RewardVLN: An Improved Agent Navigation Based On Visual-Instruction Alignment*

Ruonan Liu, Ping Kong, Shuai Wu, and Weidong Zhang

***Abstract*— Vision-and-language Navigation (VLN) is a challenging problem that requires agents to follow natural language instructions in a photo-realistic environment. The alignment between visual object information and instruction object information is critical for the navigational capabilities of intelligent agents. However, most reinforcement learning policies primarily focus on the agent's distance change to the target viewpoint as the direct reward after taking an action, with object information playing a minor role in classical reinforcement learning for VLN. To address this limitation, we construct a new reward shaping that incorporates both the changes in the agent's distance to the target and the progress made in navigating according to the given instruction. To capture the navigation progress, we propose an object alignment method that aligns the visual object information observed by the agent with the object information specified in the instructions. By leveraging the object's position within the navigation instruction, we estimate the agent's approximate progress during navigation. Experimental results demonstrate the effectiveness of our approach in reducing the navigation error (NE) and achieving high performance in terms of the success rate weighted by path length (SPL). Our method significantly enhances the agent's ability to accurately follow natural language instructions to reach the intended destination, while also exhibiting improved generalization in unseen environments.**

I. Introduction

Training an intelligent robot agent to follow human natural language instructions constitutes a long-term and complex task. A great variety of vision-and-language navigation (VLN) studies [1]–[4] have been introduced to explore this area. These research efforts have significantly contributed to the ultimate goal of enabling robots to understand and comply with human language instructions. Unlike static visual language tasks such as visual question answering [5], the VLN task entails dynamic challenges in recognizing and interacting with a changing environment during navigation. VLN integrates various sub-tasks, including understanding natural language instructions, perceiving the visual world, implementing navigational behaviors to reach the target location, and so on. Several datasets have been proposed to investigate this task, the most diverse and widely used is the R2R dataset [6] based on the Matterport3D simulator [7]. This dataset comprises 10,800 panoramas from 90 building-scale scenes. R2R has become a significant benchmark, and subsequent datasets have been further extended based on it, such as the Room-for-Room dataset for long trajectories [8] and the RxR dataset containing multilingual instructions [9]. Additionally, more complex outdoor environment datasets, such as the street view dataset Touchdown [10], have also been developed.

TABLE I
Acronyms and Their Definitions

Abbreviations	Corresponding meanings
TL	The trajectory length (in meters) of the agent navigation path
NE	The average distance (in meters) between the agent's final location and the target
SR	The rate at which the agent stops 3 meters within the target range
SPL	The success rate weighted by the normalized inverse of the trajectory length

Numerous models have been proposed for the R2R navigation task [6]. Given natural language instructions, the agent travel through different rooms or floors, explores a photo-realistic 3D environment, and eventually stops at the destination. The research conducted in the Matterport3D simulator [7] serve as a bridge for realizing the sim-to-real transfer to robot natural language navigation [11].

The evaluation of VLN is very straightforward. The execution route by the agent is considered as a success when the agent's final position is less than a certain distance from the target position (here, we set it to 3m). Generally speaking, we only need the following evaluation metrics: Success Rate (SR), Trajectory Length (TL), Success weighted by Path Length (SPL), and Navigation Error (NE). See Table I for details.

For VLN tasks, the majority of existing approaches employ a combination of imitation learning (IL) and reinforcement learning (RL) to train the agent [2], [12]–[15]. All these studies contribute to improved agent performance in navigation tasks. However, the above models that mix IL and RL do not incorporate crucial object information into the reward. In reinforcement learning, the direct reward primarily relies on the distance change between the agent and the destination, while the object information plays a limited role. We propose incorporating object information into reward shaping, hoping that object information, such

*Resrach partly supported by the National Natural Science Foundation of China and Tianjin Applied Basic Research Project.

Ruonan Liu is with Department of Automation, Shanghai Jiao Tong University, Shanghai, China (e-mail: ruonan.liu@sjtu.edu.cn).

Ping Kong is with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: kongping@tju.edu.cn).

Shuai Wu is with the College of Intelligence and Computing, Tianjin University, Tianjin 300350, China (e-mail: ws666@tju.edu.cn).

Weidong Zhang is with Department of Automation, Shanghai Jiao Tong University, Shanghai, China (corresponding author. phone: 13023173666; e-mail: wdzhang@sjtu.edu.cn).

as iconic landmarks, can be leveraged during the stage of reinforcement learning to assist the agent in finding the optimal path.

In the ablation studies, we compared the model performance across three variants: using new reward shaping in all training stages, applying new reward shaping in partial stages, and not employing new reward shaping in any stages. The results of the ablation experiment demonstrate that the method we propose achieves the best performance in terms of SR, SPL, and NE in the unseen environment. When the new reward shaping is not used in the second training stage, the SPL in the validation unseen split is only 0.513, while the NE reaches 4.636. If no new reward shaping is used in both training stages, SPL and NE are 0.519 and 4.701, respectively. These two models perform inferior to the full model. Overall, the new reward shaping approach enhances the agent's performance to a certain extent and improves its ability to accurately follow natural language instructions to reach the correct destination.

In summary, the contributions of our work are as follows:

- A simple method is proposed to align the visually observed objects with corresponding objects in the instruction. At the time of acquiring observation in each step t, the rough progress value is obtained at the same time based on the object's position in the instruction, thereby assisting the agent in following the path specified by natural language instruction during navigation.
- The corresponding relationship between visual object information and instruction object information is incorporated into reward shaping, allowing symbolic landmark information to play a role in reinforcement learning. Compared to reward shaping without object information, our proposed method reduces NE by approximately 3% while maintaining similar SPL. This indicates that our proposed method enhances the accuracy of the agent navigation path and its ability to reach the target viewpoint.

The rest of the article is organized as follows. Section II details the related work. Problem formulation and the proposed RewardVLN are discussed in Section III. In Section IV, the effectiveness of the proposed method is validated on the VLN dataset. Finally, Section V concludes our work and discusses potential avenues for future research.

II. Related Work

The research field of visual language navigation is currently in a booming stage, and because of its wide application and great practical value, it has triggered the continuous innovation and research of many experts and scholars in the field. In order to make the model more generalized when applied to the actual situation and improve the performance of the model, a hybrid reinforcement learning method based on Model-free and Model-based [12] has been proposed, and the results prove that the model obtained by using this method also has a good transfer ability for the unseen environment. Speaker-Follower Models [1] can greatly improve the performance of their models through data augmentation and semantic reasoning in the panoramic action space.

Although the BERT model has made some achievements in the VLN field, it is precisely because the original BERT architecture cannot adapt well to the needs of the VLN field, so a recurrent BERT model for VLN is proposed [2]. In this paper [16], BnB, a large-scale intra-domain pretraining dataset, is introduced for AirBert pretraining. At the same time, the introduction and application of shuffling loss also lead to higher generalization performance of the whole model. In the course of agent navigation, previously visited locations and actions taken affect subsequent states. To make better use of this information, it makes sense to introduce a History Aware Multimodal Transformer (HAMT) [17], which enables the historical information to participate in multimodal decisions and then fine-tune navigation strategies with reinforcement learning. Episodic Transformer (E.T.) [18], a new visual-language navigation architecture, uses a multimodal transformer to encode for better navigation performance in the environment; For the input images of the agent, most processing is to pre-train the image encoder on the ImageNet [19] and then encode the image information.

However, for the structural information in the scene, even though it is crucial to the target navigation task, pretraining on the ImageNet cannot properly encode. So, Structure-Encoding Auxiliary Tasks (SEA) [4] came into being, which uses the dynamic data in the navigation process environment for pre-training and improving the image encoder; Jialu Li et al. proposed the ENVEDIT method [3], which is different from previous studies. This method focuses on the environment and creates a new environment suitable for agents by changing the existing environment that agents are already familiar with to train more general agents. Dual-Scale Graph Transformer (DUET) [20] can be used to construct a topological map, which uses graph transformers and combines fine-scale and coarse-scale encodings for dynamic navigation planning.

For the problem that agents sometimes get stuck in long instructions or ignore short instructions, a model-agnostic milestone-based task tracker (M-TRACK) [21] is used to guide agents and monitor their progress. Then the agent will execute the marking instruction according to the progress of the current milestone, thus solving the problem to some extent. In the paper [13], the author migrated Graph neural network to the VLN field and proposed the Entity Relationship Graph method, thus improving the performance of the agent in the unseen environment. Different from previous work, we focus on the changes in the agent's reinforcement learning in the navigation environment when we implement the Graph method, hoping to make the agent's reinforcement learning more instructive through Reward Shaping.

III. Problem Overview and Modelling

A. Problem Overview

For the Room-to-Room dataset, the VLN problem can be described as follows:

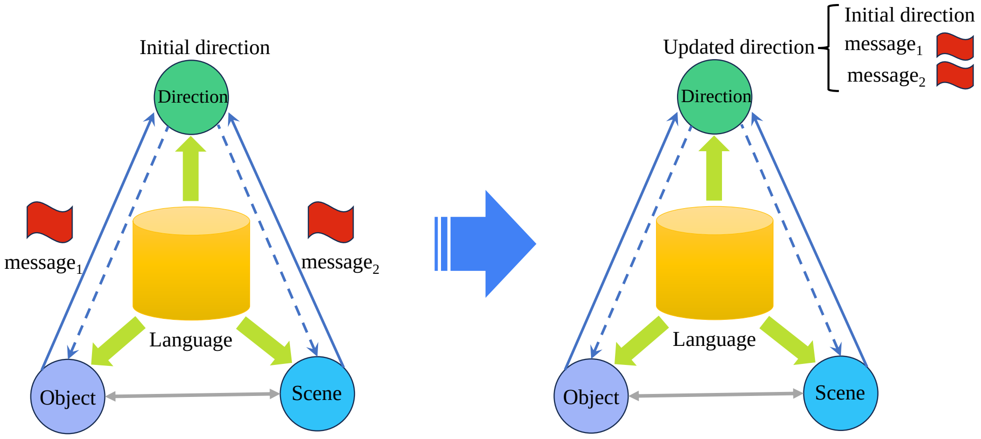


Fig. 1. Model overview: RewardVLN is based on graph neural networks, where nodes within the graph are categorized into three distinct types: scene-related, object-related, and direction-related. The edges connecting these nodes serve as conduits for information exchange and message propagation.

First of all, the agent is given an instruction. The attributes among the words may be different, for example, some are about instruction action and some are about object observation, but in a word, these words and the relationships among the words can give the agent some direction. At each time step t, the agent receives a panoramic view input about its surroundings. It is worth noting that each panoramic view contains 36 single-view images, and within each single-view image, the agent has n candidate navigation directions (n depends on the location of the agent). Each selection of the next navigation viewpoint will have an impact on the future actions and the observed visual images, so the agent needs to have a certain global awareness when choosing each step to better complete the task.

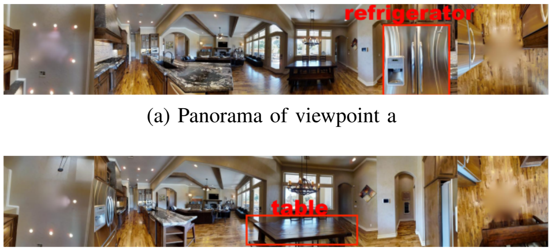
In order to facilitate simulation and calculation, we can abstract the navigation map as the "graph structure" in the data structure, and the navigation point of the agent can correspond to the vertex in the connected graph. The task of the agent is to move from the source point to the target point on a connected graph according to the received instructions and the observed view. How to effectively guide the agent and obtain better evaluation index values (such as higher success rate, lower navigation error, etc., these indicators will be discussed in detail in the following part) is the main problem to solve. For such a task, it seems relatively simple, but we notice that in the current VLN leaderboard, due to the complexity and change of the actual situation, the SPL of humans to complete such a task is only 0.76, while for machine agents, it is still difficult to complete such a task under today's technological development because it doesn't have rich prior knowledge, good perception and inference ability, high strength computing ability and other comprehensive qualities like human beings.

B. Models on Graph Neural Networks

In this experiment, we mainly use the model based on a graph neural network. In this model, we give different meanings to the nodes and edges in the graph, which plays an important role in the actual navigation process. For a given instruction, first of all, we can divide nodes in the graph into three categories, the first category is scene related, the second type is object related, and the third type is related to the direction. Secondly, in order to show the relationship between nodes of different types and those of the same type and to fully express the complete semantics of the instruction, we will carry out a special meaning analysis of edges in the graph. For the nodes related to the scene and object mentioned above, the edge between them can be expressed as $Edge_{so}$, which clearly corresponds to the meaning of the short instruction containing these two types of nodes in the instruction. Since the graph contains a lot of nodes and edges, its connection structure can express the meaning of various instructions without ambiguity, thus enabling the agent to have a higher accuracy.

The above is just the static structure transformation from the natural language instruction to the instruction that the agent can understand. In the process of agent navigation, the agent continuously obtains information from the instruction and is guided to perform the next action to reach the next state. Therefore, the graph transformation is a very key link in the whole dynamic change. The dynamic process of the graph can be roughly divided into four steps, namely node initialization, information passing, node update, and action prediction. Among them, the function of node initialization is to establish the first-level text-visual connection and prepare for the subsequent information passing. In the process of information passing, each node can not only receive information from other nodes but also send information to other nodes. Since edges in the graph are relational, they are passed through edges in the graph, which is also the process of establishing the second level of text-visual connection. After the information is passed, the node needs to update the data. Simply, the received information can be added up directly here. Finally, we need to perform a learnable mathematical mapping and multi-class prediction of actions through the Softmax function.

Based on the execution of the entire model, we visualize the abstract process, as shown in Fig. 1.



(a) Panorama of viewpoint a

(b) Panorama of viewpoint b

0: “Walk into the hallway and through the entrance to the kitchen area. Walk Passed the sink and stove area and stop between the refrigerator and dining table.”
1: “Walk through the kitchen. Go past the sink and stove stand in front of the dining table on the bench side.”
2: “Walk into the kitchen. Walk past the refrigerator. Stop directly in front of the wooden table.”

Fig. 2. An example of object alignment: The images above are panoramas from different viewpoints, while the corresponding instructions are presented below.

C. *Object Alignment*

During the navigation process of the agent, according to the natural language instruction, object information will appear in the instruction, which is of great significance as a landmark. The agent obtains observation at each step t, and the visual object information obtained will be matched with the object information in the instruction to obtain the position of the object information in the instruction. The position will be used to evaluate the progress of the agent to complete the navigation, as shown in Fig. 2, these three instructions are different expressions of the same path. In 2a panorama view, the agent will recognize the refrigerator landmark. In 2b panorama view, the agent will recognize the table landmark, and in the instruction, the position of the table object in the instruction is behind the position of the refrigerator in the instruction. Then, during the agent navigation process that implements this instruction, the observation progress in 2a panorama view will be less than the observation progress in 2b panorama view. Each time the agent makes an action, the progress difference of the observation before and after the action will participate in the training of the agent as part of reward shaping.

The agent first obtains all the object embeddings $O_1, O_2, ..., O_n$ in the observation from the obtained observation v_i, next, the agent decodes these object embeddings into the corresponding object $w_1, w_2, ..., w_n$, then, the agent matches these objects in the instruction, and finally obtains the corresponding positions of different objects in the instruction sentence $i_1, i_2, ..., i_n$, we take the maximum value as the progress of the observation.

$$\begin{aligned} o_j &= f\left(v_i\right), \quad j = 1, 2, \ldots, n \\ w_k &= L\left(o_k\right), \quad k = 1, 2, \ldots, n \\ i_k &= g_{\text{ins}}\left(w_k\right), \quad k = 1, 2, \ldots, n \\ p &= \max\left(i_1, i_2, \ldots, i_n\right) \end{aligned} \tag{1}$$

where L indicates linear mapping, f indicates a mapping from observation to object encoding, and g_{ins} represents a mapping from object w_k to object location i_k in the instruction.

D. *Reward Shaping*

Choose the right route and finally reach the right destination according to the given natural language instructions in the unknown environment. According to human habits, the most important thing in the navigation process is the feedback of the surrounding environment, namely the corresponding recognition of iconic landmarks. We hope that robot agents can also have this ability. That is, the agent can align the landmarks in the instruction with the objects it is observing at a certain point of view and use these landmarks to help the agent find the correct route. At each step t, the agent obtains the observation of the current viewpoint, obtains the candidate set of the next step and the list of objects currently recognized, and compares the recognized objects with the objects in the instruction, which reflects the completion progress of the agent's navigation according to this instruction to a certain extent. Therefore, we propose a simple method of alignment object, which extracts the object in the instruction and aligns it with the object observed visually, obtains the rough progress value of this observation, and then we apply it to reward shaping for reinforcement learning in the training process. We redesigned the direct reward after each action made by the agent and combined the change of distance from the target point with the change of progress in the navigation process of executing the instruction, hoping to ensure the accuracy of the route while reaching the final goal.

At every time step t, a direct reward will be generated after the agent makes an action. In our design, this reward includes two parts: the change of the distance $(d_{t-1} - d_t)$. And the change of the progress $(p_t - p_{t-1})$. In order to express brevity, we did not add the subscript representing time step t:

$$\begin{aligned} \Delta d &= d_{t-1} - d_t \\ \Delta p &= p_t - p_{t-1} \\ \gamma_1 &= R_{d2r}(\Delta d) \\ \gamma_2 &= R_{p2r}(\Delta p) \\ \gamma &= \gamma_1 + \gamma_2 \end{aligned} \tag{2}$$

where γ represents the direct reward for each step t, and Δd and Δp represent the change in distance and progress, respectively, and the function R_{d2r} and the function R_{p2r} respectively represent the reward generated by the distance change and the reward generated by the progress change. It is worth noting that in the process of reinforcement learning, the agent may continuously perform an action to obtain a reward, thus resulting in the so-called “reward cycle” [22]. Therefore, when designing rewards, no matter for rewards defined by distance, or for rewards defined by progress, when positive rewards are given to the agent under certain circumstances, some negative rewards will be correspondingly given in other circumstances so as to avoid the situation that the

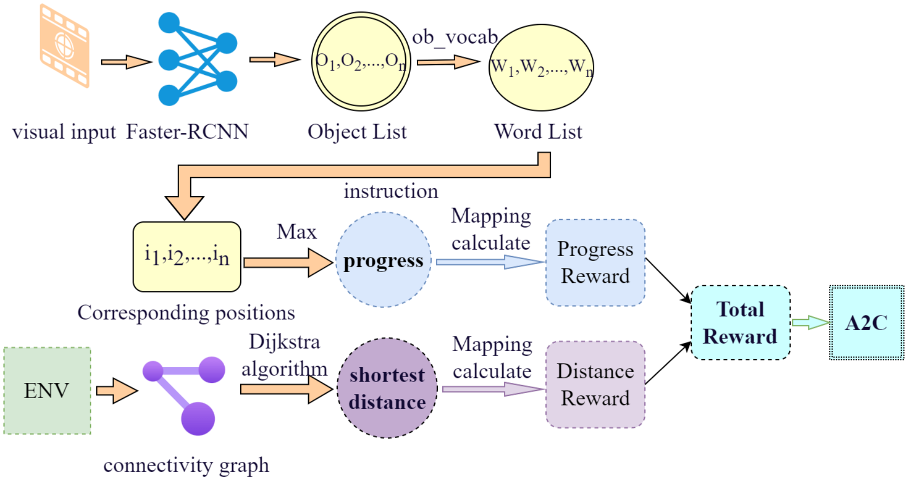


Fig. 3. New reward shaping: The estimation of progress is achieved by aligning the visual object information observed by the agent with the object information provided in the instructions. The new proposed reward shaping incorporates both the changes in the agent's distance to the target and the progress made in navigation based on the given instruction.

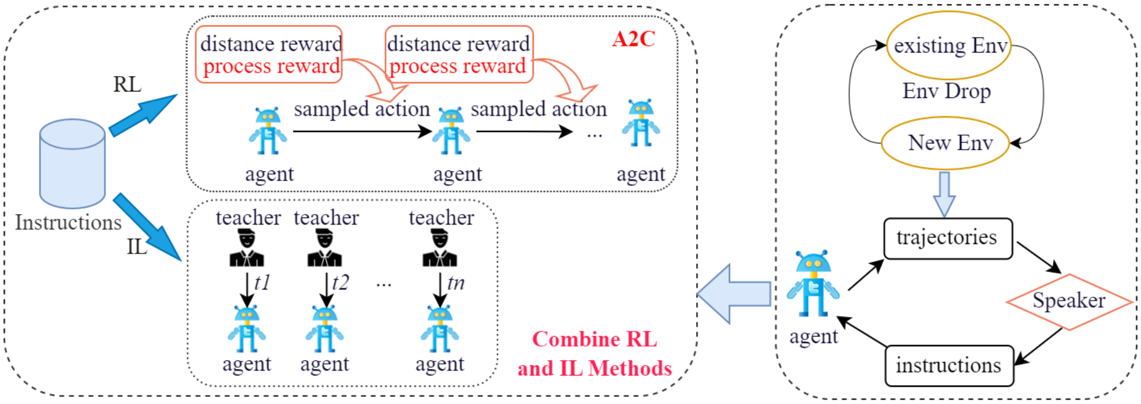


Fig. 4. Overall architecture: Firstly, new environments and new instructions are generated to augment data. Subsequently, the entire dataset is employed in a training process that combines reinforcement learning and imitation learning.

agent keeps repeating an action in order to obtain rewards. The flow of the RL phase is shown in Fig. 3.

Through such a design, we hope to make the agent get closer to the target at every step while ensuring the accuracy of the route as far as possible. The experimental results show that, compared with the performance of [13] in the test split, SR and NE both perform better when SPL is basically unchanged. This shows that our method has been improved in the accuracy of reaching the correct destination.

E. Reinforcement Learning and Imitation Learning

The training process of the experiment is mainly made of Reinforcement Learning (RL) and Imitation Learning (IL) for mixed training. To make the learning methods of the two more effective in guiding the agents to the next action. Therefore, imitation learning is coordinated to a certain extent. In form, here is:

$$L_{\text{total}} = \mu L_{IL} + L_{RL} \tag{3}$$

where μ is the loss coefficient of imitation learning.

The overall architecture is shown in Fig. 4. First, the EnvDrop method [14] is used to generate a new environment from the existing environment, then the speaker model generates new instructions from the new environment. All environments and instructions are used in the training process of the agent. For each instruction, the agent is trained by combining RL and IL. In the RL phase, the agent receives a direct reward at each step, consisting of the distance reward and the process reward, which is used in the subsequent A2C. In the IL phase, the agent receives a teacher action at each step. A mixture of IL and RL training is used to combine the advantages of both to achieve better performance.

IV. Experiments and Results

A. Experimental Setup

The computer configuration parameters of this experiment are: GPU: RTX2080TI, a total of 11.7GB of video memory; CPU: 6-core E5-2680v4; Memory: 30.1GB; Hard drive: 451.0GB. Meanwhile, the whole experiment is carried out in the Matterport3D Simulator, which requires the corresponding simulator environment to be configured on the server.

The dataset of the experiment is mainly Room-to-Room (R2R). The dataset contains 21,567 words of navigation instructions with an average sentence length of 29 words. Meanwhile, we divided the dataset into four parts: train, validation seen, validation unseen, and test unseen sets. In order to reflect the generalization of the model, we test the trained agent with new instructions in a new environment and check how well it performs.

Among them, SPL is the relatively important indicator in the R2R dataset because it can measure the accuracy and efficiency of navigation at the same time. Therefore, in this experiment, we need to pay attention to changes in this indicator to a certain extent.

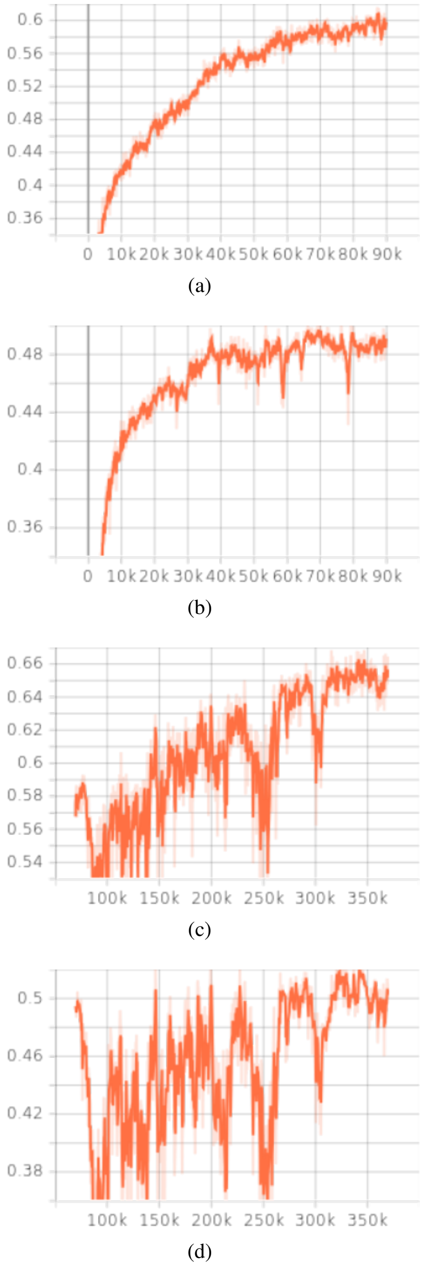
B. Training

The experiment is divided into two stages. The first stage uses data in train split to train, and the number of iterations is set to 90,000; The second stage uses the model with the best performance on the validation unseen spilt in the first stage, uses the enhanced data [14] and data in train split to train together for fine-tuning, and the number of iterations is set to 300,000.

C. Experimental Result

There are three experimental Settings: single run, beam search, and pre-exploration. Single run is the most common one, which can most intuitively show the agent's performance, navigation efficiency, and generalization in an unprecedented environment. We tested the agent's performance on the single-run evaluation setting. Because of the difference in data and computational power, we did not reach the state-of-the-art. However, with the same dataset and experimental setup, our proposed method showed a performance improvement over the baseline method Lang-Vis-Entity. In the following description, we use the model that performed best on the validation unseen split during training compared to the baseline model.

As shown in Table II, our model slightly outperforms the baseline method on NE, SR, and SPL on the validation seen split. It slightly underperforms the baseline method on both SR and SPL on the validation unseen dataset and performs better on NE, 3% lower than the baseline method. On the unseen data segmentation set test unseen split, the SPL of the two is not much different, but our model outperforms the baseline method on both NE and SR. In conclusion, our proposed method of applying object information to reward shaping performs better than the baseline method Lang-Vis-Entity. In our model, SR on the two unseen splits is basically the same, and so is SPL, which indicates that our method can generalize better in unseen environments. The proposed method has made greater progress in reducing NE, which shows that the improvement we have made in reward shaping makes the end position of the agent closer to the target. The possible reason is that the target point in the instruction usually has an obvious landmark object information. In our method, object information is integrated into reward shaping, so that object information in the instruction can be better utilized, making the navigation of the agent more accurate and its performance improved.



(a)

(b)

(c)

(d)

Fig. 5. SPL in training iterative process, (a)The first stage, val_seen, (b)The first stage, val_unseen, (c)The second stage, val_seen, (d)The second stage, val_unseen

For the change in SPL in val_seen and val_unseen environments during the two training phases, see Fig. 5.

As can be seen from Fig. 5, in the first stage, only the training data participated in the training, and SPL gradually increased with the increase of the number of training iterations. In the second stage, with the increase of the number of training iterations, the SPL presented a spiral change, the oscillation was violent in the early stage of training and tended to be stable at the end of training. We guessed that

TABLE II

COMPARISON OF SINGLE-RUN PERFORMANCE WITH THE PREVIOUS METHODS ON R2R

Model	Validation Seen				Validation Unseen				Test Unseen			
	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
Random	9.58	9.45	0.16	-	9.77	9.23	0.16	-	9.89	9.79	0.13	0.12
Human	-	-	-	-	-	-	-	-	11.85	1.61	0.86	0.76
Seq-to-Seq	11.33	6.01	0.39	-	8.39	7.81	0.22	-	8.13	7.85	0.20	0.18
Speaker-Follower	-	3.36	0.66	-	-	6.62	0.35	-	14.82	6.62	0.35	0.28
EnvDrop	11.00	3.99	0.62	0.59	10.70	5.22	0.52	0.48	11.66	5.23	0.51	0.47
Lang-Vis-Entity	10.13	3.47	0.67	0.65	9.99	4.73	**0.57**	**0.53**	1.29	4.75	0.55	**0.52**
RewardVLN	10.32	**3.41**	**0.68**	**0.66**	11.85	**4.57**	0.56	0.52	12.19	**4.65**	**0.56**	**0.52**

TABLE III

ABLATION STUDY

Model	Stage1	Stage2	Validation Unseen				Test Unseen			
			TL	NE↓	SR↑	SPL↑	TL	NE↓	SR↑	SPL↑
1			10.078	3.429	0.667	0.645	10.894	4.701	0.556	0.519
2	✓		11.183	3.541	0.655	0.629	13.009	4.636	0.554	0.513
Full model	✓	✓	10.319	**3.408**	**0.684**	**0.657**	11.846	**4.574**	**0.561**	**0.523**

it's probably because the environment in the enhanced data has been processed by the EnvDrop method [14], and the instructions in the enhanced data are generated by the speaker model. The information about objects in the environment and objects in the instructions have been weakened to some extent, so the training has been affected to some extent.

D. Analysis

According to the above experimental process, we conduct a series of comparative analyses on the experimental results (including ablation experiments, etc.) In Table III, ticked off indicated that the new reward shaping was used. The table shows the contribution of the new reward shaping to navigation results as well as the model performance at different stages in the training process. Stage 1 represents the first stage, training only with training data, and stage 2 represents the second stage, training combined with training data and augmented data. The best results are in bold font.

Model 1 does not use the new reward shaping in the first stage and the second stage. The performance of NE, SR, and SPL in validation seen and validation unseen data sets is slightly weaker than that of the full model. In the seen environment, the overall performance of the full model is better, but the difference is not big. In the unseen environment, the performance of the full model is more prominent on NE, which is reduced by about 3% compared with Model 1, indicating that using the new reward shaping can effectively help the agent to stop in the correct position. In the unseen environment, this advantage is more obvious.

Model 2 uses the new reward shaping for training in the first stage but does not use it in the second stage. The experimental results show that Model 2 performs the worst among the three models, and almost all indicators in the seen and unseen environments are weaker than the other two models. It shows that only when new reward shaping is used completely in the two stages can the performance of the agent be improved. If new reward shaping is used only in a certain stage, the performance of the agent will be reduced. However, in the unseen environment, the performance of Model 2 on NE is better than that of Model 1, which indicates that our proposed new reward shaping is helpful in navigating to the correct target point in the unseen environment. In conclusion, the proposed method plays a significant role in reducing NE.

In addition, during the training of the three models, it is found that the training process in the stage in which the new reward shaping is adopted is quite oscillatory, especially in the early stage of training. Model 1 does not use the new reward shaping in the two stages, so the training is relatively stable. In the second training stage, Model 2 does not use new reward shaping and has a large change range in the process of fine-tuning. The performance of Model 2 on the validation set is poor in the early training stage. However, with the increase in the number of training iterations, the performance of Model 2 will also rise. In the middle stage of training, the performance does not increase much. When the number of training iterations approaches 300,000, the performance will reach its peak. The same is true for the Full model, but its upper-performance limit is higher than that of Model 1 and Model 2. When the performance becomes stable, its performance is better than that of Model 1 and Model 2.

V. CONCLUSIONS AND FUTURE WORK

A. Conclusion

In this paper, we introduce a novel reward shaping approach for mixed training of imitation learning and reinforcement learning in VLN tasks. The proposed reward shaping method incorporates two key factors: the change in distance between the agent and the target, and the progress made in navigation while following instructions. This approach considers both the accuracy of the route and the correctness of reaching the navigation target. Additionally, to enhance the utilization of object information, an object alignment method is proposed. At each step, the visual object information is aligned with the corresponding object information in the navigation instruction. This alignment allows the agent to obtain a rough estimation of navigation progress based on the object's position in the instruction, thus assisting the agent in following the natural language instruction for navigation. Compared with the baseline method, our proposed approach effectively reduces navigation errors, improves the correctness of the agent's navigation route and the accuracy of reaching the target location, and enhances overall agent performance.

B. Future Work

Due to the lack of training data and the limited diversity in the training environments, we used the EnvDrop method [14] in this work to generate a new environment by processing the existing environment and the speaker model [1] to generate new instructions, but these newly generated instructions are not of high quality, generally short in length, and contain less information about objects. As a result, the instructions are not clear, which also causes confusion to the agent and makes the navigation task more difficult. However, the object alignment method proposed by us exactly needs detailed object information. The richer the object information is, the more space our method can play. Therefore, richer and more accurate command generation will undoubtedly improve the performance of our method. Recently, a new high-quality instruction generator named Marky was proposed [23], and we look forward to using the new tool to obtain higher-quality enhanced data.

The correspondence between visual object information and instruction object information is an important factor affecting navigation results. The accuracy of this object information correspondence is also important for our new reward shaping. The higher the matching degree between vision and instruction, the greater the role our proposed method plays. Therefore, we hope that a more efficient and general semantic coding method can be developed to encode objects with semantic information. In this way, when matching visual object information and instruction object information, the visual information can correctly match the encoding of objects with similar semantics, and the disturbance to the model will be reduced.

Acknowledgment

This work was partly supported by the National Natural Science Foundation of China under Grant No. 62206199 and U2141234, Tianjin Applied Basic Research Project under Grant No. 22JCQNJC00410, Alexander von Humboldt Foundation Grant No. 1226831, the National Key R&D Program of China(2022ZD0119900), Shanghai Science and Technology program (22015810300), Hainan Province Science and Technology Special Fund (ZDYF2024GXJS003), and State Key Laboratory of Reliability and Intelligence of Electrical Equipment No.EERI-KF2022001.

References

[1] D. Fried *et al.*, "Speaker-follower models for vision-and-language navigation," in *Adv. Neural Inform. Process. Syst.*, vol. 31, 2018.

[2] Y. Hong, Q. Wu, Y. Qi, C. Rodriguez-Opazo, and S. Gould, "Vln bert: A recurrent vision-and-language bert for navigation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 1643–1653.

[3] J. Li, H. Tan, and M. Bansal, "Envedit: Environment editing for vision-and-language navigation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 15 407–15 417.

[4] C.-W. Kuo, C.-Y. Ma, J. Hoffman, and Z. Kira, "Structure-encoding auxiliary tasks for improved visual representation in vision-and-language navigation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 1104–1113.

[5] H. Tan and M. Bansal, "Lxmert: Learning cross-modality encoder representations from transformers," in *arXiv preprint arXiv:1908.07490*, 2019.

[6] P. Anderson *et al.*, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 3674–3683.

[7] A. Chang *et al.*, "Matterport3d: Learning from rgb-d data in indoor environments," in *arXiv preprint arXiv:1709.06158*, 2017.

[8] V. Jain, G. Magalhaes, A. Ku, A. Vaswani, E. Ie, and J. Baldridge, "Stay on the path: Instruction fidelity in vision-and-language navigation," in *arXiv preprint arXiv:1905.12255*, 2019.

[9] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," in *arXiv preprint arXiv:2010.07954*, 2020.

[10] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi, "Touchdown: Natural language navigation and spatial reasoning in visual street environments," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 12 538–12 547.

[11] P. Anderson *et al.*, "Sim-to-real transfer for vision-and-language navigation," in *Conference on Robot Learning*. PMLR, 2021, pp. 671–681.

[12] X. Wang, W. Xiong, H. Wang, and W. Y. Wang, "Look before you leap: Bridging model-free and model-based reinforcement learning for planned-ahead vision-and-language navigation," in *Eur. Conf. Comput. Vis.*, 2018, pp. 37–53.

[13] Y. Hong, C. Rodriguez, Y. Qi, Q. Wu, and S. Gould, "Language and visual entity relationship graph for agent navigation," in *Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 7685–7696, 2020.

[14] H. Tan, L. Yu, and M. Bansal, "Learning to navigate unseen environments: Back translation with environmental dropout," in *arXiv preprint arXiv:1904.04195*, 2019.

[15] A. Parvaneh, E. Abbasnejad, D. Teney, J. Q. Shi, and A. Van den Hengel, "Counterfactual vision-and-language navigation: Unravelling the unseen," in *Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 5296–5307, 2020.

[16] P.-L. Guhur, M. Tapaswi, S. Chen, I. Laptev, and C. Schmid, "Airbert: In-domain pretraining for vision-and-language navigation," in *Int. Conf. Comput. Vis.*, 2021, pp. 1634–1643.

[17] S. Chen, P.-L. Guhur, C. Schmid, and I. Laptev, "History aware multimodal transformer for vision-and-language navigation," in *Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 5834–5847, 2021.

[18] A. Pashevich, C. Schmid, and C. Sun, "Episodic transformer for vision-and-language navigation," in *Int. Conf. Comput. Vis.*, 2021, pp. 15 942–15 952.

[19] Y. Zhu *et al.*, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Int. Conf. Robot. Autom.*, 2017, pp. 3357–3364.

[20] S. Chen, P.-L. Guhur, M. Tapaswi, C. Schmid, and I. Laptev, "Think global, act local: Dual-scale graph transformer for vision-and-language navigation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 16 537–16 547.

[21] C. H. Song, J. Kil, T.-Y. Pan, B. M. Sadler, W.-L. Chao, and Y. Su, "One step at a time: Long-horizon vision-and-language navigation with milestones," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 15 482–15 491.

[22] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Int. Conf. Mach. Learn.*, vol. 99, 1999, pp. 278–287.

[23] S. Wang *et al.*, "Less is more: Generating grounded navigation instructions from landmarks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 15 428–15 438.