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

Key Information Matching and Self-Correction Strategy 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 visual and language navigation which is an important part of multimodal intelligence tasks. Simple data augmentation technology still cannot improve visual and language input and improve the ability of cross-modal matching. In order to overcome these challenges, we propose a new multimodal matching method and navigation self-correction module. The multimodal matching method is a new neural network model KIM-Net (Key Information Matching Network). The main idea is to match the image information obtained by the visual with the entity extracted from the language input. In the original visual and linguistic information, it focuses on the matching of entity information in the data stream, thereby improving the overall cross-modal matching ability. The performance of the proposed model on various operations was then experimentally demonstrated and compared with other models using the Matterport3D Simulator and room-to-room (R2R) benchmark dataset.

Keywords Vision-and-language Navigation · Cross-modal Matching · Self-correction Module · Key Information Matching Network

1 Introduction

The VLN task (Visual and Language Navigation) 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, and then make the status in the environment. The corresponding action finally reaches the target position. In recent years, Anderson (2018) [1]first proposed a relatively simple sequence (Seq2Seq) type neural network model in which an action sequence is output from two input sequences. Then various data augmentation techniques were used to solve the problem of insufficient R2R data set for training the model, including Hao Tan (2019) [2]; Yubo Zhang (2020) [3]; Arjun Majumdar (2020) [4]. Therefore, promoting the solution of this problem and focusing on improving the ability of perceptual visual and language input cross-modal matching has become the focus of research in this field.

Improving the ability to perceive the cross-modal matching of visual and language input is also 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 subtitle generation, generates description text on the premise of understanding the input image or video content [6], etc. If there is no way to improve the quality of the input, and just import it into the overall network without processing, a lot of useful information will be lost.

xx*
E-mail: xx

xx
E-mail: xx

xx*
E-mail: xx

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

Therefore, improving the cross-modal matching ability of perceptual vision and language input is of great significance to the optimization of the entire multimodal intelligence task.

It is challenging to improve the cross-modal matching ability of perceptual visual and language input for two reasons [7,8]. First of all, we cannot accurately predict the key features of visual and textual information, including landmark objects in the scene or landmark direction words under instructions. Secondly, for complex indoor scenes, the inner connection between perceived visual and language input may only be reflected in some key information. If detailed analysis is not targeted, this will increase the challenge of the task.

In order to overcome these challenges, we propose a new multimodal matching method and navigation self-correction module. The multimodal matching method is a new neural network model KIM-Net (Key Information Matching Network). The main idea is to match the image information obtained by the visual with the entity extracted from the language input. In the original visual and linguistic information, it focuses on the matching of entity information in the data stream, thereby improving the overall cross-modal matching ability. In view of the fact that the traditional processing method only improves the matching effect by increasing the number, it greatly ignores the characteristics of the data set itself. Therefore, we follow the example of ordinary people in the process of path finding, which will give priority to matching whether to arrive at a key location, so as to determine whether to complete the navigation at this stage and proceed to the next stage of trajectory planning. Not only 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, the heuristic self-correction module is applied to the path after the matching score is lower than the threshold, it can be self-corrected to the previous stage, and the gated attention module is used 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 KIM-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-calibration 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 KIM-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 alignment of visual 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 Present situation

As Figure 1 shows, the perceptual visual information of the VLN task contains a large amount of environmental information, such as various tables, chairs, decorations, etc. in the vision. Natural language instructions include requiring the agent to perform high-level actions, 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[9]. 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. However, the previous work mainly focused on increasing the amount of input data, ignoring the matching between the target information and the key 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 [10]. After serialization, the generated trajectory is even more deviated from the original predetermined route, resulting in the poor navigation accuracy.

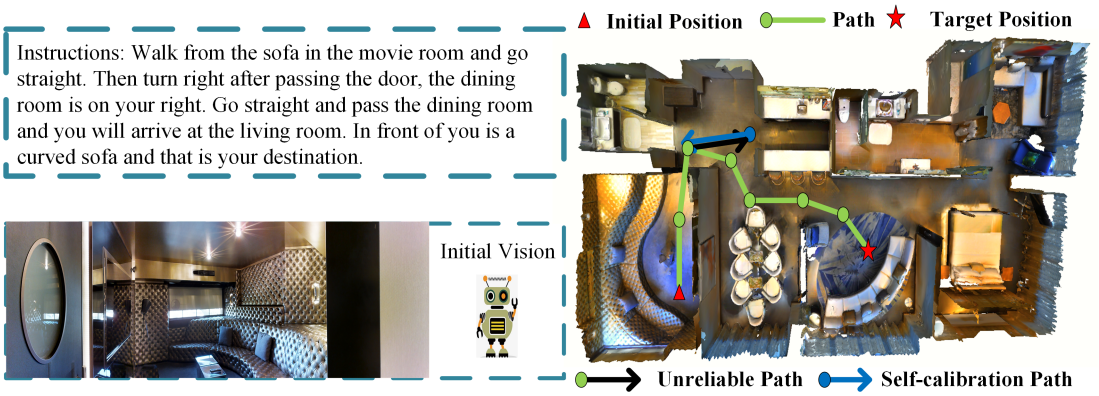


Fig. 1: Examples of vision and language navigation (VLN) tasks.

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.

2.2 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. 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. 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 [11].

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 cross-modal 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 key information in the entire scene, and greatly optimizes the data input quality of the entire network. Based on this phenomenon, we propose the KIM-Net network, a cross-modal data fusion model based on key information matching, to solve the problem that the existing methods cannot make full use of the internal characteristics of the dataset.

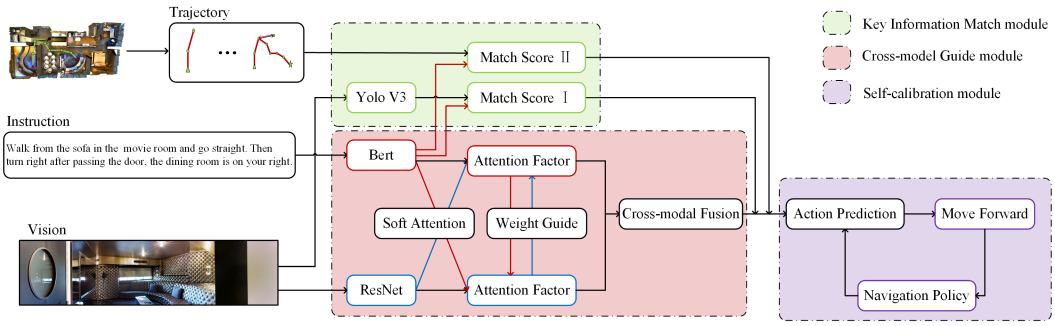


Fig. 3: Schema of the proposed architecture for VLN. The input instruction is vision, instruction and trajectory. The KIM-Net consists of Key information match module, Cross-model guide module and Self-calibration module.

3 Model Design

Based on the above motivation, we propose the KIM-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 KIM-Net network. The KIM-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 [13].

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 Ma(2019) [14], 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) [15]. 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. 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 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 Iyer et al. (2017) [16]; Suhr et al. (2018) [17], 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

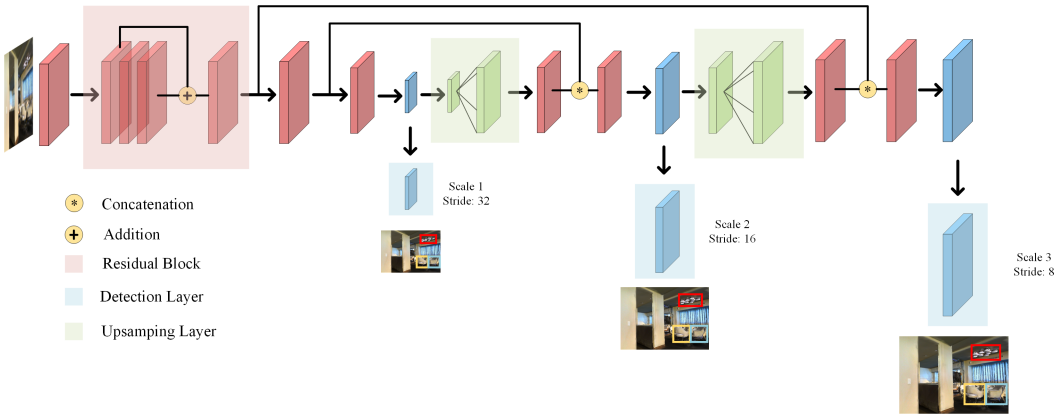


Fig. 4: Schema of the proposed architecture for Yolo.

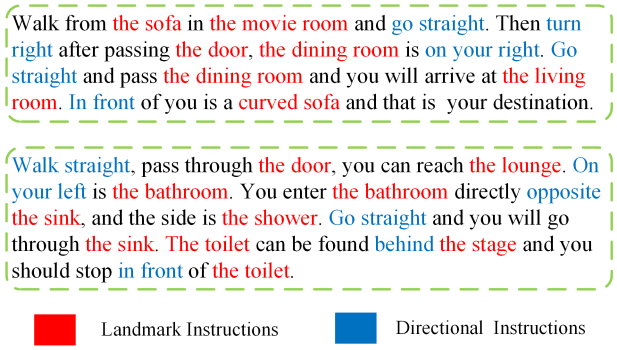


Fig. 5: Examples of instruction for navigation. We use entity abstract components to extract different types of subjects (streets, restaurants, etc.) and number them in the order of occurrence of sentences.

order of occurrence of sentences. as the Figure 5 shows, the number is reset after each instruction is completed, so the model is still a small number of key information is saved, so that the matching efficiency of the overall model is significantly improved. 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. 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 cross-modal matching ability.

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. 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.

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 key information is not visible, the prediction should rely on φ_t ; when the key 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.

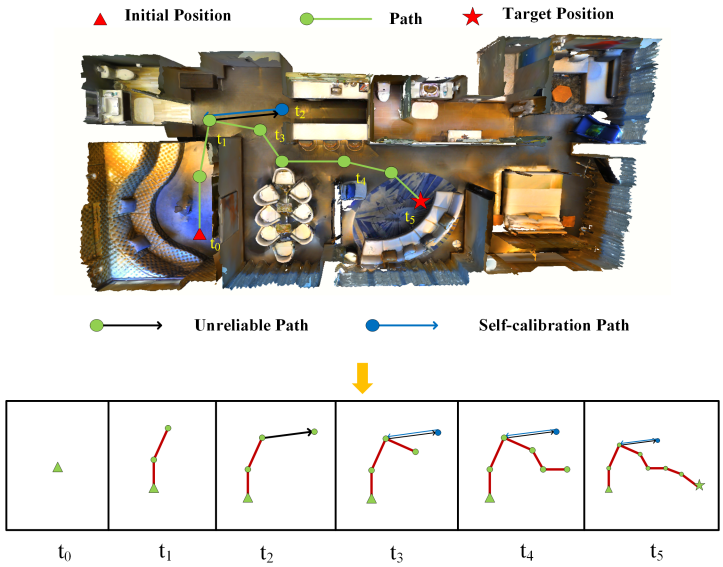


Fig. 6: Examples of navigation trajectory. The path is rasterized and written into the memory image.

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. 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 dataset consists of 7,189 paths sampled from 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

Table 1: The ablation study of our architecture on the R2R validation group and our baseline model is the Speaker-Follower model. When the key information matching module and the trajectory self-correction 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-calibration	0.64	**3.20**	0.76	0.52	0.42	**5.57**	0.45	0.34
KIM-Net	**0.69**	3.23	**0.78**	**0.64**	**0.46**	5.63	**0.57**	**0.39**

Table 2: Comparison of the-test-unseen set.

Model	Validation Seen			
	SR↑	NE↓	OSR↑	SPL↑
Baseline	0.36	6.69	0.42	0.28
Cross-model guide	0.41	5.97	0.51	0.34
Self-calibration	0.40	**5.59**	0.43	0.33
KIM-Net	**0.47**	5.76	**0.53**	**0.39**

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.

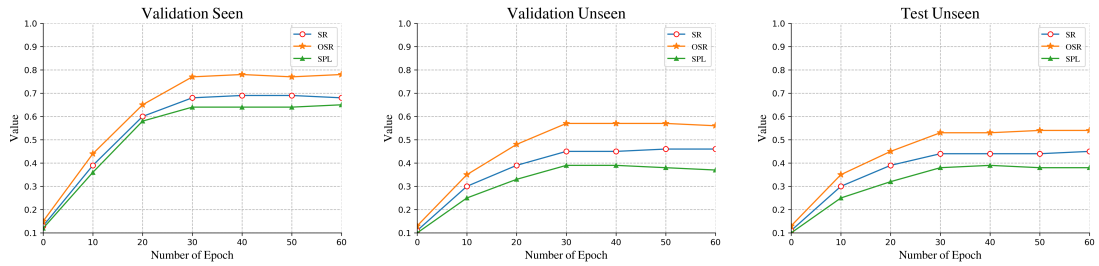
4.2 Ablation study

The results presented in Table 1 and Table 2 show, we test the impact of our implementation choices on VLN in our ablation study. First, we compare the KIM-Net model with a model that uses a simple replication data set to expand the data set to discuss the impact of the key information matching module on the entire network. Then, we introduced the importance of using the trajectory self-correction module to correct the entire navigation task.

Key information matching model. As the results show, the performance of the key 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 method improves the success rate by 4.5

Track self-calibration 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-correction module is not available, the performance will not increase significantly (44 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 (45 SR), which reflects the purpose of intelligent navigation.

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 KIM-net model, the effect of the entire model was optimal.



(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.

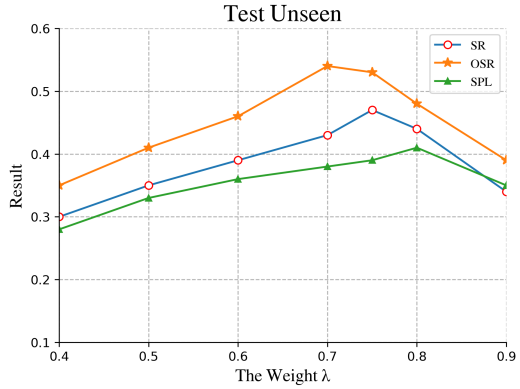


Table 3: Comparison of the Test Unseen Set.

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.

5 Related work

In order to improve the accuracy of visual language navigation tasks, many scholars (Zeng et al. (2018); Shaolei Wang et al. (2018); Giannis Bekoulis et al. (2018); Zhu. (2019);) Contribution. 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 key 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 KIM-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

To sum up, the main research in this paper is human posture transfer, which is significant but at the same time facing many difficulties. Specifically, our model allows two images of different poses (the reference image and the target image) to be reproduced. The existing work is mainly to

estimate the human body structure through a 2D detection point. This method can not represent the human body's personalized posture well and may have the problem of posture missing. In order to make the generated transfer pose more accurate, not the same as the previous methods, we use a SNS network. SNS-1 completes the overall transfer of the reference image pose to the target image pose, and SNS-2 fine-tunes the false reference image generated by SNS-1. Through the SNS network, the reference image input by the network can be constrained by the corresponding ground truth, so that the network training is more stable and the generated image is more accurate. In order to keep the image information from being lost, we propose the parallel Encoder-Decoder EDHRGAN, which can transmit the image information well and generate reasonable images after multiple resolution fusion between different branches. We train on iPER dataset and do extensive experiments. The results show that our method is more effective than other models.

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