CIT. Journal of Computing and Information Technology, Vol. 30, No. 4, December 2022, 257–273
doi: 10.20532/cit.2022.1005689


# Deep Learning-Based Visual Navigation Algorithms for Mobile Robots: A Comprehensive Study

Wei Yu and Xinzhi Tian

Xi'an Siyuan University, Xi'an, Shaanxi, China

This research addresses the challenges faced by mobile robots in efficiently navigating complex environments. A novel approach is proposed, leveraging deep learning techniques, and introducing the Neo model. The method combines Split Attention with the ResNeSt50 network to enhance the recognition accuracy of key features in the observed images. Furthermore, improvements have been made in the loss calculation method to improve navigation accuracy across different scenarios. Evaluations conducted on AI2-THOR and active vision datasets demonstrate that the improved model achieves higher average navigation accuracy (92.3%) in scene 4 compared to other methods. The success rate of navigation reached 36.8%, accompanied by a 50% reduction in ballistic length. Additionally, compared to HAUSR and LSTM-Nav, this technology significantly reduced collision rates to 0.01 and reduced time consumption by over 8 seconds. The research methodology addresses navigation model accuracy, speed, and generalization issues, thus making significant advancements for intelligent autonomous robots.

*ACM CCS (2012) Classification:* Computing methodologies → Machine learning → Machine learning algorithms

Artificial intelligence → Computer vision → Vision for robotics

*Keywords*: deep learning, Neo-model, mobile robots, visual navigation, split attention, ResNet

## 1. Introduction

In recent years, mobile robot technology has been extensively applied in various fields with visual navigation being a vital research area [1–2]. Visual navigation for mobile robots encompasses the process of capturing environmental data using visual sensors like cameras, using this information for decision-making and path planning, and enabling autonomous navigation in complex environments.

Conventional visual navigation algorithms primarily rely on manually created features and rules that are limited by feature representation and rule constraints, thus performing poorly in dynamic and complex environments [3]. Traditional sensor-based navigation methods are unable to handle sudden problems in unknown environments, weak in perceiving and analyzing the surrounding environment, and have limited emergency handling capabilities. These constraints collectively impede the development of robot navigation technology [4].

However, deep learning techniques provide a new possibility for addressing these issues. Deep learning is a subfield of machine learning that leverages artificial neural networks to attain data abstraction and representation through multi-level neural network structures [5–6]. It aims to learn advanced data representations via multi-level nonlinear transformations, enabling it to solve complex pattern recognition and decision-making problems. During training, raw data is fed into a neural network via an input layer, and features are extracted, abstracted, and transformed via hidden layers. The output layer generates predictions or classification results. Despite its numerous benefits, deep learning presents some challenges. It requires massive amounts of data and computational resources, necessitates high-quality data, and requires substantial storage. Deep learning models are

often intricate, making them less interpretable, prone to overfitting, and potentially leading to poor performance on new data. Moreover, deep learning is less reliant on human knowledge, meaning it may ignore crucial features, resulting in inaccurate model predictions.

In order to tackle the challenges faced by traditional autonomous robot navigation methods, Zhao *et al.* developed a path-planning navigation system for mobile robots based on panoramic vision [7]. The system aimed to address the issues of high computational cost and complex external environments. It employed a panoramic vision sensor and utilized a breadth-first search method with recurrent neural networks for path planning. Experimental results demonstrated that the system achieved a path length reduction ranging from 20.7% to 35.9%, thereby showing promising practical application effects.

To address the existing limitations in mobile robot navigation performance, Fang *et al.* proposed a novel approach combining imitation learning and deep reinforcement learning frameworks [8]. This approach leveraged surrounding images as observation points and employed template-matching methods for determining stop actions. Experimental comparisons indicated that this method outperformed end-to-end deep reinforcement learning approaches and exhibited stronger practicality.

Despite the notable achievements of existing algorithms in specific scenarios, there are still limitations that need to be addressed. Notably, these algorithms tend to be sensitive to environmental changes and interferences, resulting in decreased performance in complex environments. Additionally, the computational resources and runtime requirements of these algorithms are typically high, which hinders the real-time navigation capability of mobile robots. To overcome these challenges, this study adopts an attention mechanism that emulates the functioning of the human visual system [9]. A visual navigation model is constructed by integrating the proposed Neo model. Further enhancements are made through the utilization of cross-stage partial networks and split attention, aiming to improve the effectiveness of visual navigation for mobile robots. In order to address the issue of decreased navigation accuracy in deep reinforcement learning-based visual navigation algorithms caused by scene changes, a novel visual navigation model is proposed. This model combines the attention mechanism with the next expected observations (Neo). Building upon the original Neo model, split-attention and cross-connected ResNeSt50 network components are introduced to enhance the recognition accuracy of key features in the current observation image. Additionally, improvements are made to the calculation method of loss, thereby enhancing the navigation accuracy of the model across different scenarios. Furthermore, by integrating deep learning technology with mobile robot visual navigation, this research aims to achieve a more intelligent, accurate, and efficient mobile robot navigation system. The objective is to generate precise navigation decisions, thereby improving navigation effectiveness and robustness.

The article is divided into four sections. The first section covers the research background and current status of the combination of visual navigation algorithms and deep learning technology for mobile robots. The second section introduces the attention mechanism and proposes a visual navigation network based on the Neo model. Subsequently, the third section presents the improved version of the visual navigation model by introducing split attention and a cross-stage partial network to further enhance its performance. In the fourth section, the effectiveness of the proposed navigation algorithm is evaluated, including performance testing and analysis of actual application effects. Finally, the paper concludes with a summary of key findings and outlines prospective future research directions.

## 2. Research Method

### 2.1. Navigation Framework Design Based on the Neo Model

This study presents the construction of a visual navigation model based on the attention mechanism and the proposed Neo model. The attention mechanism is a technique that mimics the working principles of the human visual system and is used to selectively process input infor-

mation in machine learning and deep learning tasks [9–10]. It simulates the human attention mechanism, allowing the system to focus on input parts related to the current task and ignore other irrelevant information. Through the attention mechanism, the model can selectively focus on useful input information and dynamically adjust the level of attention to different positions, thereby improving the model's performance and generalization ability.

The Convolutional Block Attention Module (CBAM) is an attention mechanism used to enhance the performance of convolutional neural networks, consisting of two sub-modules: the Channel Attention Module (CAM) and the Spatial Attention Module (SAE) [11–12]. The structure of the two attention mechanisms is shown in Figure 1.

The spatial attention module, as depicted in Figure 1(a), learns the importance weights of feature maps by using maximum pooling and average pooling. It can adaptively adjust the weights of different spatial positions to extract more important spatial information. This helps the network to better focus on areas of interest, thereby improving the perception ability [13].

Conversely, the channel attention module, depicted in Figure 1(b), learns the importance weights of channel features through global average pooling and fully connected layers and can adaptively adjust the weights of different channels to extract more important features. This helps the network to better focus on key features, thereby improving the expression ability of features.

By combining the channel attention module and spatial attention module, the CBAM module can simultaneously extract channel and spatial attention information, thereby enhancing the network's ability to perceive important features [14]. The structure of the CBAM module is shown in Figure 2.

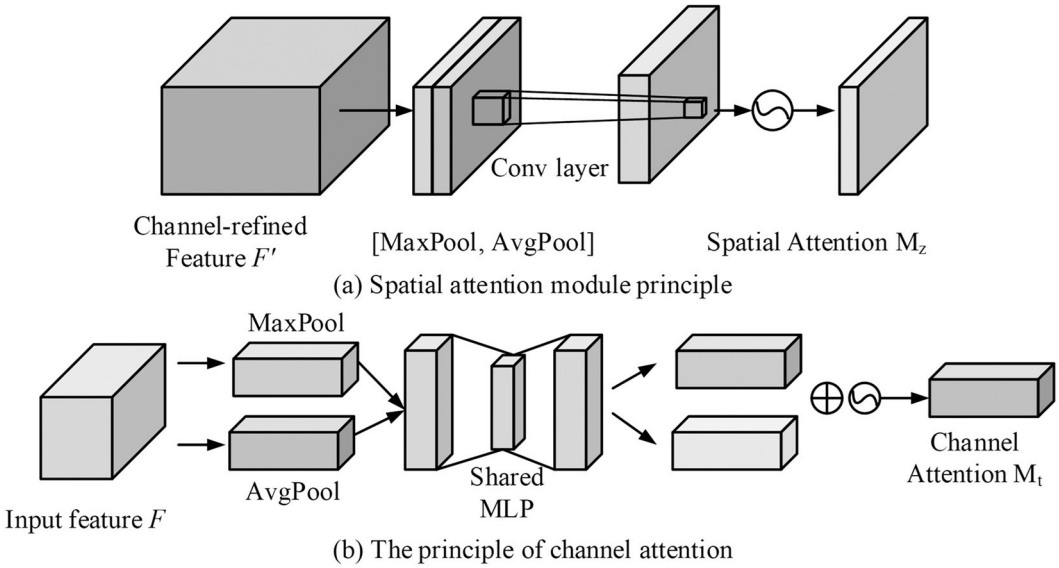



*Figure 1.* Channel attention module and spatial attention module.

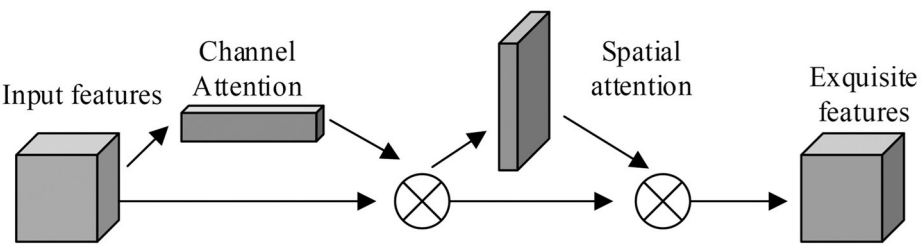



*Figure 2.* CBAM module structure.

For the visual navigation network based on the Neo model proposed in this paper, the intelligent agent optimizes its navigation by taking the minimum steps required to navigate to the target location – thus considering it as the pursuit direction. This approach allows the agent to navigate effectively in new scenarios, thereby validating the model generalization ability [15].

In scenarios where the current observation object $X$ is known, the research methodology does not need to directly predict the optimal action corresponding to the next moment. On the contrary, it sets the best action at the next moment to be known and the state to have been executed, thereby generating a model to obtain the expected observation value at the next moment, which is calculated using equation (1).

$$p_\theta\left(\hat{x},\, z \mid x,\, a\right) = p_\theta\left(\hat{x} \mid z\right) p_\theta\left(z \mid x,\, a\right) \quad (1)$$

In equation (1), $a$ represents the next action, $z$ represents the potential variable, $\hat{x}$ represents the expected observation at the next moment, $x$ corresponds to the next observation, $p_\theta\left(\hat{x},\, z \mid x,\, a\right)$ represents the parameter model composed of the joint distribution of the potential variable and the expected observation.

In order to effectively train the generated model, it is necessary to maximize the marginal logarithmic likelihood $\log p_\theta\left(\hat{x} \mid x,\, a\right)$. However, there is a certain degree of complexity in solving marginal likelihood, which can easily increase the difficulty of neural network parameterization [16]. At the same time, in essence, the goal $g$ plays a decisive role in the next best action, yet it remains unknown a priori. To address this, edge similarity is optimized by employing variational reasoning and introducing a posterior probability $p_\theta\left(z \mid x,\, a\right)$ of the inference network with parameter $\lambda$, as shown in equation (2).

$$\begin{gathered} \log p_\theta\left(\hat{x} \mid x,\, a\right) \\ \geq \\ E_{z\sim q\lambda(z|x,g)}\left[\log \frac{p_\theta\left(\hat{x},\, z \mid x,\, a\right)}{q_\lambda\left(z \mid x,\, g\right)}\right] = L\left(\hat{x}\right) \end{gathered} \quad (2)$$

In equation (2), $p_\theta\left(z \mid x,\, a\right)$ represents a posterior probability and $q_\lambda\left(z \mid x,\, g\right)$ represents the inference network with the introduced parameter. The objective function formed by this lower bound is represented by equation (3).

$$\begin{aligned} J &= -E_{z\sim q\lambda(z|x,g)}\left[\log p_\theta\left(\hat{x} \mid z\right)\right] \\ &\quad + KL\left[q_\lambda\left(z \mid x, g\right) \| \, p_\theta\left(z \mid x, a\right)\right] \\ &= -L\left(\hat{x}\right) \end{aligned} \quad (3)$$

In equation (3), $KL$ is the KL divergence (Kull back Leibler Divergence). In the case where a mixed prior is imposed on a potential distribution due to real ground actions and current observations, $p_\theta\left(z \mid x,\, a\right)$ can be estimated as a Gaussian distribution.

To achieve the goal of robot navigation, the proposed Neo model can train a navigation action classifier that predicts the next best action based on current observations, previous actions, and generated $\hat{x}$. Taking action prediction into account, the objective function is obtained as shown in equation (4).

$$\begin{aligned} J &= -\alpha E_{z\sim q\lambda(z|x,g)}\left[\log p_\theta\left(\hat{x} \mid z\right)\right] \\ &\quad + \beta KL\left[q_\lambda\left(z \mid x, g\right) \| \, p_\theta\left(z \mid x, a\right)\right] \\ &\quad + \gamma E_{a-p(a)}\left[-\log q_\varphi\left(a \mid x, \hat{x}, \tilde{a}\right)\right] \end{aligned} \quad (4)$$

In equation (4), $q_\varphi\left(a \mid x, \hat{x}, \tilde{a}\right)$ represents the generated navigation action classifier, $\beta$ and $\gamma$ are hyper parameters, with corresponding set values of 0.01, 0.0001, and 1, respectively. The probability graph of the navigation model is shown in Figure 3.

Within the proposed Neo model-based navigation framework, the input of the variational inference module comprises the current robot position and the target point view. These inputs undergo feature extraction via ResNet-50, resulting in and the 2048-D feature vectors.

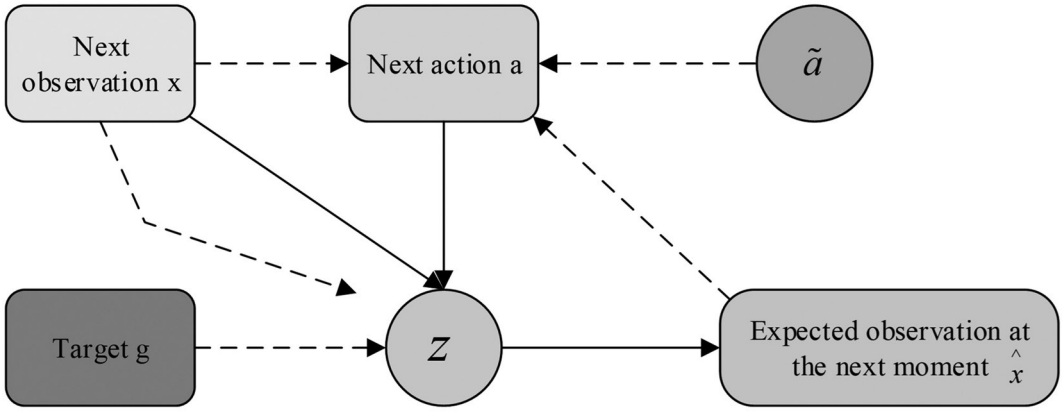



*Figure 3.* Probability Graph Model of Navigation Model.

After resizing the image input to a resolution of 64*64, a potential variable vector with a dimension of 400 is derived from 2048-D feature vectors via a multi-layer perceptron. In this step, minimizing the KL divergence loss is vital, as it ensures a closer alignment with the prior estimation of potential variable distribution.

The Neo generation module includes a 5-layer convolutional network and a 2-layer multi-layer perceptron, which can obtain the Neo model of the front view from potential vectors [17]. The action prediction module includes four layers of multi-layer perceptrons, which can concatenate and map the last layer features, previously extracted features, and current observed features of the generation module to the next action and train the parameters of the self network through ground real actions. The navigation framework based on the Neo model is shown in Figure 4.

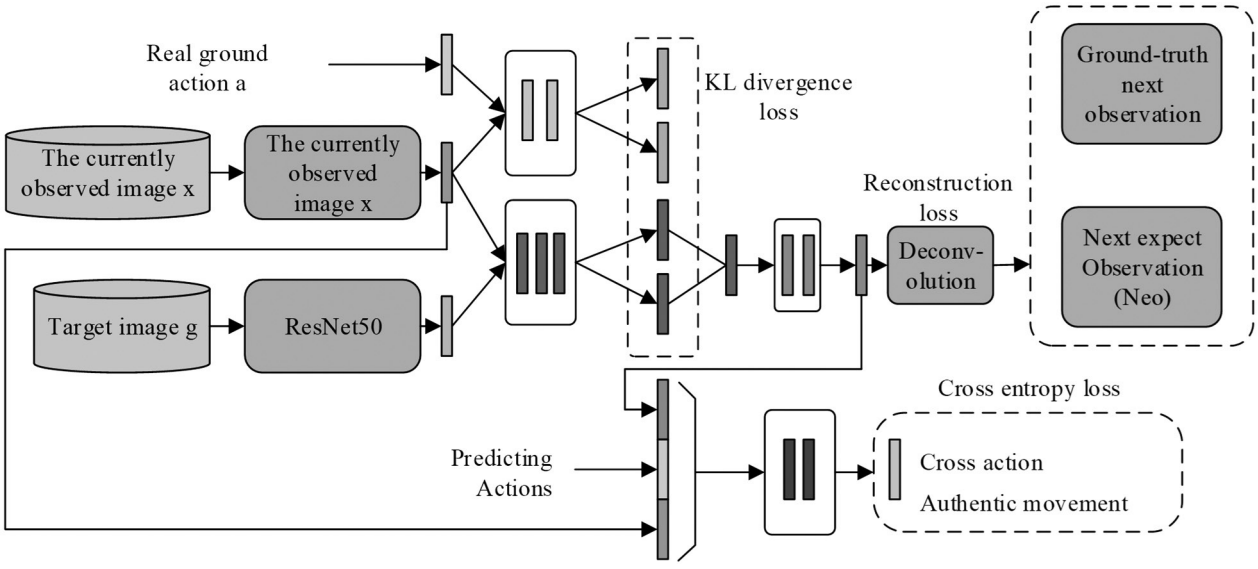



*Figure 4.* Navigation framework based on the Neo model.

## 2.2. Optimization Method of Neo Model Visual Navigation Based on Split Attention and Cross-connection

Using the proposed Neo model for visual navigation requires addressing the issue of generalization, which relates to the trained navigation model's ability to maintain its original performance in new application scenarios. Therefore, further research has been conducted to improve the ability of intelligent agents to extract the main information of input images and enhance their adaptability to new scenarios by splitting attention, cross-connection methods, and loss functions.

In this study, the ResNet50 network from the original model is replaced by ResNeSt50 with a Split Attention structure, which includes several Split Attention blocks stacked in ResNet style. Meanwhile, the improved ResNet50 can span different feature maps during the use of attention, resulting in a relatively low model complexity and better transfer conditions for the algorithm model. The proposed ResNeSt block network structure is shown in Figure 5.

In Figure 5, the Split Attention block calculation unit mainly consists of two parts, namely split attention and feature map group. Here, the number of feature map groups depends on the hyper parameter $k$, and the number of cardinality group splits depends on the parameter $R$. Therefore, the total number of feature groups is calculated as shown in equation (5).

$$p_\theta\left(\hat{x}, z \mid x, a\right) = p_\theta\left(\hat{x} \mid z\right) p_\theta\left(z \mid x, a\right) \qquad (5)$$

In equation (5), $G$ represents the total number of feature groups. The input feature map is first divided into base arrays, and then all main groups are split into $R$ parts, and each part is merged into the split attention module under convolution operation of 1*1 and 3*3. Building upon this, feature concatenation operations are performed on the output features from the $K$ base arrays while maintaining consistent input and output sizes. The split attention blocks integrate the mechanism of channel attention, assigning weights to different channels, and describing the importance of each channel [18]. The basic structure of splitting attention blocks is shown in Figure 6.

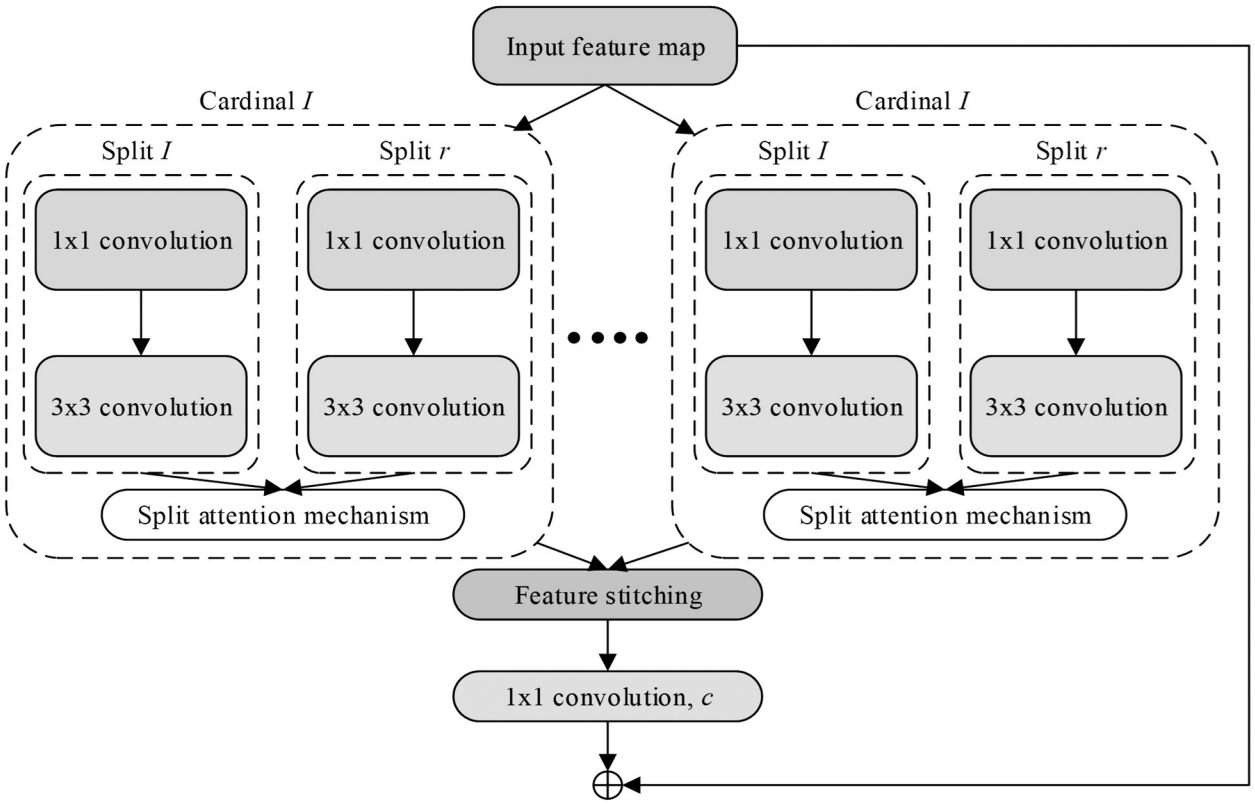



*Figure 5.* The proposed ResNeSt block network structure.

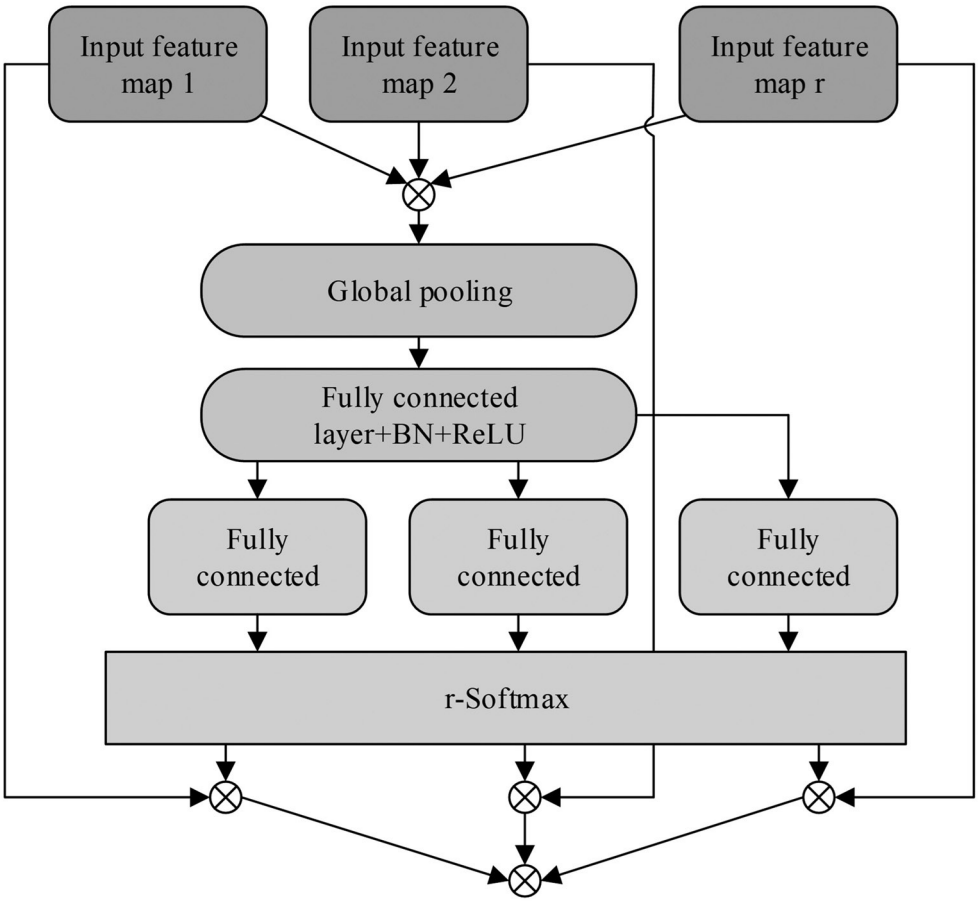



*Figure 6.* The Basic Structure of Split Attention Blocks.

In Figure 6, the input of the $k$ cardinality group is calculated using equation (6).

$$U_k = \sum_{i=R(k-1)}^{RK} U_i \tag{6}$$

In equation (6), $U_k$ represents the input of the cardinality group. Under the cross spatial global average pooling operation, global context information can be fully collected. Meanwhile, the channel weight statistics of the input feature map can be calculated. Among them, the $c$ component is obtained by equation (7).

$$s_c^k = \frac{1}{H \times W} \sum_{i}^{H} \sum_{j=1}^{W} U_c^k (i, j) \tag{7}$$

In equation (7), $H$ and $W$ respectively represent the height and width dimensions of the channel, while $s_c{}^k$ represents the global average pooling result of the $c$ component. The weights $a_i^k(c)$ obtained using the SoftMax activation function are shown in equation (8).

$$a_i^k(c) = \begin{cases} \dfrac{\exp\left(\delta_i^c\left(s^k\right)\right)}{\sum_{j=0}^{R} wxp\left(\delta_i^c\left(s^k\right)\right)} & \text{if } R > 1 \\ \dfrac{1}{\exp\left(-\delta_i^c\left(s^k\right)\right)} & \text{if } R = 1 \end{cases} \tag{8}$$

Then the output of each Cardinal obtained is concatenated, and the final output is obtained as shown in equation (9).

$$V = Concat\left\{V^1, V^2, ..., V^K\right\} \tag{9}$$

Finally, the ResNeSt block is stacked in the form of ResNet50 to obtain the proposed ResNeSt50. Compared to ResNet50, ResNeSt50

can achieve better results by increasing parameters while maintaining the same computational complexity.

In order to further reduce the complexity of the model, and achieve a lightweight design, a cross stage partial network (CSPNet) is studied to optimize ResNeSt50 and design a CSP-ResNeSt50 feature extraction network that integrates CSPNet.

After inputting the feature map, the network uses channel segmentation to obtain two segments: one represents the ResNeSt module that has gone through multiple stages, while the other represents the ResNeSt module that has passed through half of the number of channels [19]. After performing convolution and filtering operations, the first and second segments complete feature merging, resulting in a total channel count of 3c/2. In this process, the gradient flow is truncated without excessive duplicate gradients, and the migration and deployment conditions of the visual navigation model are more favorable. The designed CSP-ResNeSt50 feature extraction network structure is shown in Figure 7.

Next, the loss calculation method is improved. To measure the difference in probability distribution between inference networks and true posterior probabilities, the proposed Neo model adopts the KL divergence measurement method, which is an indicator used to measure the difference between two probability distributions and is calculated using equation (10).

$$D_{KL}\left(p_\theta\left(z \mid x,\, a\right) \| \, q_\lambda\left(z \mid x,\, g\right)\right) = \int_{-\infty}^{\infty} p_\theta\left(z \mid x,\, a\right) \ln \frac{p_\theta\left(z \mid x,\, a\right)}{q_\lambda\left(z \mid x,\, g\right)} d(x) \tag{10}$$

However, the KL divergence does not satisfy symmetry, and when comparing the differences between two probability distributions, the results will depend on the selected benchmark distribution. Meanwhile, the calculation of KL divergence depends on the distribution of data samples. When the sample size is small or not representative, the calculated KL divergence may be biased or misleading [20–21]. In this case, in order to ensure that the predicted and actual observation values in the visual navigation of mobile robots do not change based on changes in reference metrics, this study utilizes the optimal transmission idea to improve and calculate the loss through Sinkhorn distance. The optimal transmission theory takes the basic metric space as the consideration object and can provide a method for comparing degenerate

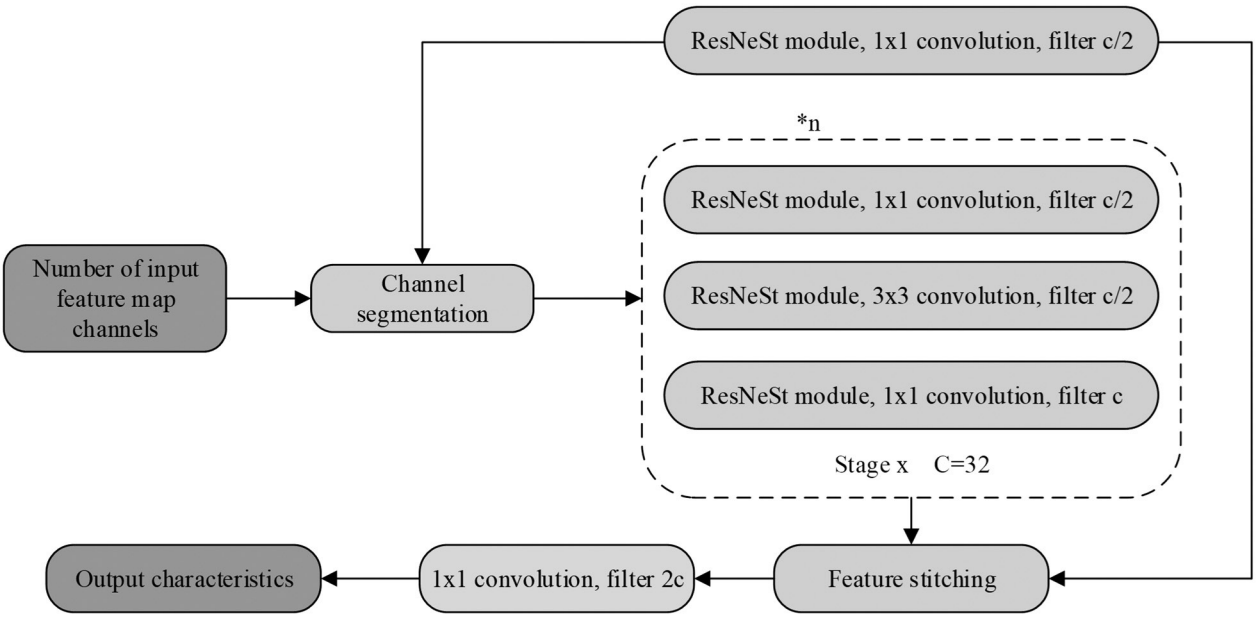



*Figure 7.* CSP-ResNeSt50 Feature Extraction Network Structure.

distributions [22–23]. This results in the updated expression of the objective function in the entire visual navigation network as shown in equation (11), where $J$ represents the objective function value.

$$\begin{aligned} J = & -\alpha E_{z\sim q_\lambda(z|x,g)}\left[\log p_\theta\left(\hat{x}\,|\,z\right)\right] \\ & + \beta w_s^{\beta}\left[p_\theta\left(z\,|\,x,a\right),\, q_\lambda\left(z\,|\,x,g\right)\right] \\ & + \gamma E_{a\sim p(a)}\left[-\log q_\varphi\left(a\,|\,x,\hat{x},\tilde{a}\right)\right] \end{aligned} \quad (11)$$

## 3. Results

To verify the effectiveness of the proposed visual navigation intelligent agent, the TensorFlow deep learning framework is employed on an NVIDIA 2080 Ti GPU. Experiments were conducted on Ubuntu 16.04.

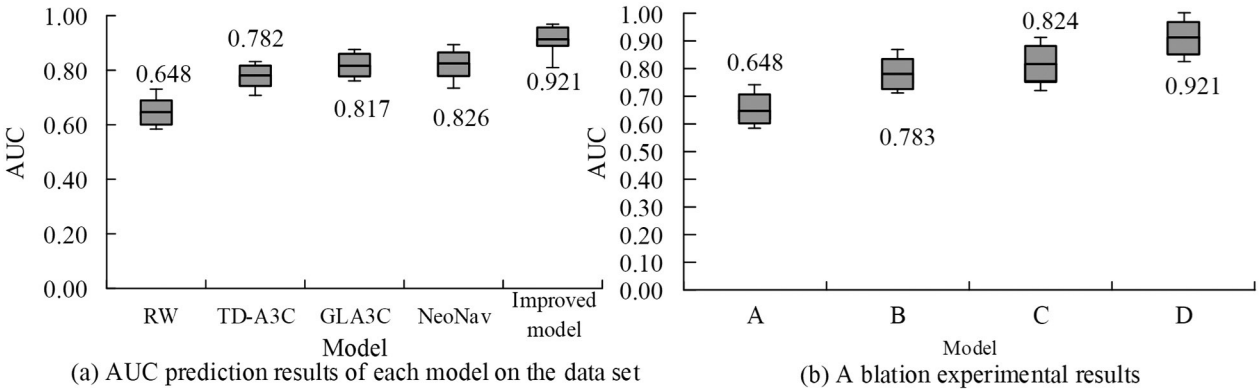
The dataset used for the experiments is the Allen Institute for Artistic Intelligence For Object Recognition (AI2-THOR), which is a three-dimensional visual and physical simulation environment for machine intelligence and reasoning ability [24]. This dataset aims to provide training and evaluation benchmarks for machine learning algorithms for tasks such as visual perception, semantic understanding, and inference, providing a virtual indoor scene that includes various home environments, furniture, objects, and sensors [25]. The AI2-THOR simulation environment includes four layouts: kitchen, living room, bedroom, and bathroom, each divided into 30 scenes. The study used the first 20 scenarios of all layout types to form a training set, and the remaining 10 scenarios became a testing set. The navigation accuracy of each method was compared, and the results are shown in Table 1.

The data presented in Table 1 clearly demonstrates that the improved model outperforms other methods with an average navigation accuracy of 92.3% across all four scenes. Compared to RW, TD-A3C, GLA3C and NeoNav, the navigation accuracy is 13.7%, 13.3%, 10.6% and 8.1% higher than the average of RW, TD-A3C, GLA3C, and NEONAV in the four medium scenarios, respectively. Figure 8 shows the ablation experimental results of the research method. A, B, C and D represent the visual navigation model, the Neo model-based visual navigation model, and the research model, respectively.

In Figure 8(a), the AUC value of the research model is 27.3%, 13.9%, 10.4% and 9.5% higher than that of classical navigation algorithms RW, TD-A3C, GLA3C, and NEONAV, respectively. In Figure 8(b), the AUC value of simple visual navigation is only 64.8%. After adding the Neo model, the AUC value is 78.3%. The research model builds on this, adding split at-tention and cross-connections, and has a higher AUC value of 92.1%. This shows that the improved model has some advantages.

*Table 1*. Comparison of Navigation Performance of Various Models in AI2-THOR Environment.

| Model type | Kitchen (%) | Living (%) | Bed (%) | Bath (%) | Avg (%) |
|---|---|---|---|---|---|
| RW | 76.1 | 72.4 | 82.6 | 83.3 | 78.6 |
| TD-A3C | 78.3 | 74.2 | 79.2 | 84.3 | 79.0 |
| GLA3C | 81.4 | 77.3 | 80.5 | 87.6 | 81.7 |
| NeoNav | 83.2 | 79.7 | 83.6 | 90.3 | 84.2 |
| Improved model | 92.9 | 89.8 | 91.6 | 94.9 | 92.3 |




(a) AUC prediction results of each model on the data set (b) A blation experimental results

*Figure 8.* AUC values and ablation results of each model.

In Table 2, the navigation Success rate (SR) and the success rate weighted by path length (SPL) of each model in the KITTI dataset are compared. These results show that after training and testing on the active visual dataset KITTI, the SR of the improved model is slightly lower than NeoNav in the Living scenario, and significantly improved in the other two scenarios, while the average value of the four scenarios has a partial improvement compared with Ne-oNav, with an increase of about 3%. With respect to the SPL, the improved model performed better than NeoNav in all four scenarios, improving by about 6%.

The results of the loss values of the two types of agents before and after improvement are shown in Figure 9. As depicted in Figure 9(a), it is evident that prior to the improvement, the maximum loss value obtained from the proposed intelligent agent testing was 2.2. After 15 rounds of self-training, the loss function curve of the intelligent agent testing converged, approximately 0.3. Figure 9(b) shows that during the testing process, there has been a significant convergence trend near the fifth round of the improved agent, with a corresponding loss value of only 0.1. This indicates that the improved performance of the intelligent agent has been significantly improved, proving the effectiveness of the improved method [26].

For evaluation using the AI2-THOR dataset, four scenarios were selected: kitchen-02, living-08, bathroom-02, and bedroom-04. The evaluation index was the average trajectory

*Table 2*. Comparison of SR and SPL of each model in KITTI data set.

| Model type | Kitchen | Living | Bed | Bath | Avg |
|---|---|---|---|---|---|
| RW | 7.0/3.5 | 1.8/1.0 | 2.6/1.5 | 17.9/ 8.0. | 7.3/3.5 |
| TD-A3C | 11.4/1.6 | 5.6/0.4 | 5.3 / 0.7 | 24.3/2.3 | 11.7/1.3 |
| GLA3C | 13.1/3.2 | 4.9/1.1 | 5.1/1.2 | 19.3/7.9 | 10.6/3.4 |
| NeoNav | 19.8/10.6 | 11.5/5.3 | 13.6/5.9 | 21 9/9.6 | 16.7/7.9 |
| Improved model | 20.7/11.1 | 11.2/55 | 14.8/7.0 | 22.6/10.1 | 17325/8.5 |

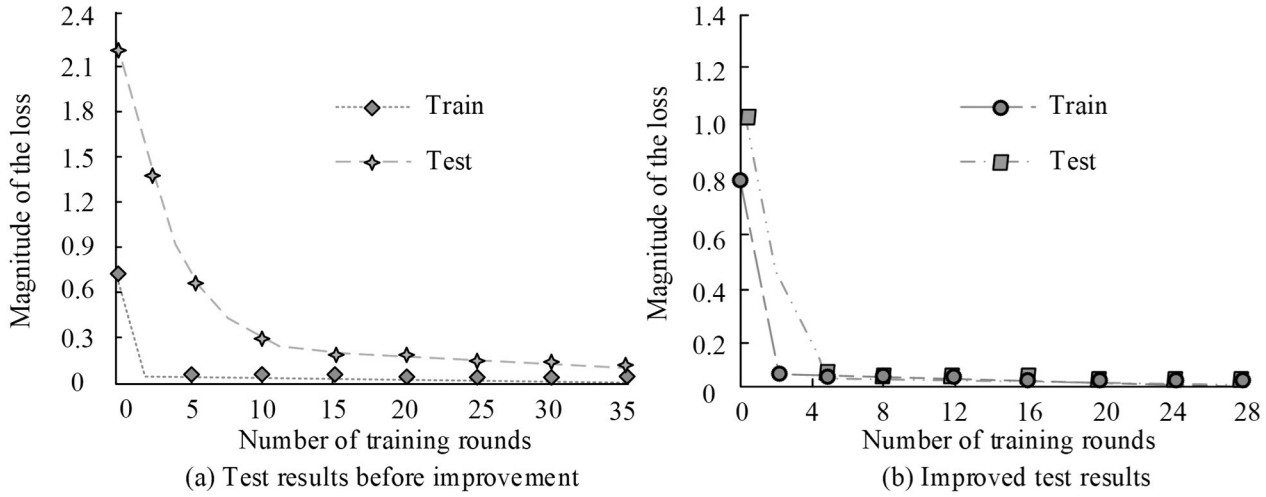



*Figure 9.* The loss value changes of the two models before and after improvement.

length. The results obtained from the improved model before and after the improvement are shown in Figure 10. From Figure 10(a), it can be observed that, with the exception of the Bathroom 02 scenario, which converges at around 5 million training frames, the other three scenarios in the improved model all converge at 9 million training frames. Moreover, there are differences in the corresponding convergence average trajectory lengths for the four scenarios. Figure 10(b) shows that the convergence of the improved model in all four scenarios occurs around 5 million training frames, and the average trajectory length converges around 10 steps [27]. In comparison, it is evident that the improved model converges faster, and the average trajectory length can converge to a better level, reducing it by about approximately 50%.

To further validate the effectiveness of the improved model, it was compared with the Baseline model, Long Short-Term Memory Navigation Model (LSTM Nav), and Hierarchical Asynchronous Universal Successor Representations (HAUSR) combined with hierarchical asynchronous universal subsequent feature representation [28]. The average trajectory length and average reward test results for the four models in the remaining 20 scenarios are shown in Figure 11.

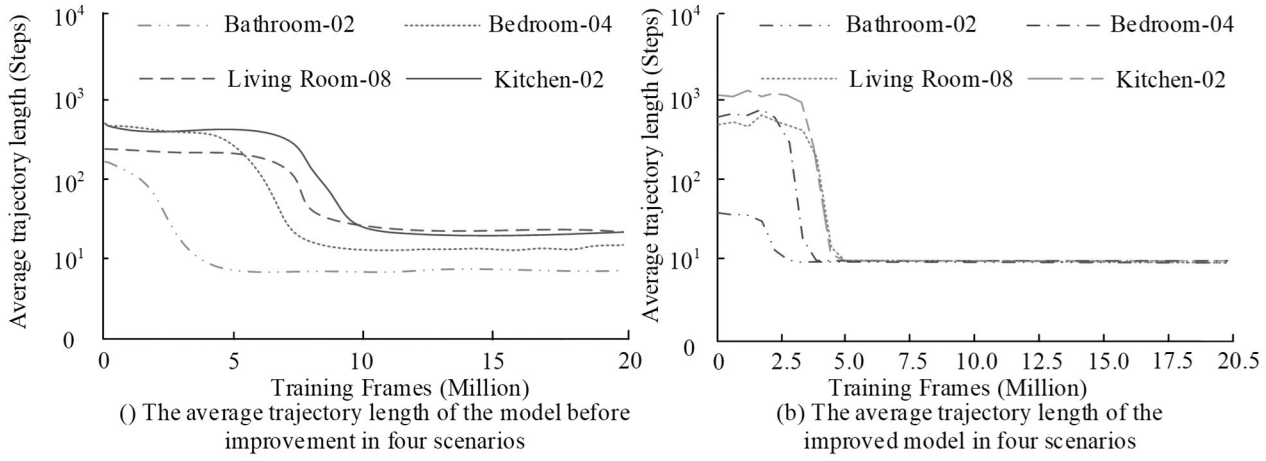



*Figure 10.* Test results of the model before and after improvement in four scenarios.

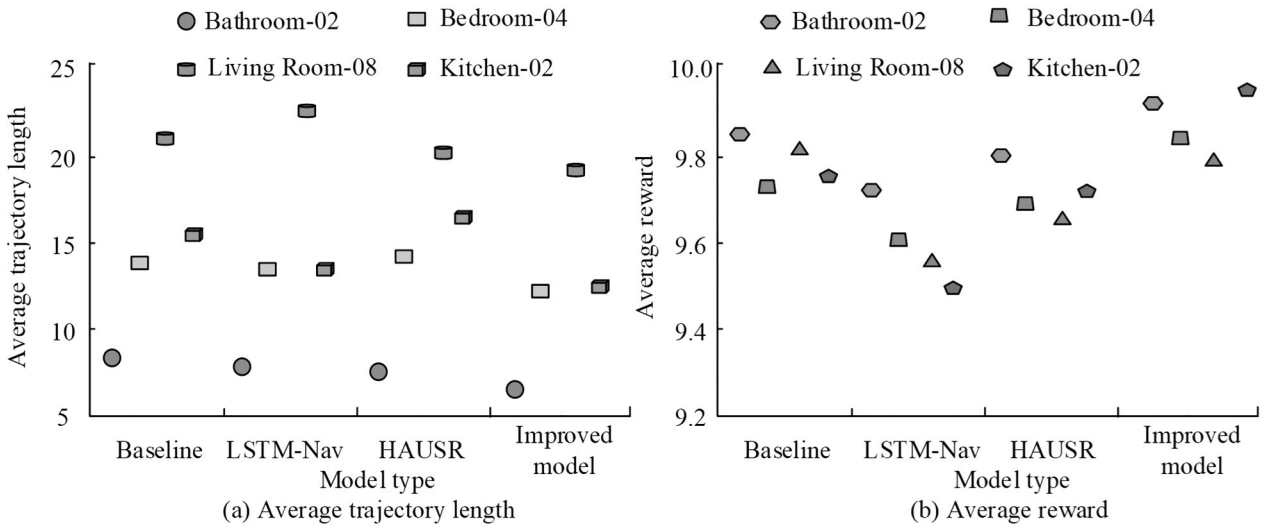



*Figure 11.* The average trajectory length and average reward test results of the four models in the remaining 20 scenarios.

From Figure 11(a), it can be observed that in the comparison of average trajectory lengths, the four models in the Bathroom02scenario are very similar. In the remaining scenarios, compared with Baseline, LSTM Nav, and HAUSR, the improved model has an average improvement of 8%, 5%, and 6%, showing better generalization performance. From Figure 11(b), it can be seen that in the comparison of average rewards, the improved model has varying degrees of leadership compared to the other three models. Among them, in the kitchen02 scenario, the improved model achieved a maximum improvement of about 0.4%, proving the good performance of the model.

Subsequently, the Active Vision Dataset (AVD) was selected for testing, which encompasses a large number of indoor scene images, including different rooms and scenes such as bedrooms, living rooms, kitchens, *etc.* These images were collected in a real-world settings, reflecting real-world objects and scenes. Additionally, a Gated Long Short Term Memory Asynchronous Advantage Actor Critic (GLA3C) visual navigation model combining Gated LSTM and A3C algorithms was added, and compared with HAUSR, NeoNav, and improved models [29].

The selected evaluation indicators are Navigation Success Rate (SR) and Path Length Weighted Success Rate (SPL), and the results are presented in Table 3. As indicated in Table 3, the success rates of the improved model in the four navigation goals of Exit, Refrigerator, Table, and Couch are 32.3%, 36.8%, 14.8%, and 12.6%, respectively. Moreover, the SPL indicator has improved by about 8% compared to NeoNav, indicating better generalization ability.

Finally, four scene categories, namely Bedroom, Bathroom, Living room, and Kitchen, were selected to compare the average collision rate and average consumption time of the four models. The results are shown in Figure 12. In Figure 12, the left side of the dashed line represents the collision rate, and the right side of the dashed line represents the time spent. From Figure 12, it can be observed that the collision rates of the four models in the Living room and Kitchen scenarios are all higher. Among them, the HAUSR model is as high as 0.33, the GLA3C model is around 0.25, and NeoNav is around 0.20. The highest value of the improved model is only 0.16, which is relatively low. Meanwhile, in the Bedroom and Bathroom scenarios, the collision rates of the other three models were all above 0.05, while the improved model had the lowest collision rate of only 0.01, indicating significantly better navigation performance. In terms of time consumption comparison, Living room and Kitchen scenes are longer, Bed-room and Bathroom scenes are shorter. The improved model takes up to 17 seconds and the shortest is about 8 seconds, which is more efficient and superior to the other three methods.

*Table 3*. Comparison of SR and SPL of variant models in the AVD dataset.

| Model type | Exit | Refrigerator | Table | Couch | Avg |
|---|---|---|---|---|---|
| HAUSR | 21.4/8.6 | 7.2/1.0 | 12.6/6.1 | 14.2/1.5 | 13.35/1.75 |
| GLA3C | 15.5/4.3 | 14.5/3.3 | 6.4/1.5 | 8.4/1.4 | 10.57/2.35 |
| NeoNav | 29.7/8.6 | 32.7/12.0 | 13.7/3.6 | 11.8/3.2 | 21.13/8.87 |
| Improved model | 32.3/9.5 | 36.8/13.25 | 14.8/3.9 | 12.6/3.4 | 21.89/6.22 |

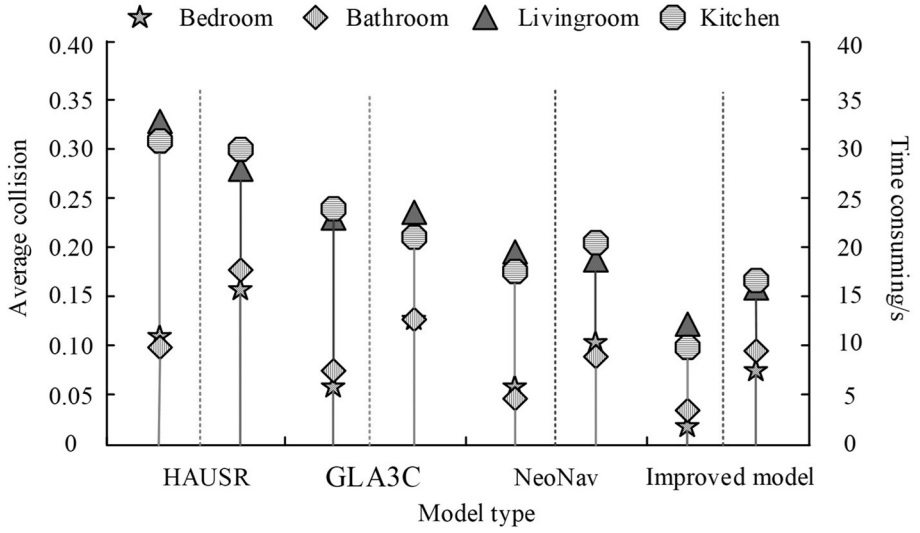



*Figure 12.* Comparison of consolidation rates and time consumption among four models.

To further verify the superiority of the proposed model, simulation experiments were carried out in more diverse environments and over longer periods of time. The comparison results with other models, conducted within 1000 steps, are presented in Table 4.

From the data given in Table 4, it is evident that the SR of the improved model is slightly lower than NeoNav for three of the objectives, while it significantly outperforms NeoNav for the other two objectives. The average SR over 1000 steps is improved by about 3% compared to NeoNav. SPL, on the other hand, is better than NeoNav in each target navigation process, with an improvement of about 7% compared to NeoNav in 1000 steps, which proves that the improved model can have better generalization ability when navigating to different targets, taking into account the success rate and the length of the path trajectory.

NeoNav, which has better navigation performance among the three models, was compared with the improved model. Figure 13 shows the navigation test results displayed in the first-person perspective and top view for four category scenarios. From Figure 13, it can be seen that within the time steps 1 to 8, the actions and routes of the mobile robot in the NeoNav model and the improved model are basically the same. In the 8th to 9th time steps, the NeoNav model waits in place for one time step before starting to move upwards, and due to its position against the wall, it is difficult to move forward and needs to turn to the right front to complete. Overall, the improved model navigation omitted two time steps and had fewer issues with collision steering, resulting in significantly better navigation performance.

*Table 4*. Comparison of SR and SPL of each model.

| Model type | Exit | Refrigerator | Table | Couch | Avg | Avg |
|---|---|---|---|---|---|---|
| HAUSR | 12.5/2.0 | 22. 3/2.7 | 15.3/1.6 | 7.3/1.1 | 13.8/1.8 | 14.24/1.84 |
| GLA3C | 6.3/1.5 | 16.4/4.4 | 83/1.2 | 14.6/3.2 | 6.7/0.9 | 10.46/2.24 |
| NeoNav | 12.6/3.5 | 30.8/8.7 | 10.7/3.0 | 34.8/12. 1 | 11.2/2.6 | 20.02/5.98 |
| Improved model | 13.7/3.4 | 31.2/9 2 | 11.5/3.3 | 35.6/12.13 | 11.6/2.84 | 20.72/6.1 7 |

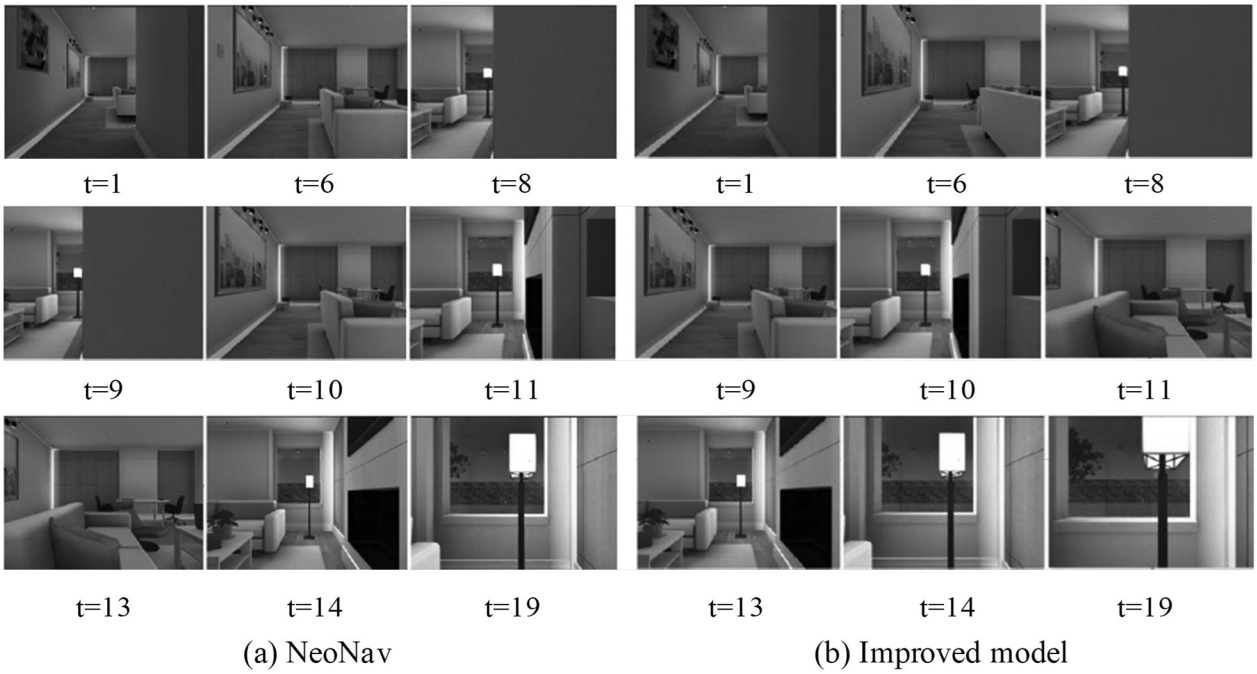



*Figure 13.* Comparison of Navigation Effects between NeoNav and Improved Models.

## 4. Discussion

This paper presents a novel approach to constructing a visual navigation network based on the Neo model. In order to address the limitations of existing navigation algorithms, the proposed model integrates segmentation attention, cross-connect methods, and an improved loss function.

To overcome the challenges posed by variations in object position, illumination, and volume during visual navigation, a cross-connection visual navigation network is proposed based on split attention. The model replaces ResNet50 with split attention-based ResNeSt50 for feature extraction of current and target states. The loss calculation method is also enhanced to improve overall navigation accuracy.

Experimental results show that the proposed model achieved a maximum improvement of approximately 0.4% in the kitchen02 scenario, verifying its superior performance. Moreover, in the AVD dataset, the success rates of the improved model in navigating Exit, Fridge, Table, and Couch targets are 32.3%, 36.8%, 14.8%, and 12.6%, respectively. The proposed research

model reduces the residual module of the basic network, utilizes shallow target feature information, and increases the network's receptive field. This helps to overcome the limitations of loss asymmetry caused by the reference frame, thus bringing the inference network closer to a true posterior. Additionally, the proposed method enables agents to make optimal decisions in the current environment, enhancing the network's performance and robustness.

## 5. Conclusion

The widespread application of intelligent robots in industries, services, and other fields underscores the critical need for these robots to efficiently and accurately navigate in dynamic environments. This study aims to enhance the visual navigation ability of mobile robots by constructing a visual navigation network based on the Neo model. This approach incorporates advancements in split-attention, cross-connection methods, and loss functions, enabling intelligent agents to extract essential information from input images.

The results show that in case of average trajectory length, the improved model exhibits a notably faster convergence rate in all four scenarios, reaching convergence at approximately 5 million training frames, with an average trajectory length of appoximately 10 steps. This marks a substantial reduction of around 50% compared to the pre-improvement state. When compared to Baseline, LSTM Nav, and HAUSR, the improved model has an average improvement of 8%, 5%, and 6%, respectively, showing superior generalization performance.

In terms of average rewards, compared with the other three models, the improved model has varying degrees of success. In the kitchen02 scenario, the improved model achieved a maximum improvement of about 0.4%, proving the good performance of the model. In the AVD dataset, the success rates of the improved model in the four navigation targets of Exit, Refrigerator, Table, and Couch were 32.3%, 36.8%, 14.8%, and 12.6%, respectively.

Notably, during real-world testing, the Neo Nav model and the improved model exhibit similar actions and routes for mobile robots. However, the improved model achieves a reduction of two time steps for navigation and has fewer issues with collision and turning. This demonstrates the effectiveness and better performance of the proposed method in the visual navigation of mobile robots.

The research methodology involves a reduction in the residual module within the original basic network and the adoption of a novel cross-connection method. These modifications enhance the network's capacity to leverage shallower target feature information, thereby increasing the network receptive field. Additionally, the method of loss calculation is improved to address the issue of loss asymmetry, which can be influenced by the reference frame. This adjustment brings the inference network closer to the real posterior, enabling the agent to make optimal decisions in the current environment. This improved performance extends to various scenes, bolstering the development of visual navigation technology.

At present, the main difficulty of goal-driven visual navigation lies in the generalization problem, which needs to be solved by making the agent understand the context relationship between the current environment and the target, transforming it into general knowledge. In dealing with some similar problems, past experience can be used, and in terms of information, the multi-modal fusion information can be used to extract the state of the current environment in order to perform navigation tasks more accurately. In the future, the application of the visual navigation algorithms within real-world robot systems will be considered, with an emphasis on modifying the model to adapt to the changes in the scene, and to also improve portability and universality of the model.

## Acknowledgement

This work is supported by 2021 Shaanxi Higher Education Teaching Reform Research Project (21BY184), the 13th Five-Year Plan Project of Shaanxi Provincial Department of Education (SGH18H530), and the Shaanxi Higher Education Teaching Reform Research Project (19BY33).

## References


[1] J. Kulhánek *et al.*, "Visual Navigation in Real-World Indoor Envi-ronments Using End-to-End Deep Reinforcement Learning", *IEEE Robotics and Auto-mation Letters*, vol. 6, no. 3, pp. 4345–4352, 2021. http://dx.doi.org/10.1109/LRA.2021.3068106

[2] S. D. Morad *et al.*, "Embodied Visual Navigation With Automatic Curriculum Learning in Real Environments", IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 683–690, 2021. http://dx.doi.org/10.1109/LRA.2020.3048662

[3] Y. Fang *et al.*, "ST-SIGMA: Spatio-temporal Semantics and Interaction Graph Aggregation for Multi-agent Perception and Trajectory Forecasting", CAAI Transactions on Intelligence Technology, vol. 7, no. 4, pp. 744–757, 2022. http://dx.doi.org/10.1049/cit2.12145

[4] Q. Wu *et al.*, "Reinforcement Learning-Based Visual Navigation With Information-Theoretic Regularization", *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 731–738, 2021. http://dx.doi.org/10.1109/LRA.2020.3048668

[5] Y. Lyu *et al.*, "Improving Target-driven Visual Navigation with Attention on 3D Spatial Relationships", *Neural Processing Letters*, vol. 54, no. 5, pp. 3979–3998, 2022. http://dx.doi.org/10.1007/s11063-022-10796-8

[6] V. Tolani *et al.*, "Visual Navigation Among Humans With Optimal Control as a Supervisor", *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2288–2295, 2021. http://dx.doi.org/10.1109/LRA.2021.3060638

[7] Y. Zhao *et al.*, "Robot Visual Navigation Estimation and Target Localization Based on Neural Network", *Paladyn, Journal of Behavioral Robotics*, vol. 13, no. 1, pp. 76–83, 2022. http://dx.doi.org/10.1515/pjbr-2022-0005

[8] Q. Fang *et al.*, "Target-driven Visual Navigation in Indoor Scenes using Reinforcement Learning and Imitation Learning", *CAAI Transactions on Intelligence Technology*, vol. 7, no. 2, pp. 167–176, 2022. http://dx.doi.org/10.1049/cit2.12043

[9] T. Guan *et al.*, "GA-Nav: Efficient Terrain Segmentation for Robot Navigation in Unstructured Outdoor Environments", *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8138–8145, 2022. http://dx.doi.org/10.1109/LRA.2022.3187278

[10] H. Sang *et al.*, "A Novel Neural Multi-Store Memory Network for Autonomous Visual Navigation in Unknown Environment", *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2039–2046, 2022. http://dx.doi.org/10.1109/LRA.2022.3140795

[11] Y. Lyu *et al.*, "Improving Target-driven Visual Navigation with Attention on 3D Spatial Relationships", *Neural Processing Letters*, vol. 54, no. 5, pp. 3979–3998, 2022. http://dx.doi.org/10.1007/s11063-022-10796-8

[12] Z. Zhou *et al.*, "Robot Navigation in a Crowd by Integrating Deep Reinforcement Learning and Online Planning", *Applied Intelligence*, vol. 52, no. 13, pp. 15600–15616, 2022. http://dx.doi.org/10.1007/s10489-022-03191-2

[13] J. Zan, "Research on Robot Path Perception and Optimization Technology Based on Whale Optimization Algorithm", *Journal of Computational and Cognitive Engineering*, vol. 1, no. 4, pp. 201–208, 2022. http://dx.doi.org/10.47852/bonviewJCCE597820205514

[14] P. Naveen *et al.*, "Improving Chatbot Performance using a Hybrid Deep Learning Approach", *Journal of System and Management Sciences*, vol. 13, no. 3, pp. 505–516, 2023. http://dx.doi.org/10.33168/jsms.2023.0334

[15] A. S. Maihulla *et al.*, "Reliability and Performance Analysis of a Series-parallel System using Gumbel - Hougaard Family Copula", *Journal of Computational and Cognitive Engineering*, vol. 1, no. 2, pp. 74–82, 2022. http://dx.doi.org/10.47852/bonviewJCCE2022010101

[16] H. Karnan *et al.*, "Socially CompliAnt Navigation Dataset (SCAND): A Large-Scale Da-taset of Demonstrations for Social Navigation", *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11807–11814, 2022. http://dx.doi.org/10.1109/LRA.2022.3184025

[17] X. Xiao *et al.*, "Motion Planning and Control for Mobile Robot Navigation using Machine Learning: A Survey", *Autonomous Robots*, vol. 46, no. 5, pp. 569–597, 2022. http://dx.doi.org/10.1007/s10514-022-10039-8

[18] S. Fountas *et al.*, "AI-assisted Vision for Agricultural Robots", *AgriEngineering*, vol. 4, no. 3, pp. 674–694, 2022. http://dx.doi.org/10.3390/agriengineering4030043

[19] S. H. Yang *et al.*, "A Guideline for Personal Service Robot Interface Design", *Journal of Logistics, Informatics and Service Science*, vol. 7, no. 2, pp. 127–140, 2020. http://dx.doi.org/10.33168/JLISS.2020.0209

[20] K. Kim *et al.*, "Virtual Testbed for Monocular Visual Navigation of Small Unmanned Aircraft Systems", *Journal of Defense Modeling and Simulation*, vol. 19, no. 3, pp. 433–451, 2022. http://dx.doi.org/10.1177/1548512920954545

[21] L. C. Hong *et al.*, "Activities of Daily Living Recognition using Deep Learning Approaches", *Journal of Logistics, Informatics and Service Science*, vol. 9, no. 4, pp. 129–148, 2022. http://dx.doi.org/10.33168/LISS.2022.0410

[22] L. Song *et al.*, "Dense Face Network: A Dense Face Detector Based on Global Context and Visual Attention Mechanism", *Machine Intelligence Research*, vol. 19, no. 3, pp. 247–256, 2022. http://dx.doi.org/10.1007/s11633-022-1327-2

[23] C. H. Lim *et al.*, "Activities of Daily Living Recognition Us-ing Deep Learning Approaches", *Journal of Logistics, Informatics and Service Science*, vol. 9, no. 4, pp. 129–148, 2022. http://dx.doi.org/10.33168/LISS.2022.0410

[24] Y. Li *et al.*, "Temporal Pyramid Network With Spatial-Temporal Attention for Pedestrian Trajectory Prediction", *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1006–1019, 2022. http://dx.doi.org/10.1109/TNSE.2021.3065019

[25] S. E. Chung and H. Y. Ryoo, "Gesture Design Attribute and Level Value of Social Robot: A User Experience Based Study", *Journal of System and Management Sciences*, vol. 10, no. 2, pp. 108–121, 2020. http://dx.doi.org/10.33168/JSMS.2020.0208

[26] L. Lin *et al.*, "Vehicle Trajectory Prediction Using LSTMs With Spa-tial–Temporal Attention Mechanisms", *IEEE Intelligent Transportation Systems Maga-zine*, vol. 14, no. 2, pp. 197–208, 2022. http://dx.doi.org/10.1109/MITS.2021.3049404

[27] J. Qiu *et al.*, "Egocentric Human Trajectory Forecasting With a Wearable Camera and Multi-Modal Fusion", *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8799–8806, 2022. http://dx.doi.org/10.1109/LRA.2022.3188101

[28] L. Yangsun, "A Study on Abnormal Behavior Detection in CCTV Images through the Supervised Learning Model of Deep Learning", *Journal of Logistics, Informatics and Service Science*, vol. 9, no. 2, pp. 196–209, 2022. http://dx.doi.org/10.33168/LISS.2022.0212

[29] C. Zhang, "Intelligent Robot Path Planning and Navigation based on Reinforcement Learning and Adaptive Control", *Journal of Logistics, Informatics and Service Science*, vol. 10, no. 3, pp. 235–248, 2023. http://dx.doi.org/10.33168/JLISS.2023.0318

*Received:* August 2023
*Revised:* October 2023
*Accepted:* October 2023

*Contact addresses*:
Wei Yu*
Xi'an Siyuan University
Xi'an
Shaanxi
China
e-mail: yuwei090117@126.com
*Corresponding author

Xinzhi Tian
Xi'an Siyuan University
Xi'an
Shaanxi
China
e-mail: litterrobot@163.com

Wei Yu is a lecturer at the Foundation Department of Xi'an Siyuan College. She obtained an MSc degree from Xi'an University of Architecture and Technology, China, in 2007. Her research interests include the Internet of Things, software development, artificial intelligence, and machine learning.

Xinzhi Tian is a full professor at the Department of Computer Science and Technology, School of Electronic Information Engineering, Xi'an Siyuan University, China. He obtained an MSc degree from Hebei University of Engineering, China, in 2006. His research interests include the Internet of Things, software development, artificial intelligence, and machine learning.