Journal of Organizational and End User Computing
Volume 36 • Issue 1

Visual-and-Language Multimodal Fusion for Sweeping Robot Navigation Based on CNN and GRU

Yiping Zhang, Sino-German Institute of Design and Communication, Zhejiang Wanli University, China
Kolja Wilker, DFI College of Communication Art and New Media, Germany*
https://orcid.org/0009-0004-2541-8144

ABSTRACT

Effectively fusing information between the visual and language modalities remains a significant challenge. To achieve deep integration of natural language and visual information, this research introduces a multimodal fusion neural network model, which combines visual information (RGB images and depth maps) with language information (natural language navigation instructions). Firstly, the authors used faster R-CNN and ResNet50 to extract image features and attention mechanism to further extract effective information. Secondly, GRU model is used to extract language features. Finally, another GRU model is used to fuse the visual- language features, and then the history information is retained to give the next action instruction to the robot. Experimental results demonstrate that the proposed method effectively addresses the localization and decision-making challenges for robotic vacuum cleaners.

KEYWORDS

GRU, Multimodal Fusion, ResNet50, Robotic Vacuum Cleaners, Visual-Language Navigation

INTRODUCTION

With the advancement of society and technology, mobile robots have gained the ability to sense, decide, and move, much like humans, by utilizing sensors integrated into their design. The applications of mobile robots have demonstrated their extensive value across various domains, ranging from industrial automation to household services, as new frontiers in robotic technology continue to emerge (Biswal & Mohanty, 2021). In this omnipresent wave of robotics, robotic vacuum cleaners, as members of the mobile robot family, have garnered widespread popularity. These intelligent machines are equipped with autonomous navigation and cleaning capabilities, and users can manipulate the robot verbally to reach the designated place for cleaning, improving people's quality of life and making daily cleaning tasks more convenient. However, with their widespread adoption in real-life scenarios, autonomous navigation requires addressing the link between language and images, which remains a formidable challenge for robotic vacuum cleaners (Li et al., 2023).

DOI: 10.4018/JOEUC.338388
*Corresponding Author

Vision-and-language navigation (VLN) is a fundamental task for achieving general-purpose robots with practical applications in industries such as household robotics and autonomous driving (Anderson et al., 2018). The challenge of this task lies in the unstructured nature of navigation instructions and the complexity of the navigation environment (Wang et al., 2021). Navigation instructions are derived from natural language descriptions, which exhibit diverse forms and intricate expressions. Additionally, real-world navigation environments exhibit considerable variability in their details. Without highly precise maps, the intelligent agent can only access partial environmental information at each moment. Consequently, deducing its current location, making informed decisions, and navigating correctly require the agent to possess robust environmental modeling and reasoning capabilities (Wen et al., 2023). Therefore, researching vision and language navigation methods is essential for robotic vacuum cleaners (Xiao & Fu, 2023).

Various scientific and technological advancements coupled with the emergence of deep learning have made artificial intelligence algorithms one of the most prominent and extensively studied areas in the field of computer science (Ning et al., 2024; Varma & James, 2021; Yang et al., 2021; Zeng et al., 2020; Zhang et al., 2023). These algorithms have achieved significant success in the domains of computer vision and natural language processing (Li et al., 2022; Wang et al., 2018; Zheng et al., 2020). For example, in image feature extraction, Ding et al. (2022) introduced a scheme for reducing the computing power requirements, extracting the multi-scale pixel-wise local features for HSI classification, and extracting the spectral features for superpixels. Verdú et al. (2023) aimed to utilize laser scattering imaging in conjunction with pre-designed convolutional neural networks (CNNs) to simulate the textural variations of pre-processed banana vegetable tissue. Using three different neural network models and four types of red–green–blue (RGB) images, the research revealed observable textural differences as storage time increased, accompanied by a reduction in firmness. It effectively captured and modeled the texture changes resulting from variations in storage and fruit zones. Lai et al. (2022) utilized images captured by an underwater video system equipped with a camera and infrared LED illuminator. They achieved the recognition and measurement of shrimp body lengths in aquaculture pond bottoms by training a YOLOv4-tiny CNN model. These studies all highlight the success of deep learning in image recognition, but there has been limited exploration in the context of natural language processing.

In the field of natural language processing, deep learning has demonstrated significant advantages. Typical models include long-short term memory (LSTM) (Jiang, 2023; Sundermeyer & Ney, 2012; Xie, 2023; Zhang et al., 2021), gated recurrent unit (GRU) (Shen & Wei, 2023; Zhang et al., 2017), Transformer (Swift et al., 2001), and bidirectional encoder representations from transformers (BERT) (Bao et al., 2021; Zhang et al., 2022). Specifically, LSTM is designed to handle and learn from time series data and sequential data, with its primary role in neural networks being to address the issue of long-term dependencies encountered in traditional recurrent neural networks (RNNs). GRU, similar to LSTM, is aimed at mitigating the problem of long-term dependencies in traditional RNNs. However, relative to LSTM, GRU boasts fewer parameters and a simpler computational structure, enabling faster training in specific scenarios, thus making it a popular choice in natural language processing. For instance, to address the issue of traditional methods providing a fixed representation for each word while ignoring variations in word meanings across different contexts, Wang et al. (2022) proposed a training model for contextual sentiment embeddings. This model utilizes a stacked two-layer GRU model as a language model and simultaneously incorporates semantic and sentiment information from labeled corpora and lexicons. As a result, it achieves emotional text recognition. Che et al. (2018) introduced an improved model known as the bidirectional GRU conditional random field (BI-GRU-CRF) based on the GRU neural network to address the typical sequence tagging task of Chinese word segmentation. This model effectively leverages text information and strengthens the correlations between adjacent tags, ultimately generating globally optimal tagging sequences. The above literature collectively demonstrates significant progress in the field of natural language processing using deep learning.

However, integrating deep learning for addressing visual-language navigation requires further exploration of multimodal fusion mechanisms.

Therefore, researchers have begun to explore the use of artificial intelligence to enable robots to receive instructions in the most natural and convenient way, natural language commands, and have the robots autonomously execute tasks as described in the instructions (Li et al., 2023; Sun, 2023). In earlier research at Stanford University, Tellex et al. (2011) constructed a task where a robot had to perform an action such as "placing a tire rack onto a truck" in a simulated environment in 2012. The interpretation of instructions was primarily achieved using hierarchical syntactic trees, and the scenarios were relatively constrained. To facilitate reinforcement learning research, DeepMind (Beattie et al., 2016) established the DM-lab experimental platform, which consisted of 30 challenging tasks. One of these tasks involved finding objects described in natural language, such as "the small red ball in the blue room." DeepMind demonstrated that an agent could successfully complete the task by using only reinforcement learning for autonomous training without the introduction of external or labeled data. This experiment verified agents' ability to autonomously generate semantic understanding through reinforcement learning. In 2017, Anderson et al. (2021) constructed the Room-to-Room dataset for VLN in the Matterport3D simulation environment (Chang et al., 2017). This dataset marked the first real-world scenario for VLN, inspiring numerous subsequent research and dataset creations.

Recently, research on VLN has seen rapid progress, building upon the foundations laid in early studies (Krantz et al., 2020; Landi et al., 2019; Li et al., 2023). Not only have datasets become more comprehensive and diverse, but agents' reasoning and decision-making capabilities have also improved, and the application scenarios have expanded. For instance, Qi et al. (2020) introduced new navigation tasks beyond basic navigation. In these tasks, navigation instructions include a path and a target object. After following the navigation instructions to reach the destination, the agent must identify the target object within the images. Thomason et al. (2020) incorporated dialogue into VLN tasks, allowing agents to ask questions to the environment during navigation, which places higher demands on the agents' ability to understand natural language. Mirowski et al. (2018) extended VLN tasks to outdoor settings, creating the Street Learn simulation environment using Google Maps street view images. This environment comprises approximately 56,000 images from Manhattan and 58,000 images from Pittsburgh. Outdoor environments are characterized by larger maps, richer image details, and increased complexity compared to indoor scenarios. Thus, to address more complex navigation scenarios, further improvement in existing perception and decision models remains necessary (Gu et al., 2022).

To harness the advantages of both visual and language modalities, multimodal techniques have garnered significant attention in the field of VLN. Addressing three key challenges in VLN—aligning progress with navigation instructions for action inference, enhancing interaction between agents and the environment, and improving model generalization—Wang et al. (2019) pioneered the concept of cross-modal matching models. On the other hand, Lu et al. (2020) developed a model that can be applied to various multimodal tasks, including VLN, by leveraging different types of multimodal data. Majumdar et al. (2020) pretrained the VLN-BERT model using internet-sourced image-text data and applied it to VLN tasks. Hao et al. (2020) were the first to undertake pretraining of Transformer models for VLN tasks. They employed a semi-supervised learning approach to train the model on a substantial volume of image-text-action triplets, resulting in a significant performance boost. Collectively, these studies highlight the critical role of multimodal techniques in advancing VLN. However, limited research has been done regarding their application in the context of robotic vacuum cleaners, which has motivated our investigation in this area.

Based on the comprehensive analysis presented above, this paper introduces an innovative multimodal approach tailored for VLN applied to robotic vacuum cleaners, showcasing robust navigational performance. The key contributions of this paper can be outlined as follows:

(1) This paper introduces a methodology within the GRU model to preserve and transmit historical information. Leveraging an RNN and incorporating an approach reminiscent of residual connections, this technique effectively addresses the issue of vanishing memory, thereby enhancing the model's reasoning capabilities.
(2) The paper proposes a multimodal reasoning model to address the challenges associated with cross-modal understanding and matching in visual-language navigation tasks. This model utilizes attention mechanisms to augment feature extraction from RGB and depth images, employing a GRU model for swift, straightforward, and efficient reasoning and decision-making.
(3) Experimental design and validation: The proposed approach undergoes rigorous testing and validation in the Room-to-Room environment. Comparative evaluations against some of the current state-of-the-art algorithms demonstrate the performance of the introduced VLN model. Additionally, we conducted comprehensive ablation experiments to affirm the effectiveness of each module and training algorithm within this approach.

Section 1 introduces the background of multimodal visual-language navigation for robots and summarizes the main contributions of this paper. Section 2 covers the technical principles relevant to the proposed solution. Section 3 provides a detailed discussion of the technical details of the proposed solution. Section 4 outlines the specific experimental settings and environments. Finally, Section 5 offers a summary and outlines directions for future research.

RELATED THEORY

Convolutional Neural Network Model

A CNN is a feedforward neural network that excels at processing grid-like data. Due to their powerful feature extraction capabilities, CNNs have been widely used in various image and video analysis tasks in recent years, such as image classification, facial recognition, object detection, and image segmentation (Zhao et al., 2018). Given input data $X \in R^{M\times N}$ and a filters $W \in R^{U\times V}$, the convolution is performed in the following form

$$y_{ij} = \sum_{u=1}^{U}\sum_{v=1}^{V} w_{uv} x_{i-u+1,j-v+1} \tag{1}$$

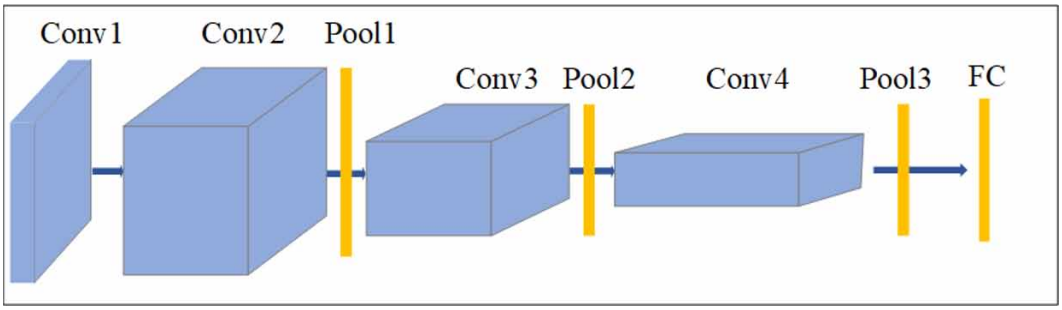
The two-dimensional convolution between input data Y and filter X is defined as:

$$Y = W * X \tag{2}$$

where operation * represents two-dimensional convolution. The working principle of a convolutional kernel is to divide the image into small regions known as receptive fields. This approach of breaking down the image into smaller segments helps the network focus on local features and reduces interference from other regions.

CNNs possess two key characteristics: local connectivity and weight sharing, which provide significant advantages for processing image data. Firstly, the local connectivity property means that neurons in the network are not connected to all image regions but only to local regions. This feature allows CNNs to use input image information more efficiently than fully connected networks, significantly reducing the number of network parameters and computational requirements. Secondly, due to the weight-sharing capability of convolutional operations, filters with the same weights can extract different features from various parts of the image. This efficiency in parameter usage sets CNNs apart from fully connected neural networks, making them more effective in handling image data.

Figure 1. The general composition of CNN



Typical CNNs include multiple convolutional layers to integrate information from individual filter responses within a convolutional layer. When RGB images are used as input data, a series of operations, including convolutional layers, nonlinear activation function, pooling layers, and fully connected layers, are used to extract higher-level semantic information gradually. As shown in Figure 1, in an example of a CNN for image classification, the image first goes through a convolutional layer for feature extraction and is downsampled through a pooling layer. Following the convolutional layers, nonlinear activation functions are typically applied to allow the network to learn and model complex data types. The rectified linear unit (ReLU) is a commonly used activation function after the convolutional layers. Pooling layers effectively reduce the size of feature maps, decreasing the number of convolution operations and, consequently, reducing computational resource consumption and processing time. This combination of convolutional and pooling layers is well-suited for feature extraction in images, and the process is concluded by passing the data through fully connected layers to output the final classification information for the image.

Object Detection

The main task of object detection is to locate the positions of all objects in an image and determine their respective categories. Object detection has a wide range of real-world applications, such as pedestrian detection, hardhat detection, and mask detection. Thanks to the advancement of CNNs and the improvement in GPU computing power, most current object detection methods are based on CNNs.

R-CNN is a two-stage object detection algorithm. It starts by generating a set of bounding boxes that might contain the objects of interest (Xu et al., 2022). These bounding boxes are referred to as region proposals. Subsequently, a series of classification operations are performed on these region proposals. Building upon this, researchers have introduced further advancements, such as Fast R-CNN (Girshick et al., 2015) and Faster R-CNN (Liu et al., 2017). These subsequent models aimed to improve the speed and accuracy of object detection by refining the region proposal and classification process within the two-stage framework. Figure 2 illustrates the basic structure of Faster R-CNN, which consists of four main components:

(1) Feature extraction layer: Faster R-CNN initially uses a pre-trained CNN model (e.g., VGG, ResNet) to extract feature maps from the input image. These feature maps are shared for subsequent region proposals and fully connected layers.
(2) Region proposal networks (RPN): The RPN takes the output from the feature extraction layer as input. It utilizes another CNN on the feature maps and uses a softmax activation function to determine whether each candidate box contains an object. Notably, the candidate boxes generated at this stage are not associated with specific object categories.
(3) Region of interest pooling (RoI pooling): RoI pooling reduces the feature map to a consistent size. Unlike fixed-sized max-pooling, RoI pooling divides the input feature map into

approximately equal regions, typically a fixed number (let us assume k), and applies max-pooling within each region. As a result, regardless of the input image's size, the output of RoI pooling is always of size k.

(4) Classification: After the RoI pooling, the extracted features from each region proposal are fed into a fully connected layer, which is typically followed by a softmax activation function. The softmax function assigns a probability distribution over the different object classes for each region proposal. This provides a classification score for each object category.

GRU Model

RNNs are commonly used for handling sequential data and can maintain short-term memory of previous inputs in a sequence. What sets them apart is their "memory" – that is, their current output is influenced by past input information. RNNs are structured with loops, and neurons not only receive information from themselves but also from other neurons.

However, when dealing with long time sequences, RNNs may face challenges in modeling dependencies between distant time steps. This is because the computation in RNNs often involves multiple matrix multiplications in the Jacobian matrix. This can lead to the vanishing gradient problem or the exploding gradient problem during the training process. In other words, if the influence of past states on the current prediction is not from the very recent past, RNN models might struggle to accurately predict the current state.

Hochreiter and his team introduced LSTM (Yu et al., 2019) to address this issue. As shown in Figure 3, LSTM employs a unique structure consisting of forget gates, memory gates, and output gates. When considering the features of the current input word, they also consider the features of the words that come before it. This clever design allows the network to recognize and retain useful information while discarding irrelevant information when propagating through the network for subsequent word recognition and computation. The mathematical calculation in LSTM can be expressed as

$$i_t = \sigma\left(x_t \cdot W_{xh}^i + h_{t-1} W_{hh}^i + b_h^i\right)$$

$$f_t = \sigma\left(x_t \cdot W_{xh}^f + h_{t-1} W_{hh}^f + b_h^f\right)$$

$$O_t = \sigma\left(x_t \cdot W_{xh}^O + h_{t-1} W_{hh}^O + b_h^O\right)$$

$$\widetilde{c}_t = tanh\left(x_t \cdot W_{xh}^c + h_{t-1} W_{hh}^c + b_h^c\right)$$

Figure 2. The composition of faster R-CNN

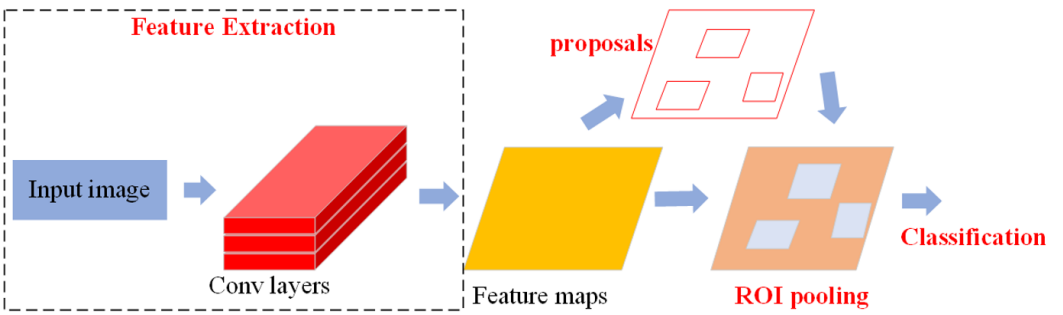
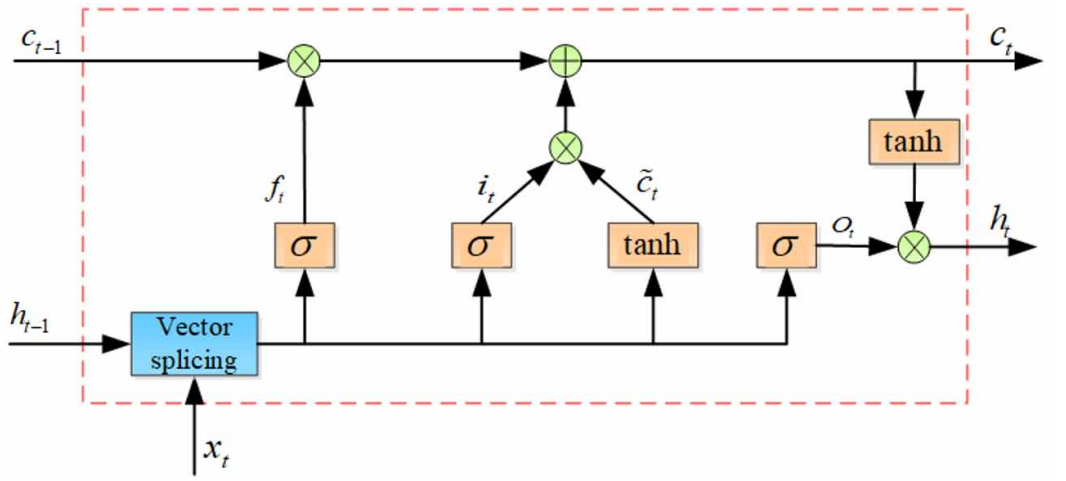


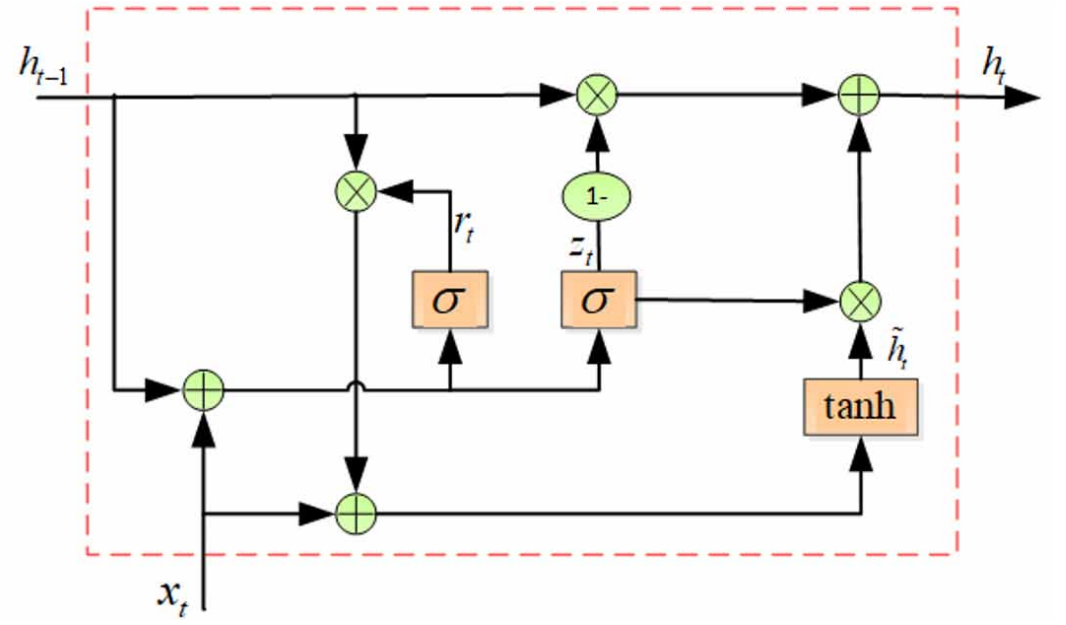
Figure 3. The recurrent unit structure of the LSTM



$$c_t = i_t \otimes \tilde{c}_t + f_t \otimes \mathrm{c}_{t-1}$$
$$h_t = O_t \otimes \tanh\left(c_t\right) \tag{3}$$

Where σ and tanh represent the sigmoid activation function and the hyperbolic tangent activation function, respectively; x_t denotes the input; i_t, f_t, and O_t are the input, output, and forget gates at moment t; W and b represent the weight matrix and bias vector of the gates, respectively; $\tilde{c}_t$ denotes the state transition matrix obtained from the current input; and h_t is the output at moment t.

Figure 4. The recurrent unit structure of the GRU



Due to the complexity of LSTM's structure and the introduction of a new state variable, c_{t-1}, Cho et al. proposed a variant based on LSTM known as the GRU. In summary, while LSTM and GRU are designed to address the vanishing gradient problem in training traditional RNNs, they achieve this with different internal mechanisms. LSTM is more complex with its three gating mechanisms, allowing it to capture long-term dependencies more effectively, but it comes with higher computational overhead. With its simplified structure, GRU is computationally more efficient but might be less effective in capturing very long-term dependencies. The choice between LSTM and GRU often depends on the specific requirements of the task at hand and the available computational resources.

Compared to the LSTM model, GRU involves fewer tensor operations (Liu et al., 2021). Therefore, GRU tends to consume less time during training. However, in practical applications, there is no clear advantage or disadvantage between the two. The choice of which model to use often depends on the specific characteristics of the training data.

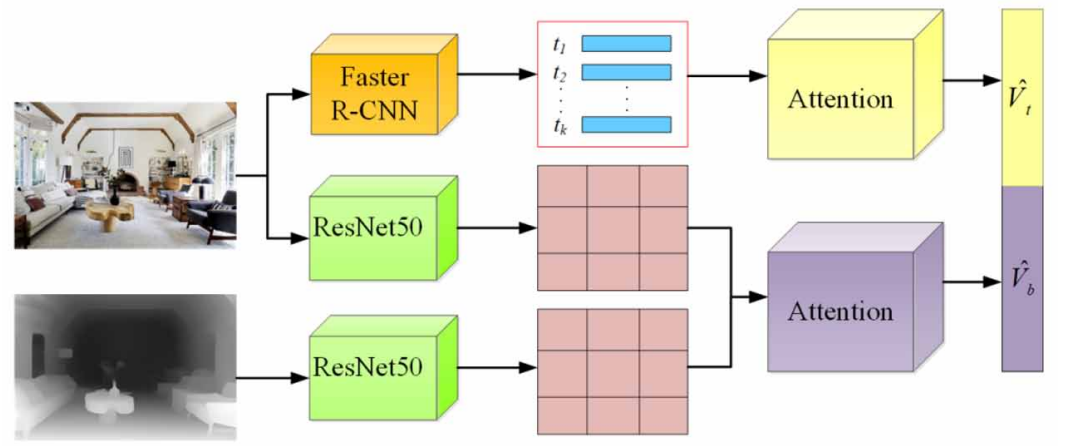
METHOD

The task of visual-language navigation has made significant progress, driven by the collaboration of computer vision and natural language processing. Based on the multi-modal fusion mechanism, it has become popular, providing a better foundation for understanding and executing natural language commands in visual scenes. This section provides a detailed introduction to the visual-language navigation strategy proposed in this paper, which primarily consists of an image feature extraction module, an instruction feature extraction module, and a decision reasoning module.

Image Feature Extraction Module

In previous solutions for visual navigation tasks, a top-down approach was typically employed for image feature extraction. This involved using a CNN to extract global features and then relying on higher-level tasks to determine which regions of the image to focus on and extract specific features. This method processed the image using receptive fields of the same size and shape without considering the specific content of the image. In contrast, the human visual system naturally focuses on the most prominent objects in an image while ignoring the background and unimportant details. This ability enables humans to effortlessly locate objects described in instructions within an image and make correct decisions.

Figure 5. Global attention feature extraction network



Considering the characteristics of indoor navigation environments, which typically feature relatively simple scenes and a limited number of object categories, this paper introduces an instance-driven global attention feature extraction network, an object-induced global attention network. This network, when provided with a visual input image, employs object detection algorithms to extract k feature vectors $v^b = \left[v_1, v_2, \ldots, v_k\right]$, where each feature represents an instance in the image. An attention mechanism is then used to compute weighted values for these image instance features. Simultaneously, the image is processed by a pre-trained ResNet50 network to obtain its global feature, v^a. This global feature is also subjected to an attention mechanism to generate an attention representation for the global feature. Finally, the feature vectors obtained through these two methods are concatenated through parallel operations to produce an augmented image feature vector $\hat{v} = \left[v^b, v^a\right]$. This combined feature vector contains rich information from both local instances and the global context. The structure of the image feature extraction module can be seen in Figure 5.

For feature extraction, this paper employs the Faster R-CNN algorithm. This algorithm is known for its good performance and fast inference speed in object detection. The Faster R-CNN algorithm consists of two steps. The first step is the region proposal network (RPN), which unifies the candidate box generation process of traditional two-step object detection algorithms directly into a CNN, discarding the previous sliding window approach. The RPN network is used to generate candidate boxes, speeding up the candidate box generation process. In the second step, region of interest (ROI) pooling is used to crop each candidate box region to the same size, and then they are grouped into a batch and fed into the final layer of a CNN network. The final model output includes a softmax distribution for all categories and fine-tuning information for each candidate box.

Subsequently, we utilize a Faster R-CNN model trained using the MS-COCO dataset. The MS-COCO dataset contains 80 object categories and provides annotations for 100,000 images in the context of object detection tasks. To validate the dataset's utility, all navigation instructions are tokenized and part-of-speech tagged, and object nouns are extracted. The intersection of these object nouns with the object categories in MS-COCO is taken. After manually removing categories that do not meet the requirements, 16 object categories remain. These 16 categories encompass the majority of objects that frequently appear in navigation instructions.

For the results extracted by Faster R-CNN, this paper selects the top $n = 8$ detection regions based on sorting by confidence probability thresholds. For each selected region i, the feature vector $vb\ i$ after average pooling is taken from it, where $v_i^b \in R^{2048}$. Then, attention is computed between the instruction feature I and the feature vector of the instance to obtain the final feature representation of the image instance part, denoted as

$$\widehat{V_b} = Att\left(I, v_i^{b_n}, v_i^{b_n}\right) \tag{4}$$

In this task, while objects in the image contribute to the decision-making process for navigation, the overall image is essential for making correct decisions. This is because the features extracted from object detection are limited, and relying solely on these features would omit a significant amount of crucial information. Additionally, information in the instructions, such as "hallway" or "kitchen," requires global image features to be perceived. Therefore, the global features of the image are retained. Initially, the RGB image and depth map are input into a pre-trained CNN to extract features V_r and V_t. These features are then used in conjunction with the text instruction feature I through an attention module to extract global image features $\hat{V}_r$ and $\hat{V}_d$. Through these steps, both bottom-up and top-down image features are obtained. They are concatenated to create the final image feature $\hat{V} = \left[\widehat{V_b}, \widehat{V_r}, \widehat{V_d}\right]$, which incorporates both global and local information.

Instruction Feature Extraction Module

As shown in Figure 6, the robot encodes language sequences using an RNN approach. This method encodes natural language instructions of varying lengths into fixed-dimensional semantic vectors, denoted as vector *S*, with a dimension of 128. The natural language instruction is a sequence of words, and this sequence can contain multiple different actions.

The first step is to perform word embedding for each word in the instruction. Word embedding converts each word in the instruction into a vector or matrix format for subsequent calculations and training. After embedding, each word is represented as a vector, denoted as $x_i, i = 1,2,3,\cdots n$, and is sequentially input into a GRU in chronological order along with the previously hidden state, h_{t-1}. (When t = 1, corresponding to a vector of length 128 with all elements set to 0).

This process encodes the language sequence. At each time step, a hidden state h_t is output, and the transformation of the RNN hidden layer is represented by the function *f* as follows

$$h_i = f\left(x_i, h_{i-1}\right) \tag{5}$$

Assuming there are *n* words in the natural language instruction, the hidden state h_n, the output of the last word, can be considered the semantic vector *S* for the entire instruction. This semantic vector can effectively represent the semantic information of the navigation instruction.

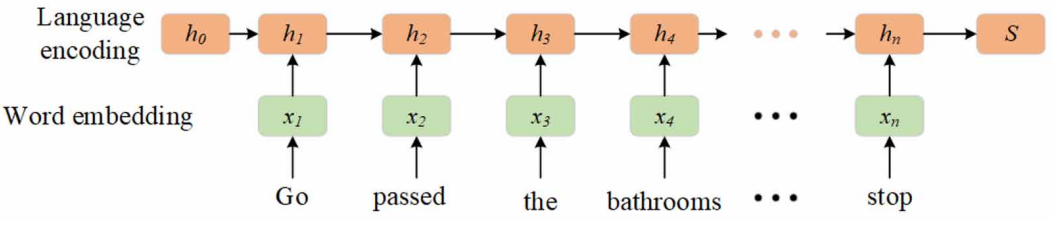
Decision Reasoning Module

GRU models are widely employed in natural language processing and sequence modeling tasks, including action decision-making and progress prediction, due to their ability to record sequential information, alleviate the vanishing gradient problem, facilitate parallelized training, offer efficient training, feature adaptive gating mechanisms, and ensure computational efficiency during inference. Therefore, this paper employs GRU as the inference model to record historical information and make action decisions while introducing an auxiliary task for progress prediction.

As shown in Figure 7, this paper constructs a multimodal fusion neural network model. The fusion model comprises two main parts: an encoding part and a decoding part. The encoding part consists of two sub-networks: the image encoding sub-network and the language encoding sub-network. The outputs of each encoding sub-network are fused to generate state features. In conjunction with the hidden layer output from the previous state and the action, the decoder produces the action and hidden layer for the current state. The robot executes this action to reach the next position.

First, the robot tasked with navigation is placed in any initial position within the scene, which is designated as the current location. An unrestricted natural language navigation command is issued to the robot. This navigation command is input to the language encoding sub-network, which encodes the language sequence of the navigation command, producing a fixed-length semantic vector, denoted as *S*. This natural language navigation command contains the destination of the navigation task and is given by the commander before the robot's movement begins. The content of this command remains

Figure 6. The encoded sub-network of natural language



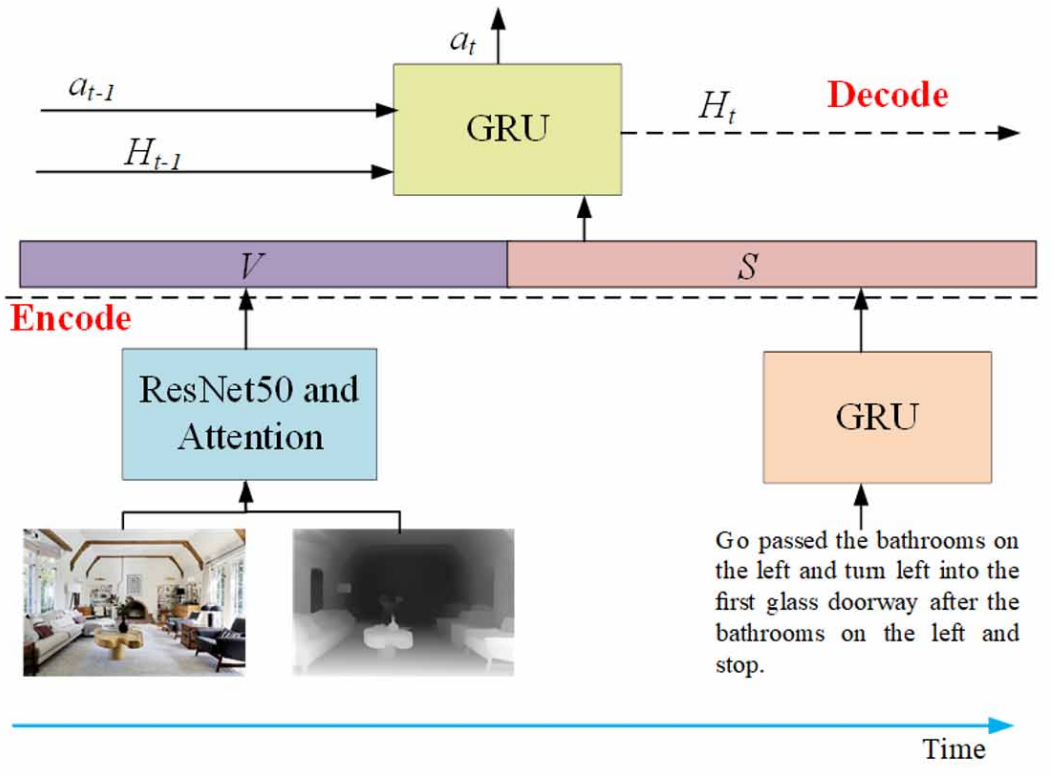
constant while the robot is in motion. Next, the robot, at its current location, uses a stereo camera to capture RGB and depth images corresponding to the current time, denoted as t. The RGB and depth images are processed through an image encoding sub-network to extract features, obtaining the visual feature V_t for the current moment. It is worth noting that extracting features from depth images is crucial for understanding the 3D structure of a scene. These features provide essential information about object distances, aiding robots in acquiring more comprehensive and accurate environmental information and better accomplishing navigation tasks in 3D environments.

Subsequently, the semantic vector S and the visual feature V_t are fused, and the fused feature is input into the GRU within the decoding network. Then, the network outputs the hidden state H_t and the corresponding action probability distribution. The action corresponding to the maximum probability is selected as the robot's action, denoted as a_t, at the current time t. Finally, the robot executes action a_t. When the next time step arrives, setting $t = t + 1$, until the robot's action a_t at the current time t corresponds to stopping, signifying that the robot has reached the destination location specified by the navigation command.

Decoding Network

The decoding network serves the crucial function of translating the learned representations from the encoder into a format that aligns with the desired output. In the context of the discussed VLN model, the decoding network generates actionable navigation instructions based on the fused visual and language features obtained from the encoder. It essentially decodes the internal representations into a sequence of commands or actions that guide the robotic vacuum cleaner through the navigation task. The clarity and effectiveness of the decoding process directly influence the model's ability

Figure 7. Multimodal fusion neural network model



to interpret complex visual and language cues, facilitating accurate and meaningful navigation decisions. As shown in Figure 8, during the feature fusion process of semantic vector S_t and visual feature V_t, both types of features are initially concatenated to form the current state feature, which is a 512-dimensional vector. A dropout mechanism is applied to the state feature, where dropout refers to randomly deactivating certain features during the feature input process with a defined probability (in this paper, a value of 0.2 is used, meaning that 20% of the features in the concatenated feature are randomly set to 0). This results in the final current state feature and enhances data diversity, thereby reducing overfitting.

The local visual information obtained by the robot at two consecutive time steps exhibits relevance and continuity. Therefore, the actions taken by the robot at these two time steps should also exhibit a correlation. In other words, in the sequence of actions the robot takes, there is a strong correlation between the action to be taken at the current time step and the action taken at the previous time step. To achieve this, the action taken at the previous time step, denoted as a_{t-1}, is embedded into a 128-dimensional feature representation (initially, all values are set to 0). This representation is concatenated with the current state feature, forming a feature vector of length 640. The concatenation, as described, involves joining feature vectors or matrices along a specific dimension.

To reduce the parameter count and facilitate the calculation of the next action within the GRU, a one-layer fully connected network is used to map the 640-dimensional feature vector to a 128-dimensional state feature vector. This state feature vector is then output. The state feature, along with the previous time step's hidden state H_{t-1} (with an initial value of all zeros and a vector length of 128), is fed into the gate logic unit of the GRU in the decoding network. This results in the output of the current action feature A_t and the hidden state H_t, which can be used for the output of the next action feature. The output action feature A_t can be represented as follows:

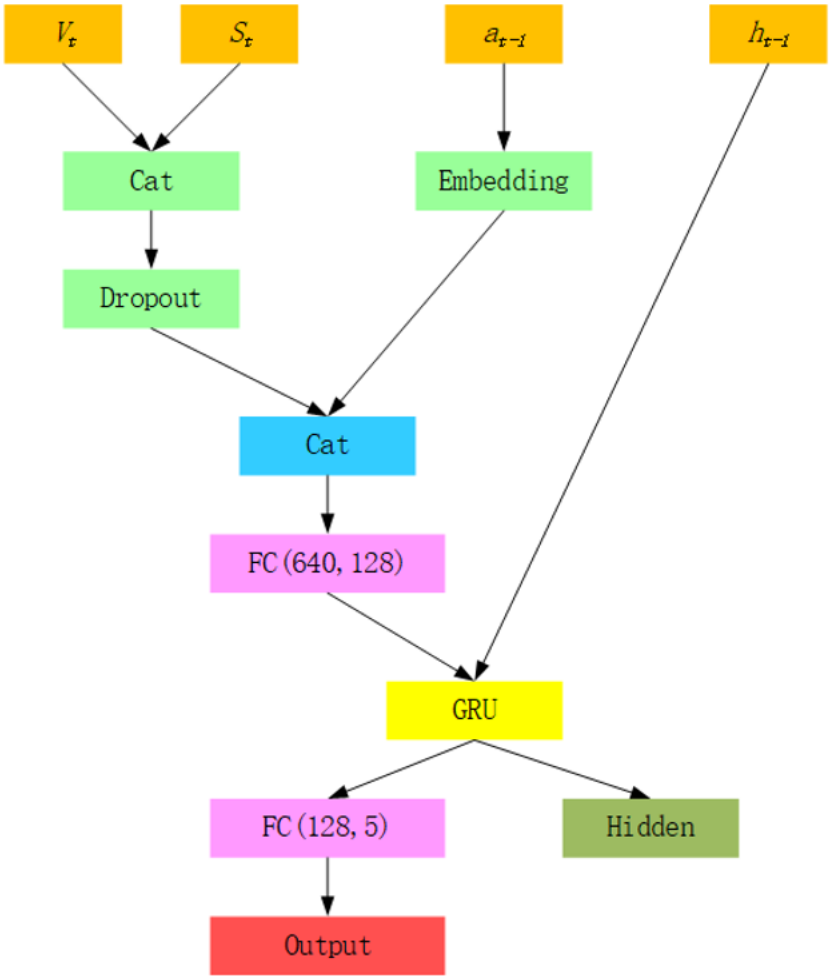
$$A_t = GRU\left(\left[S_t, V_t, a_{t-1}\right], H_{t-1}\right) \tag{6}$$

The robot's set of executable low-level actions includes "forward," "backward," "left turn," "right turn," and "stop." If the robot takes the "stop" action, it signifies that the robot believes it has reached its destination, and the navigation task is considered complete. The action feature A_t is passed through a fully connected layer to produce five corresponding action probability values. These probabilities are then normalized using the softmax function, transforming the model's predictions into a non-negative probability distribution. The action a_t that corresponds to the highest probability is selected as the action the robot will execute next. This process can be represented as follows:

$$a_t = argmax\left(softmax\left(W_a A_t + b_a\right)\right) \tag{7}$$

The robot receives a sequence of natural language instructions and executes an action at each state, effectively converting the instruction sequence into an action sequence. The robot's position is randomly initialized, and it is provided with unrestricted natural language instructions. Based on the current visual observations, the robot continues to observe the current state and takes actions it deems necessary as long as it believes the natural language navigation instruction has not been completed, indicating that it has not reached the target location. When the robot determines that it has reached the target location, it stops and awaits the next natural language navigation instruction.

Figure 8. The detailed structure of the decoded network



EXPERIMENTAL DESIGN AND ANALYSIS OF RESULTS

Dataset

The Matterport3D simulation platform is a software framework for various visual tasks, utilizing the MatterPort3D panoramic RGB-D dataset. This dataset comprises 194,400 RGB-D images extracted from 10,800 panoramic images captured using Matterport cameras. The capture viewpoints are at approximately human observer height, with RGB panoramic images and depth panoramic images taken at each observation point. The Matterport3D dataset provides detailed descriptions of image textures and spatial relationships in 90 indoor scenes and annotates 50,811 object instances within the images. The Matterport3D dataset can be used for tasks such as keypoint matching and semantic label prediction.

Room-to-Room constructed the first real indoor navigation task in a true-to-life indoor setting using the Matterport3D dataset. This dataset includes annotated visual-language navigation data for navigation instructions and paths. The paths are sequences composed of observation points from the Matterport3D dataset, meaning agents can only move discretely between these observation points

within the simulated environment. We collected 7,189 paths, and for each path, we annotated three related navigation instructions. The average length of these instructions is 29 words, notably longer than the questions in visual question-answering tasks.

Experimental Environment

Based on the dataset introduced in Section 4.1, this section describes the sweeping robot navigation based on the multimodal fusion neural network model, with the specific flow shown in Figure 7. For the visual part, we used RGB and depth images as inputs to the model. In this case, the model structure includes Faster-RNN, Restnet50, and attention, and we built the model on the PyTorch framework. We set the learning rate to 0.001 and the batch size to 32. We built a GRU model based on the PyTorch framework for natural language processing. In this case, we set the size of the input sample to 128, the dimension of the hidden layer to 256, the number of stacked GRU layers N to 8, the dropout parameter to 0.01, and the batch size to 32. In addition, we configured the hardware environment for this experiment with a CPU of version Intel Core i9-10900CPU. Since deep learning requires high-performance computation, we also utilized GPUs for computing, with a GPU of version Quadro P2200, and a server with 32 GB of RAM.

Evaluation Indicators

In visual-language navigation tasks, the commonly used evaluation metrics include the following five:

(1) Trajectory length (TL): The total length of the path traveled by the agent during the navigation task, typically measured in meters.
(2) Navigation error (NE): The distance between the location where the agent stops and the target point in meters.
(3) Oracle success (OS) rate: The success rate of completing the task, even if the agent did not stop at the target point but passed through it.
(4) Success rate (SR): The success rate of completing the task by stopping at the target point.
(5) Success weighted by inverse path length (SPL): This metric measures the success of the task while considering the path length, with shorter paths receiving higher scores. It can be expressed as

$$SPL = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{\max\left(p_i, l_i\right)} \tag{8}$$

where S_i is equal to 1 when the navigation task is successfully completed and 0 when it fails, l_i represents the shortest path length from the starting point to the destination, while p_i is the actual path length traveled by the agent.

Among the various metrics for this task, SPL is generally considered the primary measure because it simultaneously considers the success rate of the navigation task and the length of the path traveled. This avoids the limitations of using a single metric, such as when some agents continuously explore the environment without stopping. Such a strategy may yield a high SR, but the TL would also be high. Using these two individual metrics alone cannot comprehensively evaluate the agent's performance.

Experimental Analyses

The Room-to-Room validation set comprises 18 indoor environments, categorized into "val-seen" and "val-unseen." The "val-seen" set encompasses indoor environments in the training set, while the "val-unseen" set includes environments never encountered by the agent during training. Navigation

instructions exclusively form the content of the validation set. The official evaluation server for the Room-to-Room dataset enables the submission of locally generated paths corresponding to test set navigation instructions for evaluation.

The training process of the entire multimodal fusion neural network involves both forward and backward propagation. Forward propagation primarily assesses the loss with the current parameters, while backward propagation updates the parameters based on the gradient of the loss function. Backpropagation occurs for each task associated with a navigation instruction. During backpropagation, the parameters of the convolutional sections of the RGB image and depth image encoding networks remain fixed.

This precaution is taken because errors in model-generated results can impede learning speed and lead to model instability as subsequent outputs become influenced by mistakes. The training process employs the teacher forcing mechanism to avoid the model's excessive dependence on ground truth labels. Teacher forcing is an effective strategy for training RNN models, utilizing outputs from previous time steps as inputs. In essence, during training, the model does not use the previous state's action output as the input for the next state every time; instead, based on a certain probability, it directly incorporates correct actions from the training data as inputs for the next state.

In the training process, the Adam optimizer is used to optimize the model's parameters with a learning rate set to 0.001. Training data is randomly selected from the training dataset for each iteration, and the training is completed after 1,000 iterations. This comparison experiment focuses on Seq2Seq (Anderson et al., 2021), SMNA (Ma et al., 2019), PRESS (Li et al.,2019), EnvDrop (Tan et al., 2019), PREVALENT (Hao et al., 2020), RelGraph (Hong et al., 2020), VLNBERT (Hong et al., 2020), and our proposed scheme.

Table 1 illustrates the experimental outcomes in the "val-seen" set, where "GRVLN-BERT" represents the proposed recurrent visual-language navigation BERT model. The results reveal an enhancement from 0.65 to 0.71 in the SPL metric, marking a 0.06 increase (9.2%). The SR also saw improvement from 0.70% to 0.75%. In a single run without additional datasets, the method introduced in this paper achieves the highest performance on the NE, SR, and SPL metrics. The improvements in the OS and SR metrics suggest that the model, with its capacity to retain historical information, notably boosts navigation reasoning, enabling accurate navigation decisions.

Table 2 showcases comparative experiments in unseen environments. Our method attains a 1% enhancement in the SR metric and a notable 3% increase in the OS metric. Furthermore, by comparing results in both seen and unseen environments, it becomes evident that the proposed method exhibits significant performance variations between the two settings. Nonetheless, it still delivers commendable results in both environments, indicating a degree of generalization capability in the proposed method. Subsequent research should delve deeper into the model's generalization capacities.

Table 1. Comparative experiments in the val-seen environment

	TL	NE	SR	SPL
Seq2Seq	11.33	6.01	0.39	---
SMNA	---	3.22	0.67	0.58
PRESS	10.57	4.39	0.58	0.55
EnvDrop	11	3.99	0.62	0.59
PREVALENT	10.32	3.67	0.69	0.65
RelGraph	10.13	3.47	0.67	0.65
VLNBERT	11.17	3.16	0.7	0.65
Ours	**11.13**	**2.47**	**0.78**	**0.73**

Finally, this paper tested the model's performance on the test dataset and uploaded the results to the test server, as shown in Table 3. The method proposed in this paper outperforms the previous models on each metric, particularly excelling in the SR metric, with a notable improvement of 2% compared to the previous best model.

Ablation Experiments

We conducted detailed ablation experiments to validate the effectiveness of the various components in the approach proposed in this paper. These ablation experiments mainly include replacing the LSTM model with the GRU model, referred to as AE1, removing the attention mechanism from image feature extraction, referred to as AE2, and eliminating Faster R-CNN, referred to as AE3. We conducted these three ablation experiments in both val-seen and val-unseen scenarios. The specific results are depicted in Figures 8 and 9.

In comparing the LSTM and GRU networks, the TL, NE, OS, SR, and SPL metrics demonstrate close similarities in both scenarios, suggesting a comparable performance between the two models. However, training the LSTM in the val-seen scenario takes 43.5 hours, while the GRU proposed in this paper requires 38 hours, resulting in a time savings of 5.5 hours. A similar trend is observed in the val-unseen scenario. In summary, GRU exhibits a lower computational cost during training than LSTM, with a performance nearly equivalent to LSTM. Moreover, upon removing the attention mechanism and object detection using Faster RCNN, a noticeable decline occurs in the robot's accuracy and precision in autonomous navigation. This underscores the importance of the attention mechanism and Faster RCNN, as proposed in this paper, in enhancing the effectiveness of image features.

Table 2. Comparative experiments in the val-unseen environment

	TL	NE	SR	SPL
Seq2Seq	8.39	7.81	0.22	---
SMNA	---	5.52	0.45	0.32
PRESS	10.36	5.28	0.49	0.45
EnvDrop	10.7	5.22	0.52	0.48
PREVALENT	10.19	4.71	0.58	0.53
RelGraph	9.99	4.73	0.57	0.53
VLNBERT	11.63	4.13	0.61	0.56
Ours	**12.53**	**3.85**	**0.63**	**0.57**

Table 3. Comparative experiments in the test environment

	TL	NE	SR	SPL
Seq2Seq	8.13	7.85	0.20	0.18
SMNA	18.04	5.67	0.48	0.35
PRESS	10.77	5.49	0.49	0.45
EnvDrop	11.66	5.23	0.51	0.47
PREVALENT	10.51	5.3	0.54	0.51
RelGraph	10.29	4.75	0.55	0.52
VLNBERT	11.68	4.35	0.61	0.57
Ours	**12.76**	**3.94**	**0.64**	**0.58**

Figure 9. Ablation experiments in the val-seen scenario

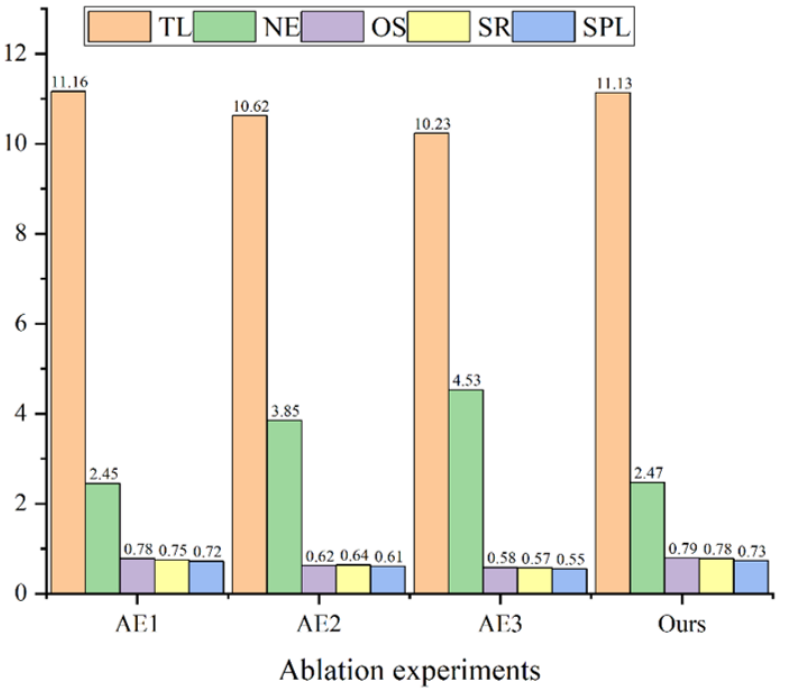
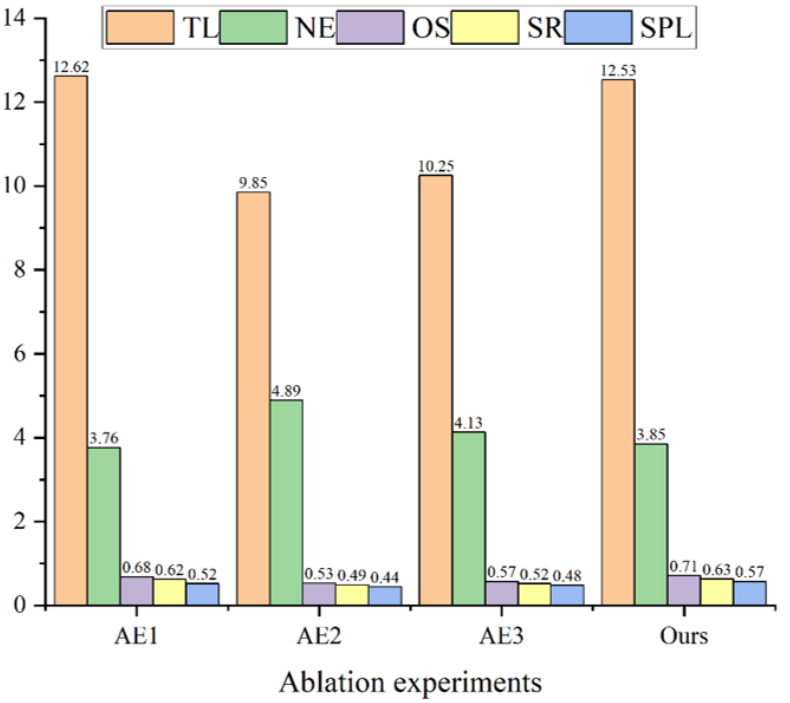


Figure 10. Ablation experiments in the val-unseen scenario



CONCLUSION

This paper presents a multimodal approach to visual-language navigation for robotic vacuum cleaners. Regarding image processing, RGB images are transformed into depth images, and Faster R-CNN and attention mechanisms are employed to extract their visual features. In the domain of language processing, a simpler and more efficient GRU model is used to extract language features. For inference and decision-making, the GRU model is also utilized to acquire the robot's next action values. Multiple sets of comparative experiments and ablation study results indicate the feasibility and effectiveness of the proposed approach. Furthermore, our approach demonstrates improvements in navigation accuracy and performance compared to previous methods. While the multimodal robotic vacuum navigation method proposed in this paper has achieved significant results, some potential areas for improvement remain. Firstly, for visual-language navigation tasks, the current approach may perform sub-optimally in handling highly complex environments or scenarios with occlusions, necessitating more robust environmental modeling and perception capabilities. Secondly, although the attention mechanism employed in this paper plays a crucial role in fusing multimodal information, room exists for further optimization to enhance the model's efficiency in identifying and utilizing critical information. Future research could explore the integration of more advanced visual and language models and more sophisticated multimodal fusion strategies to enhance the navigation performance of robotic systems.

REFERENCES

Anderson, P., Shrivastava, A., Truong, J., Majumdar, A., Parikh, D., Batra, D., & Lee, S. (2021, October). Sim-to-real transfer for vision-and-language navigation. In *Proceedings of the 2020 Conference on Robot Learning* (pp. 671–681). PMLR.

Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., & Van Den Hengel, A. (2018). 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). IEEE. doi:10.1109/CVPR.2018.00387

Biswal, P., & Mohanty, P. K. (2021). Development of quadruped walking robots: A review. *Ain Shams Engineering Journal*, *12*(2), 2017–2031. doi:10.1016/j.asej.2020.11.005

Che, J., Tang, L., Deng, S., & Su, X. (2018). Chinese word segmentation based on bidirectional GRU-CRF model. *International Journal of Performability Engineering*, *14*(12), 3066. doi:10.23940/ijpe.18.12.p16.30663075

Ding, Y., Zhang, Z., Zhao, X., Hong, D., Cai, W., Yu, C., Yang, N., & Cai, W. (2022). Multi-feature fusion: Graph neural network and CNN combining for hyperspectral image classification. *Neurocomputing*, *501*, 246–257. doi:10.1016/j.neucom.2022.06.031

Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1440–1448). IEEE.

Hao, W., Li, C., Li, X., Carin, L., & Gao, J. (2020). 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). IEEE/CVF. doi:10.1109/CVPR42600.2020.01315

Hong, Y., Rodriguez, C., Qi, Y., Wu, Q., & Gould, S. (2020). Language and visual entity relationship graph for agent navigation. *Advances in Neural Information Processing Systems*, *33*, 7685–7696.

Jiang, G. (2023). Security detection design for laboratory networks based on enhanced LSTM and AdamW algorithms. *International Journal of Information Technologies and Systems Approach*, *16*(2), 1–13. doi:10.4018/IJITSA.319721

Lai, P. C., Lin, H. Y., Lin, J. Y., Hsu, H. C., Chu, Y. N., Liou, C. H., & Kuo, Y. F. (2022). Automatic measuring shrimp body length using cnn and an underwater imaging system. *Biosystems Engineering*, *221*, 224–235. doi:10.1016/j.biosystemseng.2022.07.006

Li, J., Wu, L., Qi, J., Zhang, Y., Wu, Z., & Hu, S. (2023). Determinants affecting consumer trust in communication with AI chatbots: The moderating effect of privacy concerns. *Journal of Organizational and End User Computing*, *35*(1), 1–24. doi:10.4018/JOEUC.328089

Li, K., Zhang, Y., Li, K., Li, Y., & Fu, Y. (2022). Image-text embedding learning via visual and textual semantic reasoning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(1), 641–656. doi:10.1109/TPAMI.2022.3148470 PMID:35130144

Li, W., Hong, R., Shen, J., Yuan, L., & Lu, Y. (2023). Transformer memory for interactive visual navigation in cluttered environments. *IEEE Robotics and Automation Letters*, *8*(3), 1731–1738. doi:10.1109/LRA.2023.3241803

Liu, B., Zhao, W., & Sun, Q. (2017, October). Study of object detection based on Faster R-CNN. In *Proceedings of the 2017 Chinese Automation Congress (CAC)* (pp. 6233–6236). IEEE. doi:10.1109/CAC.2017.8243900

Liu, Y., Pei, A., Wang, F., Yang, Y., Zhang, X., Wang, H., Dai, H., Qi, L., & Ma, R. (2021). An attention-based category-aware GRU model for the next POI recommendation. *International Journal of Intelligent Systems*, *36*(7), 3174–3189. doi:10.1002/int.22412

Lu, J., Goswami, V., Rohrbach, M., Parikh, D., & Lee, S. (2020). 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10437–10446). IEEE/CVF. doi:10.1109/CVPR42600.2020.01045

Majumdar, A., Shrivastava, A., Lee, S., Anderson, P., Parikh, D., & Batra, D. (2020). Improving vision-and-language navigation with image-text pairs from the web. In *Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Part VI* (pp. 259–274). Springer International Publishing. doi:10.1007/978-3-030-58539-6_16

Mirowski, P., Grimes, M. K., Malinowski, M., Hermann, K. M., Anderson, K., Teplyashin, D., Simonyan, K., Kavukcuoglu, K., Zisserman, A., & Hadsell, R. (2018). Learning to navigate in cities without a map. In *Proceedings of the 32rd Conference on Neural Information Processing Systems*. NeurIPS.

Ning, E., Wang, Y., Wang, C., Zhang, H., & Ning, X. (2024). Enhancement, integration, expansion: Activating representation of detailed features for occluded person re-identification. *Neural Networks*, *169*, 532–541. doi:10.1016/j.neunet.2023.11.003 PMID:37948971

Qi, Y., Wu, Q., Anderson, P., Wang, X., Wang, W. Y., Shen, C., & Hengel, A. V. D. (2020). Reverie: Remote embodied visual referring expression in real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9982–9991). IEEE/CVF. doi:10.1109/CVPR42600.2020.01000

Shen, J., & Wei, K. (2023). CR-BiGRU intrusion detection model based on residual network. *Journal of Jilin University: Science Edition*, *61*, 353–361.

Sun, P. (2023). Personalized course resource recommendation algorithm based on deep learning in the intelligent question answering robot environment. *International Journal of Information Technologies and Systems Approach*, *16*(3), 1–13. doi:10.4018/IJITSA.320188

Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. *Proceedings of the Thirteenth Annual Conference of the International Speech Communication Association*.

Swift, G., Molinski, T. S., & Lehn, W. (2001). A fundamental approach to transformer thermal modeling. I. Theory and equivalent circuit. *IEEE Transactions on Power Delivery*, *16*(2), 171–175. doi:10.1109/61.915478

Tellex, S., Kollar, T., Dickerson, S., Walter, M., Banerjee, A., Teller, S., & Roy, N. (2011). Understanding natural language commands for robotic navigation and mobile manipulation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *25*(1), 1507–1514. doi:10.1609/aaai.v25i1.7979

Thomason, J., Murray, M., Cakmak, M., & Zettlemoyer, L. (2020). Vision-and-dialog navigation. In *Proceedings of the Conference on Robot Learning* (pp. 394–406). PMLR.

Varma, S., & James, D. P. (2021). An efficient deep learning-based video captioning framework using multi-modal features. *Expert Systems: International Journal of Knowledge Engineering and Neural Networks*, exsy.12920. doi:10.1111/exsy.12920

Verdú, S., Barat, J. M., & Grau, R. (2023). Laser scattering imaging combined with CNNs to model the textural variability in a vegetable food tissue. *Journal of Food Engineering*, *336*, 111199. doi:10.1016/j.jfoodeng.2022.111199

Wang, J., Zhang, Y., Yu, L. C., & Zhang, X. (2022). Contextual sentiment embeddings via bi-directional GRU language model. *Knowledge-Based Systems*, *235*, 107663. doi:10.1016/j.knosys.2021.107663

Wang, L., Li, Y., Huang, J., & Lazebnik, S. (2018). Learning two-branch neural networks for image-text matching tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(2), 394–407. doi:10.1109/TPAMI.2018.2797921 PMID:29994350

Wang, T., Wu, Z., & Wang, D. (2021). Visual perception generalization for vision-and-language navigation via meta-learning. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(8), 5193–5199. doi:10.1109/TNNLS.2021.3122579 PMID:34780332

Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y. F., Wang, W. Y., & Zhang, L. (2019). Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6629–6638). doi:10.1109/CVPR.2019.00679

Wen, S., Lv, X., Yu, F. R., & Gong, S. (2021). Vision-and-language navigation based on cross-modal feature fusion in indoor environment. *IEEE Transactions on Cognitive and Developmental Systems*, *15*(1), 3–15. doi:10.1109/TCDS.2021.3139543

Xiao, S., & Fu, W. (2023). Visual navigation based on language assistance and memory. *IEEE Access : Practical Innovations, Open Solutions*, *11*, 13996–14005. doi:10.1109/ACCESS.2023.3239837

Xie, Y. (2023). Optimization of enterprise financial performance evaluation system based on AHP and LSTM against the background of carbon neutrality. *Journal of Organizational and End User Computing*, *35*(1), 1–14. doi:10.4018/JOEUC.332810

Xu, X., Zhao, M., Shi, P., Ren, R., He, X., Wei, X., & Yang, H. (2022). Crack detection and comparison study based on faster R-CNN and mask R-CNN. *Sensors (Basel)*, *22*(3), 1215. doi:10.3390/s22031215 PMID:35161961

Yang, X., Li, H., Ni, L., & Li, T. (2021). Application of artificial intelligence in precision marketing. *Journal of Organizational and End User Computing*, *33*(4), 209–219. doi:10.4018/JOEUC.20210701.oa10

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, *31*(7), 1235–1270. doi:10.1162/neco_a_01199 PMID:31113301

Zeng, F., Wang, C., & Ge, S. S. (2020). A survey on visual navigation for artificial agents with deep reinforcement learning. *IEEE Access : Practical Innovations, Open Solutions*, *8*, 135426–135442. doi:10.1109/ACCESS.2020.3011438

Zhang, H., Wang, C., Tian, S., Lu, B., Zhang, L., Ning, X., & Bai, X. (2023). Deep learning-based 3D point cloud classification: A systematic survey and outlook. *Displays*, *79*, 102456. doi:10.1016/j.displa.2023.102456

Zhang, Y., Zuo, X., Zuo, W., Liang, S., & Wang, Y. (2021). Bi-LSTM+GCN causality extraction based on time relationship. *Journal of Jilin University: Science Edition*, *59*, 643–648.

Zhao, J., Mao, X., & Chen, L. (2018). Learning deep features to recognise speech emotion using merged deep CNN. *IET Signal Processing*, *12*(6), 713–721. doi:10.1049/iet-spr.2017.0320

Zheng, M., Qi, G., Zhu, Z., Li, Y., Wei, H., & Liu, Y. (2020). Image dehazing by an artificial image fusion method based on adaptive structure decomposition. *IEEE Sensors Journal*, *20*(14), 8062–8072. doi:10.1109/JSEN.2020.2981719

Yiping Zhang was born in Shanghai, China, in 1978. He studied in Shanghai Jiaotong University, majoring Industrial Design B.A. and graduated from DFI College of Communication Art and New Media in Hamburg, Germany, majoring Digital Communication Design, and was awarded M.A. degree of Design in Digital Media at University of Portsmouth, UK. Now he is an Associate Professor in Zhejiang Wanli University at Sino-German Institute of Design and Communication. His research areas include digital media design, technology in digital interaction, user interface and user experiences.

Kolja Wilker was born in Hamburg, Germany, in 1974. He is a programmer and senior lecturer of interaction technology. He graduated from DFI College of Communication Art and New Media in Hamburg, Germany. Now he is working in Zalando STUPS as software developer and lecturing at various universities in Germany. His research areas include Algorithm and Visual-language Navigation, Computer Vision. (Corresponding author)