stop" action to end the episode. Note that differently from some robotic navigation settings, here the agent is not provided with a goal image, but must identify from the textual description and environment whether it has reached the goal. - the Matterport3D navigation graphs, where each path consists of 5 to 7 discrete viewpoints

and the average physical path length is 10m. Each path has three instructions written by humans, giving 21.5k instructions in total, with an average of 29 words per instruction. The dataset is split into training, validation, and test sets. The validation set is split into two parts: seen, where routes are sampled from environments seen during training, and unseen with environments that are not seen during training. All the test set routes belong to new environments unseen in the training and validation sets.

Evaluation metrics. Following previous work on the R2R task, our primary evaluation metrics are navigation error (NE), measuring the average distance between the end-location predicted by the follower agent and the true route's end-location, and success rate (SR), the percentage of predicted end-locations within 3m of the true location. As in previous work, we also report the oracle success rate (OSR), measuring success rate at the closest point to the goal that the follower has visited along the route, allowing the agent to overshoot the goal without being penalized.

Implementation details. We produce visual feature vectors v using the output from the final convolutional layer of a ResNet trained on the ImageNet classification dataset. These visual features are fixed, and the ResNet is not updated during training. To better generalize to novel words in the vocabulary, we also experiment with using BERT to initialize the word-embedding vectors. We generate dynamic filters with 512 channels using a linear layer with dropout (p = 0.5).In our attention module, q and K have 128 channels and we apply a ReLU non-linearity after the linear transformation. For our action selection, we apply dropout with p = 0.5 to the policy hidden state before feeding it to the linear layer.

4.2 Ablation Study

Table 1: The ablation study of our architecture on the R2R validation group and our baseline model is the Speaker-Follower model. When the Vital Information matching module and the trajectory self-tuning module are added, the effect of the whole model is better.

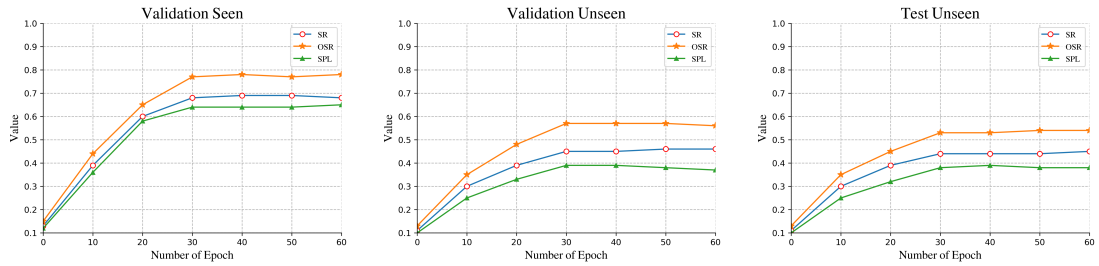
Model	Validation Seen				Validation Unseen			
	SR↑	NE↓	OSR↑	SPL↑	SR↑	NE↓	OSR↑	SPL↑
Baseline	0.63	3.4	0.71	-	0.38	6.68	0.42	-
Cross-model guide	0.68	3.31	0.76	0.59	0.41	5.94	0.53	0.35
Self-tuning	0.64	**3.20**	0.76	0.52	0.42	**5.57**	0.45	0.34
VIM-Net	**0.69**	3.23	**0.78**	**0.64**	**0.46**	5.63	**0.57**	**0.39**

Method	Num	Cross-model Guide	Track Self-tuning	Test Seen			
				SR↑	NE↓	OSR↑	SPL↑
Speaker-Follower				0.36	6.69	0.42	0.28
VIM-Net	1	√		0.43	5.97	0.51	0.34
	2		√	0.41	**5.59**	0.43	0.33
	3	√	√	**0.47**	5.76	**0.53**	**0.39**

The results presented in Table 1 show, we test the impact of our implementation choices on VLN in our ablation study. First, we compare the VIM-Net model with a model that uses a simple replication data set to expand the data set to discuss the impact of the Vital Information matching module on the entire network. Then, we introduced the importance of using the trajectory self-tuning module to correct the entire navigation task.

Cross-model guide module. As the results show, the performance of the Vital Information matching model in data set processing largely exceeds the traditional data enhancement method of VLN. This is because the model can more accurately dig out the inner connection between the perceptual vision and text instructions under the input data set. Compared with the baseline model using purely replicated data sets, our Cross-model guide module improves the success rate by 5.

Track Self-tuning module. Our method is significantly better than the self-monitoring agent using greedy decoding. When the progress marker can use the features of each navigable direction previously accessed, but the trajectory self-tuning module is not available, the performance will not increase significantly (36 SR) . However, the use of the gated attention mechanism is good for extracting the overlap between the position of the landmark under the text instruction and the real trajectory, which means that the network can use this information to improve action selection. Compared with the baseline model that uses pure soft attention to compare the error of the entire trajectory, our method can achieve a moderate gain (41 SR), which reflects the purpose of intelligent navigation.



(a) Validation Seen (b) Validation Unseen (c) Test Unseen

Fig. 7: Comparison of performance on the different condition: (a) Validation Seen, (b) Validation Unseen, and (c) Test Unseen.

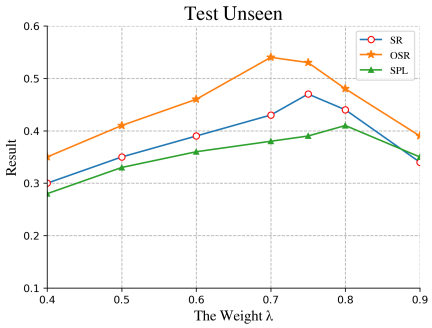


Table 2: Comparison of the Test Unseen Set.

The Weight λ	Test Unseen		
	SR↑	OSR↑	SPL↑
0.4	0.30	0.35	0.28
0.5	0.35	0.41	0.33
0.6	0.39	0.46	0.36
0.7	0.43	**0.54**	0.38
0.75	**0.47**	0.53	0.39
0.8	0.44	0.48	**0.41**
0.9	0.34	0.39	0.35

Fig. 8: Comparison of performance on the different weight of Scoring module. The best result is approximately obtained when the is equal to 0.75.

We compared different results and plotted images to further observe the changes in the results of different training epoch. when λ was equal to 0.75 in the formula $\lambda\Psi_t + \varphi_t$ in the VIM-net model, the effect of the entire model was optimal.

5 Related Work

In order to improve the accuracy of visual language navigation tasks, many scholars [2,8,10,12, 26,27] have made lots of contributions. The Speaker-Follower model is proposed in Fried. The model is mainly divided into two modules: Speaker module and Follower module. Speaker outputs the corresponding language label according to the path, and Follower is responsible for outputting the path according to the input text command, so that by duplicating the original data set The function of expanding the data set is achieved, and the action space of the environment has been changed. The original one can only rotate 90 degrees stiffly to any angle, which increases the freedom of the action space and makes it more accurate decision; Zhu made an integration of all the previous methods, first combining the three-alignment mechanism of seq2seq, reinforcement- learning and imitation learning, and adding the structure of the graph built in the entire scene, and optimizing the specific loss function to adapt to the new algorithm framework. However, the model proposed by the author still ignores the Vital Information of the entire navigation task, that is, typical landmark objects or obvious location words, and the error between the real trajectory and the text instruction, which leads to the effect of the entire model is still not ideal. Therefore, this paper proposes the VIM-Net model. In the network, we propose two main modules, the visual instruction matching module and the trajectory instruction matching module. The detailed information under the navigation task is deeply excavated, and this method fundamentally solves the above-mentioned problems.

6 Conclusion

Conventional navigation focuses on pre-constructing a map of the entire scene, and marking the initial location and destination location. The most suitable trajectory is derived by beam search or greedy algorithms, but the visual language navigation based on deep learning focuses on vision and text The navigation behavior is deduced, in which the visual and text processing are relatively independent, and each is processed by the more mature algorithms in the field, and then simply aligned and spliced, and the appropriate navigation behavior is judged under supervision. This kind of work undoubtedly saves labor costs and time costs, making it more generalizable. This research proposes a new type of deep neural network model VIM-Net as an effective tool to solve VLN tasks. The proposed model aims to use the past temporal context and the multi-modal background extracted with the joint multi-modal embedding module. In addition, VIM-Net, a new greedy local search algorithm with backtracking function, improves the task success rate and search efficiency. Finally, we verify the advantages of the proposed model through various experiments using the R2R benchmark dataset.

Declarations

Funding

Conflicts of Interest/Competing Interests

The authors declare that they have no conflicts of interest, and they have no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of Data and Material

All data generated or analysed during this study are included in this published article and its supplementary information files.

Code Availability

Not applicable.

Authors' Contributions

Ethics Approval

Not applicable.

Consent to Participate

Not applicable.

Consent for Publication

Not applicable.

References

1. P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3674–3683, 2018.

2. D. Fried, R. Hu, V. Cirik, A. Rohrbach, J. Andreas, L.-P. Morency, T. Berg-Kirkpatrick, K. Saenko, D. Klein, and T. Darrell, "Speaker-follower models for vision-and-language navigation," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 3318–3329, 2018.
3. Y. Zhu, F. Zhu, Z. Zhan, B. Lin, J. Jiao, X. Chang, and X. Liang, "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.
4. V. Jain, G. Magalhaes, A. Ku, A. Vaswani, E. Ie, and J. Baldridge, "Stay on the path: Instruction fidelity in vision-and-language navigation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1862–1872, 2019.
5. S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.
6. A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra, "Visual dialog," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 326–335, 2017.
7. F. Landi, L. Baraldi, M. Corsini, and R. Cucchiara, "Embodied vision-and-language navigation with dynamic convolutional filters," in *30th British Machine Vision Conference*, pp. 1–12, 2019.
8. J. Hwang and I. Kim, "Joint multimodal embedding and backtracking search in vision-and-language navigation," *Sensors*, vol. 21, no. 3, p. 1012, 2021.
9. H. Huang, V. Jain, H. Mehta, A. Ku, G. Magalhaes, J. Baldridge, and E. Ie, "Transferable representation learning in vision-and-language navigation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7404–7413, 2019.
10. A. Majumdar, A. Shrivastava, S. Lee, P. Anderson, D. Parikh, and D. Batra, "Improving vision-and-language navigation with image-text pairs from the web," in *European Conference on Computer Vision*, pp. 259–274, Springer, 2020.
11. Y. Qi, Z. Pan, S. Zhang, A. van den Hengel, and Q. Wu, "Object-and-action aware model for visual language navigation," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pp. 303–317, Springer, 2020.
12. C.-Y. Ma, Z. Wu, G. AlRegib, C. Xiong, and Z. Kira, "The regretful agent: Heuristic-aided navigation through progress estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6732–6740, 2019.
13. K. Nguyen and H. Daumé III, "Help, anna! visual navigation with natural multimodal assistance via retrospective curiosity-encouraging imitation learning," *arXiv preprint arXiv:1909.01871*, 2019.
14. A. Yan, X. E. Wang, J. Feng, L. Li, and W. Y. Wang, "Cross-lingual vision-language navigation," *arXiv preprint arXiv:1910.11301*, 2019.
15. H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "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.
16. W. Hao, C. Li, X. Li, L. Carin, and J. Gao, "Towards learning a generic agent for vision-and-language navigation via pre-training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13137–13146, 2020.
17. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.
18. L. Ke, X. Li, Y. Bisk, A. Holtzman, Z. Gan, J. Liu, J. Gao, Y. Choi, and S. Srinivasa, "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.
19. Q.-C. Mao, H.-M. Sun, Y.-B. Liu, and R.-S. Jia, "Mini-yolov3: real-time object detector for embedded applications," *IEEE Access*, vol. 7, pp. 133529–133538, 2019.
20. R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.
21. A. Suhr, S. Iyer, and Y. Artzi, "Learning to map context-dependent sentences to executable formal queries," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2238–2249, 2018.
22. T. Paz-Argaman and R. Tsarfaty, "Run through the streets: A new dataset and baseline models for realistic urban navigation," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6449–6455, 2019.
23. S. Iyer, I. Konstas, A. Cheung, J. Krishnamurthy, and L. Zettlemoyer, "Learning a neural semantic parser from user feedback," in *ACL (1)*, 2017.
24. A. B. Vasudevan, D. Dai, and L. Van Gool, "Talk2nav: Long-range vision-and-language navigation with dual attention and spatial memory," *International Journal of Computer Vision*, vol. 129, no. 1, pp. 246–266, 2021.
25. M. Zhao, P. Anderson, V. Jain, S. Wang, A. Ku, J. Baldridge, and E. Ie, "On the evaluation of vision-and-language navigation instructions," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1302–1316, 2021.
26. X. Wang, Q. Huang, A. Celikyilmaz, J. Gao, D. Shen, Y.-F. Wang, W. Y. Wang, and L. Zhang, "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6622–6631, IEEE Computer Society, 2019.
27. F. Zhu, Y. Zhu, X. Chang, and X. Liang, "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.